Université de Lorraine
DIPLOME: Telecom Nancy 2A - IL et SLE
DIPLOME: ENSEM
Épreuve: MOVEX Première épreuve
Durée du sujet : 1 h 30
Date : Mardi 19 mars 2024 de 8 h 00 à 9 h 30
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 trois (3) exercices.

Premier écrit

CORRECTION MANUSCRITE D. MERY

Exercice 1 10 points

Une annotation simple est une expression de la forme suivante:

$$\begin{array}{l} \ell_1 : P_1(v0, v) \\ \mathsf{v} := \mathsf{f}_{\ell_1,\ell_2}(\mathsf{v}) \\ \ell_2 : P_2(v0, v) \end{array}$$

Elle signifie que si les valeurs des variables v vérifient $P_1(v0, v)$, alors l'exécution de l'affectation $\mathsf{v} := \mathsf{f}_{\ell_1,\ell_2}(\mathsf{v})$ conduit aux nouvelles valeurs des variables v notées v' qui satisfont $P_2(v0, v')$. Pour vérifier cette condition, elle est traduite sous la forme suivantre:

condition de vérification: $P_1(v0, v) \wedge v' = f_{\ell_1,\ell_2}(v) \Rightarrow P_2(v0, v')$

Question 1.1 On suppose que a, b, u et v sont des valeurs constantes entières dans l'expression suivante:

$$\begin{array}{l} \ell_1 : y = a * x + b \wedge a, b, u, v, x, y \in \mathbb{Z} \\ \mathsf{x} := \mathsf{x} + \mathsf{u} \\ \ell_2 : y = a * x + b + v \end{array}$$

En utilisant la condition de vérification ci-dessus, donner une expession la plus simple possible pour que cette condition soit valide.

Question 1.2 On suppose que x, y et z sont des variables entières et que n est un entier naturel non nul et on considère l'expression suivante:

$$\begin{array}{l} \ell_1 : x + y = z \wedge x * y = 2^n \\ (\mathsf{x}, \mathsf{z}) := (\mathsf{x} * \mathsf{y}, \mathsf{y}^2); \\ \ell_2 : z \geq 2^n \end{array}$$

Question 1.3 Montrer que l'annotation suivante est correcte ou incorrecte selon les conditions de vérifications

$$\begin{array}{l} \ell_1 : x = 12 \ \wedge \ z = 3 * x \ \wedge y = 2 \ \wedge z = 4 * y \\ (x, y) := (z + y, x + y + z); \\ \ell_2 : x = 38 \ \wedge \ y = 2 \end{array}$$

Question 1.4 Montrer que l'annotation suivante est correcte ou incorrecte selon les conditions de vérifications

$$\begin{array}{l} \ell_1 : x = 3 \ \wedge \ y = 9 \\ x := 3 * y \\ \ell_2 : x = 27 \ \wedge \ y = 9 \end{array}$$

Question 1.5 Soit p un nombre différent d'un multiple de 3 c'est-à-dire différent de 0, 3, 6, 9, 12, …
;Montrer que l'annotation suivante est correcte ou incorrecte selon les conditions de vérifications

$$\ell_1 : x = 3 + z \ \wedge\ y = 1 \ \wedge\ z = 3 \ \wedge x = y$$
$$x := p * y$$
$$\ell_2 : x = z \ \wedge\ y = z \ \wedge z = 4 * p$$

Exercice 2 (4 points)
Soit le contrat suivant qui met en jeu les variables X,Y, Z,C,R.

variables `int` X, Y, Z, C, R

requires $x_0, y_0, z_0, c_0, r_0 \in \mathbb{Z}$

ensures $r_f = 0$

begin
$0 : x = x_0 \wedge y = y_0 \wedge z = z_0 \wedge c = c_0 \wedge r = r_0 \wedge x_0, y_0, z_0, c_0, r_0 \in \mathbb{Z}$
$(X, Z, Y) := (625, 2 * C, (2 * C + 1) * (2 * C + 1));$
$1 : x = 625 \wedge z = 2 * c \wedge y = (z + 1) * (z + 1)$
$Y := X + Z + 1;$
$2 : x = 625 \wedge z = 2 * c \wedge y = (c + 1) * (c + 1)$
end

Question 2.1 Ecrire les conditions de vérification associée au contrat ci-dessus en vous aidant du rappel de la définition de ces conditions de vérification.

