Université de Lorraine
DIPLÔME: Telecom Nancy 2A - Apprentissage
Épreuve: MVSI annales

Annales MVSI 2020-2023

1 Rappel 1

On rappelle que la condition de vérification $\forall v.P_{\ell_1}(v) \ \wedge \ cond_{\ell_1,\ell_2}(v) \ \wedge \ (v') = f_{\ell_1,\ell_2}(v) \Rightarrow P_{\ell'}(v')$ et correspond à une instruction de la forme

$$\begin{array}{|l|}\hline \ell_1 : P_{\ell_1}(v) \\ V := f_{\ell_1,\ell_2}(V) \\ \ell_2 : P_{\ell_2}(v) \\ \\ \hline \end{array}$$

2 Rappel 2

Une équation du second degré de la forme $ax^2 + bx + c = 0$ est analysée comme suit:

- $\Delta = b^2 - 4ac$ est le discriminant.
- **Cas 1** Le discriminant est négatif non nul et l'équation n'a pas de solutions réelles.
- **Cas 2** Le discriminant est nul et l'équation a une unique solution $x_0 = \frac{-b}{2a}$ et l'équation peut être factorisée sous la forme $a(x - x0) * (x - x_0) = ax^2 + bx + c$.
- **Cas 3** Le discriminant est positif non nul et l'équation a deux solutions
  - $x_1 = \frac{-b-\sqrt{\Delta}}{2a}$
  - $x_2 = \frac{-b+\sqrt{\Delta}}{2a}$

  et l'équation peut être factorisée sous la forme $a(x - x_1) * (x - x_2) = ax^2 + bx + c$.

3 Rappel 3

On rappelle que l'annotation suivante du listing 1 est correcte , si les conditions suivantes sont vérifiées:

- $pre(v_0) \wedge v = v_0 \Rightarrow A(v_0, v)$
- $pre(v_0) \wedge B(v_0, v) \Rightarrow post(v_0, v)$
- $A(v_0, v) \Rightarrow wp(v = f(v))(B(v_0, v)$ où $wp(v = f(v))(B(v_0, v))$ est définie par $B(v_0, v)[f(v)/v)]$.

Dans le cas de Frama-c, la valeur initiale d'une variable v est notée $\backslash at(v, Pre)$ et aussi $\backslash old(v)$. Nous utiliserons la notation v_0 dans cet exercice.

Listing 1: contrat

```
requires pre(v)
ensures post(\old(v),v)
type1 truc(type2 v)
  /*@ assert A(v0,v); */
  v = f(v);
  /*@  assert  B(v0,v);      */
return  val;
```

4 Rappel 4

On rappelle que { P } S { Q } est défini par l'implication $O \Rightarrow WP(S)(Q)$. La dÃ©finition du terme $WP(S)(Q)$ est donnÃ©e dans ce qui suit:

S	wp(S)(P)
X:=E(X,D)	$P[e(x,d)/x]$
SKIP	P
$\mathrm{S}_1;\mathrm{S}_2$	$wp(\mathrm{S}_1)(wp(\mathrm{S}_2)(P))$
IF B S_1 ELSE S_2 FI	$(B \Rightarrow wp(S_1)(P)) \wedge (\neg B \Rightarrow wp(S_2)(P))$
WHILE B DO S OD	$\mu.(\lambda X.(B \Rightarrow wp(S)(X)) \wedge (\neg B \Rightarrow P))$

ex2021.tex

Exercice 1

$\ell_0 : u = a * a \wedge v = b * b \wedge a \in \mathbb{N} \wedge b \in \mathbb{N}$
$w := u + v;$
$\ell_1 : w = (a+b)^2 - 2 * a * b$

Soit l'annotation suivante. On suppose que a et b sont des constantes entières positives.

Montrer que l'annotation est correcte ou incorrecte selon les conditions de vérifications 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* c'est-à-dire en indiquant les differents pas de transformation.

