

Cours Modélisation et vérification des systèmes informatiques
Exercices
Modélisation d'algorithmes en PlusCal (II)
par Dominique Méry
18 novembre 2024

Exercice 1 (Vérification de l'annotation de l'algorithme du calcul du maximum d'une liste)
appex5_1.tla

Question 1.1 Ecrire un module TLA^+ contenant une définition PlusCal de cet algorithme.

Question 1.2 Ecrire la propriété à vérifier pour la correction partielle.

Question 1.3 Ecrire la propriété à vérifier pour l'absence d'erreurs à l'exécution.

Vérification **precondition** : $\left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right)$

postcondition : $\left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f) \wedge \\ (\forall j. j \in 0..n-1 \Rightarrow f(j) \leq m) \end{array} \right)$

local variables : $i \in \mathbb{Z}$

```

m := f(0);
i := 1;
while i < n do
  if f(i) > m then
    m := f(i);
  ;
  i++;
;

```

Algorithme 1: Algorithme du maximum d'une liste non annotée

Exercice 2 Exponentiation *appex5_2.tla*

Soit l'algorithme annoté calculant la puissance $z = x_1^{x_2}$.

— *Precondition* : $x_1 \in \mathbb{N} \wedge x_2 \in \mathbb{N}$

— *Postcondition* : $z = x_1^{x_2}$

On suppose que x_1 et x_2 sont des constantes.

Question 2.1 Ecrire un module TLA/TLA^+ permettant de valider les conditions de vérification et, en particulier, de montrer la correction partielle.

Question 2.2 Modifier la machine pour prendre en compte l'absence d'erreurs à l'exécution.

Exercice 3 (*appex5_3.tla*)

On considère l'algorithme suivant :

/* algorithme de calcul du maximum avec une boucle while de l'exercice ?? */

precondition : $\left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right)$

postcondition : $\left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f) \wedge \\ (\forall j \cdot j \in 0..n-1 \Rightarrow f(j) \leq m) \end{array} \right)$

local variables : $i \in \mathbb{Z}$

$\ell_0 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in \mathbb{Z} \wedge i \in \mathbb{Z} \wedge \dots \right\}$

$m := f(0);$

$\ell_1 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in \mathbb{Z} \wedge m = f(0) \right\}$

$i := 1;$

$\ell_2 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i = 1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i-1]) \wedge \\ (\forall j \cdot j \in 0..i-1 \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

while $i < n$ **do**

$\ell_3 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in 1..n-1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i-1]) \wedge \\ (\forall j \cdot j \in 0..i-1 \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

if $f(i) > m$ **then**

$\ell_4 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in 1..n-1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i-1]) \wedge \\ (\forall j \cdot j \in 0..i-1 \Rightarrow f(j) \leq m) \end{array} \right) \wedge \right.$

$f(i) > m \}$

$m := f(i);$

$\ell_5 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in 1..n-1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i]) \wedge \\ (\forall j \cdot j \in 0..i \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

;

$\ell_6 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in \mathbb{Z} \wedge i \in 1..n-1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i]) \wedge \\ (\forall j \cdot j \in 0..i \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

$i++;$

$\ell_7 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in 1..n-1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i-1]) \wedge \\ (\forall j \cdot j \in 0..i-1 \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

;

$\ell_8 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i = n \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f) \wedge \\ (\forall j \cdot j \in 0..n-1 \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

Algorithme 2: Algorithme du maximum d'une liste annoté

```

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 :  $y_1, y_2, y_3 \in \mathbb{Z}$ 

 $\ell_0 : \{y_1, y_2, y_3, z \in \mathbb{Z}\}$ 
 $y_1 := x_1; y_2 := x_2 : y_3 := 1;$ 
 $\ell_1 : \{y_1 = x_1 \wedge y_2 = x_2 \wedge y_3 = 1 \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$ 
 $\ell_{11} : \{y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$ 
while  $y_2 \neq 0$  do
   $\ell_2 : \{y_2 \neq 0 \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$ 
  if  $\text{impair}(y_2)$  then
     $\ell_3 : \{\text{impair}(y_2) \wedge y_2 \neq 0 \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$ 
     $y_2 := y_2 - 1;$ 
     $\ell_4 : \{y_2 \geq 0 \wedge \text{pair}(y_2) \wedge y_3 \cdot y_1 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$ 
     $y_3 := y_3 \cdot y_1;$ 
     $\ell_5 : \{y_2 \geq 0 \wedge \text{pair}(y_2) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$ 
  ;
   $\ell_6 : \{y_2 \geq 0 \wedge \text{pair}(y_2) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$ 
   $y_1 := y_1 \cdot y_1;$ 
   $\ell_7 : \{y_2 \geq 0 \wedge \text{pair}(y_2) \wedge y_3 \cdot y_1^{y_2 \text{ div } 2} = x_1^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$ 
   $y_2 := y_2 \text{ div } 2;$ 
   $\ell_8 : \{y_2 \geq 0 \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$ 
;
 $\ell_9 : \{y_2 = 0 \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z}\}$ 
 $z := y_3;$ 
 $\ell_{10} : \{y_2 = 0 \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_1, y_2, y_3 \in \mathbb{N} \wedge z \in \mathbb{Z} \wedge z = x_1^{x_2}\}$ 