Question 2.2 Simplifier les conditions de vérification et préciser les conditions que doivent vérifier les valeurs initiales des variables X,Y,Z,C,R pour que les conditions de vérification soient toutes vraies. En particulier, il faudra s'assurer que la précondition est satisfaisable.

Exercice 3 6 points
Nous allons étudier l'algorithme A annoté de la figure ??.

Question 3.1 Compléter les annotations incomplètes où vous pourrez voir .

Question 3.2 Vérifier les conditions de vérification associées aux transitions suivantes:

1. ℓ_0, ℓ_1
2. ℓ_1, ℓ_2
3. ℓ_2, ℓ_3
4. ℓ_0, ℓ_{10}

Question 3.3 Donner et vérifier les points pour assurer la correction partielle de cet algorithme.

Question 3.4 Que calcule cet algorithme?

On rappelle qu'un contrat pour la correction partielle d'un petit programme est donné par les éléments ci-dessou en colonne de gauche et que les conditions de vérification associées sont définies par le texte de la colonne de droite.

VARIABLES N, X, Y, Z

$$pre(n_0, x_0, y_0, z_0) \stackrel{def}{=} \begin{cases} n_0 \in \mathbb{N} \\ x_0, y_0, z_0 \in \mathbb{Z} \end{cases}$$

REQUIRES $pre(n_0, x_0, y_0, z_0)$

ENSURES $\begin{pmatrix} z_f = n_f \land n_f = n_0 \\ x_f + y_f = n_f \ \land x_f = (n_f/2)(n_f/2 + 1) \\ even(n_f) \Rightarrow y_f = (n_f/2) * (n_f/2) \\ odd(n_f) \Rightarrow y_f = ((n_f/2) + 1) * ((n_f/2) + 1) \end{pmatrix}$

$\ell_0 : (\ pre(n_0, x_0, y_0, z_0) \ \land \ (n, x, y, z) = (n_0, x_0, y_0, z_0)$
$Z := 0$
$\ell_1 : (\ pre(n_0, x_0, y_0, z_0) \ \land \ z = 0 \ \land \ (n, x, y) = (n_0, x_0, y_0)$
$(X, Y) := (0, 0)$
$\ell_2 : \begin{pmatrix} pre(n_0, x_0, y_0, z_0) \ \land \ x + y = z \ \land \ x = (z/2) * (z/2 + 1) \ \land \ n = n_0 \\ even(z) \Rightarrow y = (z/2) * (z/2) \ \land \ odd(z) \Rightarrow y = (z/2 + 1) * (z/2 + 1) \end{pmatrix}$
WHILE $Z < N$ DO
$\ell_3 : \begin{pmatrix} pre(n_0, x_0, y_0, z_0) \ \land \ x + y = z \ \land \ x = (z/2) * (z/2 + 1) \land \ n = n_0 \\ even(z) \Rightarrow y = (z/2) * (z/2) \ \land \ odd(z) \Rightarrow y = (z/2 + 1) * (z/2 + 1) \\ 0 \leq z < n \end{pmatrix}$
$Z := Z + 1;$
$\ell_4 : \begin{pmatrix} pre(n_0, x_0, y_0, z_0) \ \land \ x + y = z - 1 \ \land \ x = (z - 1/2) * (z - 1/2 + 1) \ \land \ n = n_0 \\ even(z - 1) \Rightarrow y = (z - 1/2) * (z - 1/2) \ \land \ odd(z - 1) \Rightarrow y = (z - 1/2 + 1) * (z - 1/2 + 1) \\ 0 < z \leq n \end{pmatrix}$
IF $even(Z)$ THEN
$\ell_5 : \begin{pmatrix} pre(n_0, x_0, y_0, z_0) \ \land \ x + y = z - 1 \ \land \ x = (z - 1/2) * (z - 1/2 + 1) \ \land \ n = n_0 \\ even(z - 1) \Rightarrow y = (z - 1/2) * (z - 1/2) \ \land \ odd(z - 1) \Rightarrow y = (z - 1/2 + 1) * (z - 1/2 + 1) \\ 0 < z \leq n \land even(z) \end{pmatrix}$
$X := X + Z$
$\ell_6 : \begin{pmatrix} pre(n_0, x_0, y_0, z_0) \ \land \ x + y = z \ \land \ x = (z/2) * (z/2 + 1) \ \land \ n = n_0 \\ even(z) \Rightarrow y = (z/2) * (z/2) \ \land \ odd(z) \Rightarrow y = (z/2 + 1) * (z/2 + 1) \\ 0 < z \leq n \ \land odd(z) \end{pmatrix}$
ELSE
$\ell_7 : \begin{pmatrix} pre(n_0, x_0, y_0, z_0) \ \land \ x + y = z - 1 \ \land \ x = (z - 1/2) * (z - 1/2 + 1) \ \land \ n = n_0 \\ even(z - 1) \Rightarrow y = (z - 1/2) * (z - 1/2) \ \land \ odd(z - 1) \Rightarrow y = (z - 1/2 + 1) * (z - 1/2 + 1) \\ 0 < z \leq n \land odd(z) \end{pmatrix}$
$Y := Y + Z$
$\ell_8 : \begin{pmatrix} pre(n_0, x_0, y_0, z_0) \ \land \ x + y = z \ \land \ x = (z/2) * (z/2 + 1) \ \land \ n = n_0 \\ even(z) \Rightarrow y = (z/2) * (z/2) \ \land \ odd(z) \Rightarrow y = (z/2 + 1) * (z/2 + 1) \\ 0 < z \leq n \ \land even(z) \end{pmatrix}$
FI;
$\ell_9 : \begin{pmatrix} pre(n_0, x_0, y_0, z_0) \ \land \ x + y = z \ \land \ x = (z/2) * (z/2 + 1) \ \land \ n = n_0 \\ even(z) \Rightarrow y = (z/2) * (z/2) \ \land \ odd(z) \Rightarrow y = (z/2 + 1) * (z/2 + 1) \\ 0 < z \leq n \end{pmatrix}$
OD;
$\ell_{10} : \begin{pmatrix} pre(n_0, x_0, y_0, z_0) \ \land \ x + y = z \ \land \ x = (z/2) * (z/2 + 1) \land \ \land n = n_0 \\ even(z) \Rightarrow y = (z/2) * (z/2) \ \land \ odd(z) \Rightarrow y = (z/2 + 1) * (z/2 + 1) \\ z = n \end{pmatrix}$