Exercice 2

$\ell_1 : x = r \;\wedge\; u = x^r \;\wedge\; z = 6 \;\wedge x = u$
$y := r * r * r$
$\ell_2 : x = z \;\wedge\; y = z \;\wedge z = 4 * p$

Soit r un nombre cubique c'est-à-dire de la forme $r = q^3$ où q est un entier positif. p est un entier positif.

Montrer que l'annotation est correcte ou incorrecte selon les conditions de vérifications 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* c'est-à-dire en indiquant les differents pas de transformation.

Exercice 3

$\ell_1 : x = 5 + z \;\wedge\; y = 1 \;\wedge\; z = 3 \;\wedge x = y$
$x := p * y$
$\ell_2 : x = z \;\wedge\; y = z \;\wedge z = 4 * p$

Soit p un nombre différent d'une puissance de 5 c'est-à-dire différent de 5, 10, 15, 20, ...

Montrer que l'annotation est correcte ou incorrecte selon les conditions de vérifications 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* c'est-à-dire en indiquant les differents pas de transformation.

Exercice 4

Soit l'algorithme annoté suivant se trouvant à la page suivante et les pré et postconditions définies pour cet algorithme comme suit :

- Precondition: $x_1 \in \mathbb{N} \wedge x_2 \wedge \mathbb{N} \wedge x_1 \neq 0$
- Postcondition: $z = {x_1}^{x_2}$

On suppose que x_1 et x_2 sont des constantes.
Pour chaque paire (ℓ, ℓ') d'étiquettes correspondant à un pas élémentaire; on vérifie la propriété suivante :
$\forall x, y, q, r, x', y', q', r'.P_\ell(y_1, y_2, y_3, z) \wedge cond_{\ell,\ell'}(y_1, y_2, y_3, z) \wedge (y'_1, y'_2, y'_3, z') = f_{\ell,\ell'}(y_1, y_2, y_3, z) \Rightarrow P_{\ell'}(y'_1, y'_2, y'_3, z')$

Question 4.1 Enoncer et vérifier cette propriété pour les paires d'étiquettes suivantes: (ℓ_0, ℓ_1);

precondition : $x_1 \in \mathbb{N} \wedge x_2 \in \mathbb{N} \wedge x_1 \neq 0$
postcondition : $z = {x_1}^{x_2}$
local variables : $y_1, y_2, y_3 \in \mathbb{Z}$

$\ell_0 : \{y_1, y_2, y_3, z \in \mathbb{Z}\}$
$(y_1, y_2, y_3) := (x_1, x_2, 1);$
$\ell_1 : \{y_3 * {y_1}^{y_2} = {x_1}^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$
while $y_2 \neq 0$ **do**
  $\ell_2 : \{y_2 \neq 0 \wedge y_3 * {y_1}^{y_2} = {x_1}^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$
  if $impair(y_2)$ **then**
    $\ell_3 : \{impair(y_2) \wedge y_2 \neq 0 \wedge \boxed{\ldots} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$
    $y_2 := y_2 - 1;$
    $\ell_4 : \{y_2 \geq 0 \wedge pair(y_2) \wedge y_3 * y_1 * {y_1}^{y_2} = {x_1}^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$
    $y_3 := y_3 * y_1;$
    $\ell_5 : \{y_2 \geq 0 \wedge pair(y_2) \wedge y_3 * {y_1}^{y_2} = {x_1}^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$
  ;
  $\ell_6 : \{y_2 \geq 0 \wedge pair(y_2) \wedge \boxed{\ldots} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$
  $y_1 := y_1 * y_1;$
  $\ell_7 : \{y_2 \geq 0 \wedge pair(y_2) \wedge y_3 * {y_1}^{y_2 \; div 2} = {x_1}^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$
  $y_2 := y_2 \; div \; 2;$
  $\ell_8 : \{y_2 \geq 0 \wedge \boxed{\ldots} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$
;
$\ell_9 : \{y_2 = 0 \wedge \boxed{\ldots} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$
$z := y_3;$
$\ell_{10} : \{y_2 = 0 \wedge y_3 * {y_1}^{y_2} = {x_1}^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z} \wedge z = {x_1}^{x_2}\}$

Algorithme 1: Algorithme de l'exponentitaion indienne annoté

Question 4.2 Enoncer et vérifier cette propriété pour les paires d'étiquettes suivantes: (ℓ_1, ℓ_2);

Question 4.3 Enoncer et vérifier cette propriété pour les paires d'étiquettes suivantes: (ℓ_4, ℓ_5);

Question 4.4 Enoncer et vérifier cette propriété pour les paires d'étiquettes suivantes: (ℓ_7, ℓ_8);.

ex2122.tex

Exercice 5 Simplifier les expressions suivantes:

1. WP(X:=45)(x+y=45)
2. WP(X:=X+Y)(x+y=x)
3. WP(X:=X+Y+Z)(x+y=5)
4. WP(X:=X-Y)($\exists k.k \in \mathbb{N} \wedge a + x = a * k + y$)

Exercice 6 On rappelle que $\{P\}\mathbf{S}\{Q\}$ est défini par l'implication $O \Rightarrow WP(S)(Q)$. Pour chaque point énuméré ci-dessous, montrer que la propriété $\{P\}\mathbf{S}\{Q\}$ est valide ou pas en utilisant la définition suivante:

$$\{P\}\mathbf{S}\{Q\} = \text{P} \Rightarrow \text{WP(S)(Q)}$$

1. $\{x + y = 5\}$**X:=Y+2*X**$\{3 * x + y = 9\}$
2. $\{x \leq y\}$**X:=Y-8;Y:=Y+5**$\{2 * x \leq 2 * y\}$
3. $\{x \leq y\}$**X:=Y-8;Y:=Y+5**$\{2 * x \leq 2 * y\}$
4. $\{x > y\}$**IF X $\neq$ Y THEN X:= 1 ELSE X:= 0 FI**$\{x = 17 \wedge y = 45\}$
5. $\{x > y\}$**IF X $\neq$ Y THEN X:= 1 ELSE X:= 0 FI**$\{x = 7 \wedge y = 0\}$
6. $\{x > y\}$**IF X $\neq Y$ THEN X:= 1 ELSE X:= 0 FI**$\{x > y\}$

Exercice 7

Les annotations suivantes sont correctes ou non correctes et, pour cela, on utilise, soit la fonction WP pour traduire l'annotation sous la forme d'une implication qui est ensuite validée ou non, soit la technique classique de traduction avec la condition $(\forall x, y, z, x', y', z'.P_{\ell_1}(x, y, z) \ \wedge \ cond_{\ell,\ell'}(x, y, z) \ \wedge \ (x', y', z') = f_{\ell,\ell'}(x, y) \Rightarrow P_{\ell'}(x', y', z'))$. Appliquer l'une des techniques aux questions suivantes.

Question 7.1

$$\boxed{\begin{array}{l} \ell_1 : x = 1000 \ \wedge \ y = z + x \ \wedge z = 2 * x \\ y := z + x \\ \ell_2 : x = 2000/2 \ \wedge \ y = x + 2 * 1000 \end{array}}$$

Question 7.2 Soient trois variables x,y,z qui ont des valeurs entières a priori.

$$\boxed{\begin{array}{l} \ell_1 : x = 25 \ \wedge \ z = 2 * c \ \wedge y = (z + 1)^2 \\ y := x + z + 1 \\ \ell_2 : x = 25 \ \wedge \ z = 2 * c \wedge y = (c + 1)^2 \end{array}}$$

En utilisant la condition de vérification, déterminer la valeur ou les valeurs de c pour que l'annotation soit correcte.