Figure 1: Algorithme A

Contrat de la correction partielle

variables $type\ X$

definitions
$def1 \stackrel{def}{=} text1$

requires $pre(x_0)$

ensures $post(x_0, x_f)$

$$\left[\begin{array}{l} \text{begin} \\ 0 : P_0(x_0, x) \\ \texttt{instruction}_0 \\ 1 : P_i(x_0, x) \\ \texttt{instruction}_1 \\ f : P_f(x_0, x) \\ \text{end} \end{array}\right.$$

Conditions de vérification

- $pre(x_0) \wedge x = x_0 \Rightarrow P_0(x_0, x)$
- $pre(x_0) \wedge P_f(x_0, x) \Rightarrow post(x_0, x)$
- Pour toutes les paires ℓ, ℓ', telles que $\ell \longrightarrow \ell'$, on vÃ©rifie que, pour toutes les valeurs $x, x' \in$ MEMORY
$$\left(\begin{array}{l} \left(\begin{array}{l} pre(x_0) \wedge P_\ell(x_0, x)) \\ \wedge cond_{\ell,\ell'}(x) \wedge x' = f_{\ell,\ell'}(x) \end{array} \right) \\ \Rightarrow P_{\ell'}(x_0, x') \end{array} \right),$$

Correction de l'épreuve du 19/3/2024

Exercice 1

Q1. Soit l'annotation suivante

$e_1 : y = ax+b \wedge a,b,u,v,x,y \in \mathbb{Z}$.

$x := x + u;$

$e_2 : y = ax+b+v$

On écrit la condition à appliquer.

$y = ax+b \wedge a,b,u,v,x,y \in \mathbb{Z}$

$\wedge\ x' = x+u \wedge u' = u \wedge a' = a \wedge b' = b \wedge v' = v \wedge y' = y$.

$\Rightarrow\ y' = a'x' + b' + v'$

On applique les simplifications usuelles de 2ᵉ ligne en ligne du calcul +

$y = ax+b \wedge a,b,u,v,x,y \in \mathbb{Z} \wedge$

$x' = x+u \wedge (a,b,u,v,y)' = (a,b,u,v,y)$

$\vdash y' = a'x' + b' + v'$

$\vdash y = a(x+u) + b + v$ (substitution des temps logiques)

$\vdash y = ax + au + b + v$ (simplifiat)

$\vdash y = y + au + v$ ("$y = ax + b$")

$\vdash 0 = au + v$ (simplifiats).