Question 7.3 On suppose que a et b sont des constantes entières et que x, y, z et t sont des variables entières.

$$\boxed{\begin{array}{l} \ell_1 : x = a \wedge z = x^2 \wedge y = b * b \wedge t = b \\ Y := X * Z + 3 * Z * T + 3 * X * Y + Y * T \\ \ell_2 : y = (t + x)^3 \end{array}}$$

Exercice 8 Soit trois variables x,y,z qui ont des valeurs entières a priori.

$\ell_1 : x = 121 \ \wedge \ z = 2*c \ \wedge y = (z+1)^2$
$y := x + z + 1$
$\ell_2 : x = 121 \ \wedge \ z = 2*c \wedge y = (c+1)^2$

Question 8.1 L'annotation est correcte si la propriété suivante est vraie: $\forall x, y, z, x', y', z'.P_{\ell_1}(x, y, z) \ \wedge \ cond_{\ell,\ell'}(x, y, z) \ \wedge \ (x', y', z') = f_{\ell,\ell'}(x, y) \Rightarrow P_{\ell'}(x', y', z')$

En utilisant la condition de vérification, déterminer la valeur ou les valeurs de c pour que l'annotation soit correcte.

Question 8.2 Vérifiez votre réponse précédente en utilisant le calcul WP.

Exercice 9

On considère le petit programme se trouvant à droite de cette colonne. On va faire quelques opérations de vérifications sur ce programme à l'aide des WP. On ne s'intéresse pas aux erreurs à l'exécution.

Question 9.1 Montrer la validité des annotations entre le points ℓ_2 et ℓ_3 et entre les points ell_4 et ℓ_5 en énonçant les deux conditions avec le calcul WP et en vérifiant les deux conditions de vérification.

Question 9.2 En utilisant le calcul WP pour l'instruction IF, énoncer la propriété mettant en œuvre le WP de ce IF et appliquer le à la vérification entre les points ell_1 et ℓ_6. Vous pouvez aussi utiliser la règle du If rappelée à la fin de ce document.

Question 9.3 En utilisant les deux questions précédentes, montrer l'annotation entre ell_0 et ℓ_6 soit en utilisant le WP soit en utilisant la règle du if rappelée à la fin de ce document.

VARIABLES N, X
REQUIRES $n_0, x_0 \in \mathbb{Z}$
ENSURES $\begin{pmatrix} n_0 < 10 \Rightarrow x_f = n_0 + 1 \\ n_0 \geq 10 \Rightarrow x_f = 10 \\ n_f = n_0 \end{pmatrix}$
BEGIN
ℓ_0 :
$X := N;$
$\ell_1 : n_0, x_0 \in \mathbb{Z} \wedge n = n_0 \wedge x = n_0$
IF $X < 10$ **THEN**
$\ell_2 : n_0, x_0 \in \mathbb{Z} \wedge n = n_0 \wedge x = n_0 \wedge x < 10$
$X := X + 1;$
$\ell_3 : n_0, x_0 \in \mathbb{Z} \wedge n = n_0 \wedge x = n_0 + 1$
ELSE
$\ell_4 : n_0, x_0 \in \mathbb{Z} \wedge n = n_0 \wedge x = n_0 \wedge x \geq 10$
$X := 10;$
$\ell_5 : n_0, x_0 \in \mathbb{Z} \wedge n = n_0 \wedge x = 10$
FI
$\ell_6 : n_0, x_0 \in \mathbb{Z} \wedge (n_0 < 10 \Rightarrow x = 10) \wedge (n_0 \geq 10 \Rightarrow x = 10)$
END

ex2223.tex

On rappelle que la condition de vérification $\forall v.P_{\ell_1}(v) \ \wedge \ cond_{\ell_1,\ell_2}(v) \ \wedge \ (v') = f_{\ell_1,\ell_2}(v) \Rightarrow P_{\ell'}(v')$ et correspond à une instruction de la forme

$\ell_1 : P_{\ell_1}(v)$
$V := f_{\ell_1,\ell_2}(V)$
$\ell_2 : P_{\ell_2}(v)$

Exercice 10 (6 points)

Evaluer la validité de chaque annotation dans les questions suivantes.

Question 10.1 (3 points))

$$\ell_1 : x = 32 \ \wedge \ y = 2 * x * z \ \wedge z = 2 * x$$
$$Y := X * Z$$
$$\ell_2 : \ y * z = 2 * x * x * z$$

Question 10.2 (3 points))

Soient trois constantes n,m,p

$$\ell_1 : x = 3^n \wedge y = 3^p \wedge z = 3^m;$$
$$T := 8 * X * Y * Z;$$
$$\ell_2 : \ t = (y + z)^3 \wedge y = x;$$

Exercice 11 (4 points))

Soit le petit programme suivant annoté mais incomplet. Les valeurs des trois constantes p,q,r ne sont pas connues.

Listing 2: schema de contrat

```
#define p ?
#define q ?
#define r ?
/*@ requires x == 2*y && z == 4*y;
  ensures \result == r;
*/

int qq(int x,int y, int z){
  /*@ assert x +y ==  p * \at(y,Pre);  */
  y = x+z;
  /*@  assert  y == q*\at(y,Pre);      */
  return r;
  }
```

On rappelle que l'annotation suivante du listing 3 est correcte , si les conditions suivantes sont vérifiées:

- $pre(v_0) \wedge v = v_0 \Rightarrow A(v_0, v)$
- $pre(v_0) \wedge B(v_0, v) \Rightarrow post(v_0, v)$
- $A(v_0, v) \Rightarrow wp(v = f(v))(B(v_0, v)$ où $wp(v = f(v))(B(v_0, v))$ est définie par $B(v_0, v)[f(v)/v)]$.

Dans le cas de frama-c, la valeur initiale d'une variable v est notée $\backslash at(v, Pre)$ et aussi $\backslash old(v)$. Nous utiliserons la notation v_0 dans cet exercice. Dans l'exemple que nous allons traiter, il faut noter que v est une liste (x, y, z, r) où r désigne la variable correspondant à $return\ val$ ou $r = val$.

Listing 3: schema de contrat

```
requires pre(v)
ensures post(\old(v),v)
type1 truc(type2 v)
  /*@ assert A(v); */
  v = f(v);
  /*@  assert  B(v);      */
return  val;
  }
```

Question 11.1 (2 points))

Le listing 7 décrit un contrat avec un code associé. Enoncer et simplifier les trois conditions de correction de l'annotation du listing 7.

Question 11.2 (2 points)) Proposer un jeu de trois valeurs pour p,q,r, afin que les conditions de vérification soient correctes.

Listing 4: schema de contrat

Exercice 12

```
int main(){
  int x=5;
  int y;
  int z;
  /*@ assert x == 4 ; */
  y = x + 6;
  /*@ assert x == 5 && y + x == 16  ; */
  z = x + y;
  /*@ assert x == 5 && y + x == 16 && z + x + y == 32 ; */
  return 0;
  }
```

Soit le listing 6.
Appliquer la technique de remontée des plus faibles pré-conditions pour établir ou non la correction de cette annotation.

Exercice 13 (2 points)
Simplifier les expressions suivantes:

1. WP(X:=45)(x+y+z==789)
2. WP(X:=X-Y)($\exists k.k \in \mathbb{N} \wedge a + x = a * k + y$)

Exercice 14 (4 points)
On rappelle que $\{P\}$**S**$\{Q\}$ est défini par l'implication $O \Rightarrow WP(S)(Q)$. Pour chaque point énuméré ci-dessous, montrer que la propriété $\{P\}$**S**$\{Q\}$ est valide ou pas en utilisant la définition suivante:

$\{P\}$**S**$\{Q\}$ = P $\Rightarrow$ WP(S)(Q)