$\vdash \boxed{au + v = 0}$

La condition de validité de cette équivalence est $\boxed{au + v = 0}$.

Q1.2. x, y, z sont des variables et $n \in \mathbb{N}^*$.

$e_1 : x + y = z \wedge x * y = 2^n$

$(x, z) := (x * y, y^2);$

$e_2 : z \geq 2^n.$

On utilise les mêmes transformations et on déduit.

la condition suivante :

$$x+y=z \wedge xy=2^n \wedge x'=xy \wedge z'=y^2 \wedge y'=y \Rightarrow z' \geq 2^n.$$

On applique les règles du calcul $\vdash$:

$x+y=z \wedge xy=2^n \wedge x'=xy \wedge z'=y^2 \wedge y'=y$

$\vdash z' \geq 2^n$

$\vdash y^2 \geq 2^n$ (puisque $z'=y^2$).

$\vdash y^{2q} \geq 2^n$ (puisque y est de la forme 2^q avec x de la forme 2^p et $p+q=n$.)

La condition est donc $\boxed{2q \geq n}$

pour que l'assertion soit vraie.

Q 1.3.

e1 : $x=12 \wedge z=3x \wedge z=4y$

$(x,y) := (z+y, x+y+z)$;

e2 : $x=38 \wedge y=2$

On traduit cette annotation:

$$x=12 \wedge z=3x \wedge y=2 \wedge z=4y$$
$$\wedge\ x'=z+y \wedge y'=x+y+z \wedge z'=z.$$
$$\Rightarrow x'=38 \wedge y'=2.$$

On traduit dans le calcul $\vdash$:

$$x=12 \wedge z=3x \wedge y=2 \wedge z=4y$$
$$\wedge\ x'=z+y \wedge y'=x+y+z \wedge z'=z$$
$$\vdash x'=38 \wedge y'=2.$$

Avant de démarrer, on observe que les hypothèses permettent de montrer:

$$x=12, z=3x, y=2, z=4y \vdash z=36 \wedge z=8$$

On peut donc ajouter $\Vdash$ FALSE

D'où

$$x=12, z=3x, y=2, z=4y, \text{FALSE}, x'=z+y, y'=x+y+z$$
$$, z'=z \vdash x'=38 \wedge y'=2.$$

On the adult cello annotation.

$x + y = z \wedge xy$

On applique la règle :

$P_1, FALSE, P_2 \vdash P$ pour toute propriété P.

Donc le séquent est correct.

Q 1.4.

$e1 : x = 3 \wedge y = 9$

$x := 3y;$

$e2 : x = 27 \wedge y = 9.$

On traduit l'annotation

$$x = 3 \wedge y = 9 \wedge x' = 3y \wedge y' = y$$
$$\Rightarrow x' = 27 \wedge y' = 9$$

Puis.

$$x = 3 \wedge y = 9 \wedge x' = 3y \wedge y' = y$$
$$\vdash x' = 27 \wedge y' = 9.$$
$$\vdash 3y = 27 \wedge y = 9$$
$$\vdash 3.9 = 27 \wedge 9 = 9$$
$$\vdash TRUE.$$

Q 1.5.

$e_1: x = 3+z \wedge y = 1 \wedge z = 3 \wedge x = y$.

$x := py$;

$e_2: x = z \wedge y = z \wedge z = 4p$.

On traduit l'annotation sous la forme suivante :

$x = 3+z \wedge y = 1 \wedge z = 3 \wedge x = y$

$\wedge\, x' = py \wedge y' = y \wedge z' = z$.

$\Rightarrow x' = z' \wedge y' = z' \wedge z' = 4p$

Puis on applique le calcul $\vdash$.

$x = 3+z, y = 1, z = 3, x = y, x' = py, y' = y, z' = z$

$\vdash x' = z' \wedge y' = z' \wedge z' = 4p$

$\vdash px = z \wedge y = z \wedge z = 4p$.

On applique la même règle pour déduire FALSE

$x = 3+3, y = 1, z = 3, x = y \vdash x = 6 \wedge x = y \wedge y = 1$

$\vdash x = 6 \wedge x = 1$.