1. $\{x \leq y\}$**X:=Y-8;Y:=Y+5**$\{2 * x \leq 6 * y\}$
2. $\{x > y\}$**IF X $\neq$ Y THEN X:= 1 ELSE X:= 0 FI**$\{x = 17 \wedge y = 45\}$
3. $\{x > y\}$**IF X $\neq$ Y THEN X:= 1 ELSE X:= 0 FI**$\{x = 7 \wedge y = 0\}$
4. $\{x > y\}$**IF X $\neq$ Y THEN X:= 1 ELSE X:= 0 FI**$\{x > y\}$

Exercice 15 (4 points))
Soient deux fonctions C power2 et p qui satisfont les contrats ci-dessous. Les deux fonctions calculent la même valeur pour un entier donné positif n. La fonction check est incomplète. Compléter la fonction check de manière à ce que l'utilisation de frama-c permette de conclure de l'équivalence des deux fonctions. Exliquer votre idée.

Listing 5: schema de contrat

```
#include <limits.h>
/*@ axiomatic auxmath {
  @ axiom  rule1: \forall int n; n >0 ==> n*n == (n-1)*(n-1)+2*n+1;
  @ } */
/*@  requires  0 <= x;
     ensures \result == x*x;
*/
int power2(int x);

/*@  requires  0 <= x;
     ensures \result == x*x;
*/
int p(int x);

/*@  ensures \result == ???;
*/

int check(int n){
  int r1,r2,r;
  r1 = power2(n);
  r2 = p(n);
  if (r1 == r2)
    { r = ???;
    }
  else
    { r = ???;
    };
  return r;
 }
```

Notations pour WP

La définition structurelle des transformateurs de prédicats est rappelée dans le tableau ci-dessous:

S	wp(S)(P)
X:=E(X,D)	$P[e(x,d)/x]$
SKIP	P
$\mathrm{S}_1;\mathrm{S}_2$	$wp(\mathrm{S}_1)(wp(\mathrm{S}_2)(P))$
IF B S_1 ELSE S_2 FI	$(B \Rightarrow wp(S_1)(P)) \wedge (\neg B \Rightarrow wp(S_2)(P))$

Axiomes et règles d'inférence de la Loique de Hoare

- Axiome d'affectation: $\{P(e/x)\}$**X:=E(X)**$\{P\}$.
- Axiome du saut: $\{P\}$**skip**$\{P\}$.
- Règle de composition : Si $\{P\}\mathbf{S}_1\{R\}$ et $\{R\}\mathbf{S}_2\{Q\}$, alors $\{P\}\mathbf{S}_1;\mathbf{S}_2\{Q\}$.
- Si $\{P \wedge B\}\mathbf{S}_1\{Q\}$ et $\{P \wedge \neg B\}\mathbf{S}_2\{Q\}$, alors $\{P\}$**if B then** $\mathbf{S}_1$ **then** $\mathbf{S}_2$ **fi**$\{Q\}$.
- Si $\{P \wedge B\}\mathbf{S}\{P\}$, alors $\{P\}$**while B do S od**$\{P \wedge \neg B\}$.
- Règle de renforcement/affaiblissement: Si $P' \Rightarrow P$, $\{P\}\mathbf{S}\{Q\}$, $Q \Rightarrow Q'$, alors $\{P'\}\mathbf{S}\{Q'\}$.

Fin de l'énoncé

ex23-7.tex

Exercice 16 (3 points))
Soit le petit programme suivant annoté. Les deux constantes entières p et q ne sont pas connues.

Listing 6: exercice 16

```
/*@ requires x ==  3*q && y == 5*q && x+z == p;
  ensures \result == 2*p-q;
*/

int q2(int x,int y, int z,int p, int q){
    /*@ assert 5*x ==  3*y && x+z == p  ;  */
  x = y+z;
  /*@  assert  x == 2*q+p; */
  y = x + z;
  /*@  assert  y == 2*p-q; */
  return y;
  }
```

La section 3 rappelle les éléments liés à la vérification d'un contrat.

Question 16.1 (1 point))
Le listing 6 décrit un contrat avec un code associé. Enoncer et simplifier les trois conditions de correction de l'annotation du listing 6.

Question 16.2 (2 points)) Vérifiez la validité des conditions et conclure sur la validité ou non de ce contrat.

ex23-6.tex

Exercice 17 (3 points))
Soit le petit programme suivant annoté mais incomplet. Les valeurs des deux constantes entières p et q ne sont pas connues.

Listing 7: exercice 17

```
#define p ??
#define q ??


/*@ requires x == 3*y && z == 6*y;
  ensures \result == q*\at(y,Pre);
*/

int q1(int x,int y, int z){
  /*@ assert x +y +z  ==  p * \at(y,Pre);  */
  y = 2*x+z;
  /*@  assert  y == q*\at(y,Pre);       */
  return y;
  }
```

La section 3 donne les éléments pour vérifier le contrat.

Question 17.1 (1 point))
Le listing 7 décrit un contrat avec un code associé. Enoncer et simplifier les trois conditions de correction de l'annotation du listing 7.

Question 17.2 (2 points)) Proposer un jeu de deux valeurs pour p et q, afin que les conditions de vérification soient correctes.

ex23-8.tex

Exercice 18 (4 points)
Le rappel de la section 4 donne des éléments pour cette question. Pour chaque point énuméré ci-dessous, montrer que la propriété $\{P\}$**S**$\{Q\}$ est valide ou pas en utilisant la définition suivante:

$$\{P\}\mathbf{S}\{Q\} = \text{P} \Rightarrow \text{WP(S)(Q)}$$

1. $\{x + y = 5\}$**X:=Y+2*X**$\{x + y = 10\}$
2. $\{0 \leq y\}$**X:=Y-8;Y:=Y+5**$\{2 * x \leq 6 * y\}$
3. $\{x \leq y\}$**X:=Y-8;Y:=Y+5**$\{2 * x \leq 2 * y\}$
4. $\{x > y \wedge y \geq 6\}$**IF X $\neq$ Y THEN X:= 1 ELSE X:= 0 FI**$\{x + y \geq 4\}$

ex23-10.tex

Exercice 19 (4 points))
Soit le petit programme suivant annoté mais incomplet.

```
/*@ requires z == exp1 ;
  ensures \result == exp2;
*/

int q2(int x,int y, int z){

  /*@ assert B(x,y,z)   ;  */
  x = y+1;
  /*@  assert  A(x,y,z);  */
  y = x + z;
  /*@  assert  y == 3 + x ;  */
  return y;
  }
```

En utilisant l'opérateur wp, proposer des assertions pour $A(x, y, z)$ et $B(x, y, z)$ et des valeurs pour les expressions expr1 et expr2, afin que le contrat soit correct.

ex23-11.tex

Exercice 20 (6 points)
Nous proposons d'analyser un algorithme dont les annotations sont partielles.