$\vdash$ FALSE.

6

On en déduit que l'annotation est correcte.

Exercice 2

$pre(x_0, y_0, z_0, c_0, r_0) \triangleq$

$x_0, y_0, z_0, c_0, r_0 \in \mathbb{Z}$.

Init) $x_0, y_0, z_0, c_0, r_0 \in \mathbb{Z} \wedge x = x_0 \wedge y = y_0 \wedge z = z_0 \wedge r = r_0 \Rightarrow \begin{pmatrix} x = x_0 \wedge y = y_0 \wedge z = z_0 \wedge c = c_0 \\ \wedge r = r_0 \wedge x_0, y_0, z_0, c_0, r_0 \in \mathbb{Z} \end{pmatrix}$

End) $x = 625 \wedge z = 2c \wedge y = (c+1) \wedge (c+1) \wedge pre(x_0, y_0, z_0, c_0, r_0)$

$\Rightarrow r = 0$

Init) $\circ \longrightarrow 1$.

$x = x_0 \wedge y = y_0 \wedge z = z_0 \wedge c = c_0 \wedge r = r_0 \wedge pre(x_0, y_0, z_0, c_0, r_0)$
$\wedge x' = 625 \wedge z' = 2c \wedge y' = (2c+1)^2 \wedge c' = c \wedge r' = r$
$\Rightarrow x' = 625 \wedge z' = 2c' \wedge y' = (z'+1)^2$.

+

$1 \longrightarrow 2$

$x = 625 \wedge z = 2c \wedge y = (z+1)^2 \wedge$

$y' = \cancel{x} + z + 1 \wedge x' = x \wedge z' = z \wedge c' = c \wedge v' = v.$

$\Rightarrow x' = 625 \wedge z' = 2c \wedge y' = (c'+1)(c'+1)$

Les deux conditions init et arret sont valides par construction.

$0 \longrightarrow 1$: on remplace les valeurs primées pour obtenir

$\vdash 625 = 625 \wedge 2c = 2c \wedge (2c+1)^2 = (2c+1)^2$

qui est vrai.

$1 \longrightarrow 2$: on remplace les valeurs primées

$\vdash x = 625 \wedge z = 2c \wedge \cancel{x} + z + 1 = (c+1)^2$

$\vdash 625 = 625 \wedge \cancel{(625)} \overset{2c}{+ z} + 1 = (c+1)^2$

$\vdash 625 + 2c + c^2 + 2c + 1$

$\vdash c^2 = 625$

On en déduit $c \in \{-25, 25\}$.

8

On notera que la condition initiale est satisfaite mais que la valeur finale de r doit être 0. Or r ne change pas et donc

$r_0 = 0$.

Exercice 3

Q3.1. cf la feuille de l'algorithme.

Q3.2. $\mathit{lo} \to \ell_1$.

$pre(m_0, n_0, 40, 30) \wedge (m, n, y, z) := (m_0, n_0, 40, 30)$
$\wedge\ \cancel{z}' = 0 \wedge \cancel{y}' = \cancel{x} \wedge \ldots = 4$.
$\Rightarrow pre(m_0, n_0, 40, 30) \wedge z' = 0 \wedge \ldots$
$\wedge (m', x', y) = (m_0, n_0, 40)$.
$\vdash h.(m_0, m_0, 40, 30)$ (par hypothèse)
$\vdash z = 0$ (substituts)
$\vdash (m, x, y) = (m_0, m_0, 40)$ (substituts)

$1 \longrightarrow 2$

$pre(n_0, x_0, y_0, z_0) \wedge z=0 \wedge (n, x, y) = (n_0, x_0, y_0) \wedge x'=0 \wedge y'=0 \wedge z'=2$.

$\Rightarrow pre(n_0, x_0, y_0, z_0) \wedge x'+y'=2 \wedge$
$x' = (z'/2) \times (z'/2 - 1)$ ~~[illegible]~~
$\wedge n' = n_0 \wedge$
$(even(z') \Rightarrow y' = (z'/2)(z'/2))$
$(odd(z') \Rightarrow y = (z'/2-1)((z'/2-1))$

Q33.

On vérifie que la précondition est établie et que la postcondition est établie. Puis on démontre chaque relation l-sé' possible.

Q34. Cet algorithme calcule la somme des valeurs paires et impaires de 1 à n_0.