PRECONDITION $\left(x0 \in NAT \right)$
POSTCONDITION $\left(zf = x^3 \right)$
VARIABLES $int\ x, z, v, w, t, u;$
$l0:$
$x = x0; u = 0; z = 0; v = 0; w = 1; t = 3;$
$l1 : x = x0 \wedge w = 3*u+1 \wedge v = 3*u^2 \wedge z+v+w = (u+1)^3 \wedge v+t = 3*(u+1)^2 \wedge t = 3*(2*u+1) \wedge u = 0$
WHILE $(u < x)$
$l2:$
$z = z+v+w;$
$l3 : \left(\begin{array}{l} x = x0 \wedge w = 3*u+1 \wedge v = 3*u^2 \wedge z = (u+1)^3 \wedge v+t = 3*(u+1)^2 \\ \wedge t = 3*(2*u+1) \wedge \left(0 \leq u \wedge u < x \right) \end{array} \right)$
$v = v+t;$
$l4 : \left(\begin{array}{l} x = x0 \wedge w = 3*u+1 \wedge v = 3*(u+1)^2 \wedge z = (u+1)^3 \wedge v = 3*(u+1)^2 \\ \wedge t = 3*(2*u+1) \wedge \left(0 \leq u \wedge u < x \right) \end{array} \right)$
$t = t+6;$
$l5 : \left(\begin{array}{l} x = x0 \wedge w = 3*u+1 \wedge v = 3*(u+1)^2 \wedge z = (u+1)^3 \wedge v = 3*(u+1)^2 \\ \wedge t-6 = 3*(2*u+1) \wedge \left(0 \leq u \wedge u < x \right) \end{array} \right)$
$w = w+3;$
$l6:$
$u = u+1;$
$l7 : \left(\begin{array}{l} x = x0 \wedge w = 3*u+1 \wedge v = 3*u^2 \wedge z+v+w = (u+1)^3 \wedge v+t = 3*(u+1)^2 \\ \wedge z = u^3 \wedge t = 3*(2*u+1) \wedge 0 \leq u \wedge u \leq x \end{array} \right)$
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.

$pc \in L$
$z, v, w, t, u \in \mathbb{Z}$
$pc = "l0" \Rightarrow$
$pc = "l1" \Rightarrow \left(\begin{array}{l} x = x0 \wedge w = 3*u+1v = 3*u^2 \wedge z+v+w = (u+1)^3 \wedge v+t = 3*(u+1)^2 \\ \wedge t = 3*(2*u+1) \wedge u = 0 \end{array} \right)$
$pc = "l2" \Rightarrow$
$pc = "l3" \Rightarrow \left(\begin{array}{l} x = x0 \wedge w = 3*u+1 \wedge v = 3*u^2 \wedge z = (u+1)^3 \wedge v+t = 3*(u+1)^2 \\ \wedge t = 3*(2*u+1) \wedge 0 \leq u \wedge u < x \end{array} \right)$
$pc = "l4" \Rightarrow \left(\begin{array}{l} x = x0 \wedge w = 3*u+1 \wedge v = 3*(u+1)^2 \wedge z = (u+1)^3 \wedge v = 3*(u+1)^2 \\ \wedge t = 3*(2*u+1) \wedge 0 \leq u \wedge u < x \end{array} \right)$
$pc = "l5" \Rightarrow \left(\begin{array}{l} x = x0 \wedge w = 3*u+1 \wedge v = 3*(u+1)^2 \wedge z = (u+1)^3 \wedge v = 3*(u+1)^2 \\ \wedge t-6 = 3*(2*u+1) \wedge 0 \leq u \wedge u < x \end{array} \right)$
$pc = "l6" \Rightarrow \left(\begin{array}{l} x = x0 \wedge w = 3*(u+1)+1 \wedge v = 3*(u+1)^2 \wedge z = (u+1)^3 \wedge v = 3*(u+1)^2 \\ \wedge t-6 = 3*(2*u+1) \wedge 0 \leq u \wedge u < x \end{array} \right)$
$pc = "l7" \Rightarrow \left(\begin{array}{l} x = x0 \wedge w = 3*u+1 \wedge v = 3*u^2 \wedge z+v+w = (u+1)^3 \wedge v+t = 3*(u+1)^2 \\ \wedge z = u^3 \wedge t = 3*(2*u+1) \wedge 0 \leq u \wedge u \leq x \end{array} \right)$
$pc = "l8" \Rightarrow x = x0 \wedge w = 3*x+1 \wedge v = 3*x^2 \wedge z+v+w = (x+1)^3 \wedge v+t = 3*(x+1)^2 \wedge z = x^3$

Question 20.1 Enoncer la précondition et la postcondition de cet algorithme..

Question 20.2 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 20.3 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 20.4 Déduire un algorithme qui calcule la racine cubique entière d'un nombre x c'est-à-dire la valeur a satisfaisant la propriété $a^3 \leq x < (a+1)^3$.