```

Algorithme 3: Version solution annotée

```

START
{ $x_1 \geq 0 \wedge x_2 > 0$ }
 $(y_1, y_2, y_3) \leftarrow (x_1, 0, x_2)$ ;
while  $y_3 \leq y_1$  do  $y_3 \leftarrow 2y_3$ ;
while  $y_3 \neq x_2$  do
  begin  $(y_2, y_3) \leftarrow (2y_2, y_3/2)$ ;
  end; if  $y_3 \leq y_1$  do  $(y_1, y_2) \leftarrow (y_1 - y_3, y_2 + 1)$ 
 $(z_1, z_2) \leftarrow (y_1, y_2)$ 
{ $0 \leq z_1 < x_2 \wedge x_1 = z_2x_2 + z_1$ }
HALT

```

Question 3.1 Montrer que cet algorithme est aptriellement correct par rapport à sa précondition et à sa postcondition qu'il faudra énoncer. Pour cela, on traduira cet algorithme sous forme d'un module à partir du langage PlusCal.

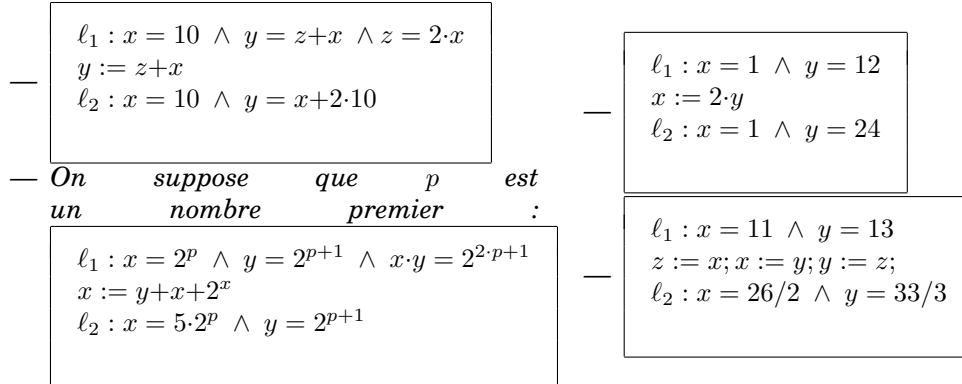
Question 3.2 Montrer qu'il est sans erreur à l'exécution.

Exercice 4 annotation

Montrer que chaque annotation est correcte ou incorrecte selon les conditions de vérifications énoncées comme suit

$\forall x, y, x', y'. P_\ell(x, y) \wedge \text{cond}_{\ell, \ell'}(x, y) \wedge (x', y') = f_{\ell, \ell'}(x, y) \Rightarrow P_{\ell'}(x', y')$

Pour cela, on utilisera une machine et un ciontexte Event-B.



Exercice 5 (Vérification de l'annotation de l'algorithme du calcul du maximum d'une liste)
 Vérifier l'annotation de l'algorithme de calcul du maximum d'une liste 5. On se donne l'annotation et on demande de construire une machine permettant de vérifier cette annotation.

Exercice 6 (Annotation du calcul de la racine carrée entière appex5_6.tla)

L'algorithme annoté ?? calcule la racine carrée entière d'un nombre entier. Vérifier les annotations par un mdodèle Event-B.

Vérification **precondition** : $\left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0 .. n-1 \rightarrow \mathbb{N} \end{array} \right)$

postcondition : $\left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f) \wedge \\ (\forall j. j \in 0 .. n-1 \Rightarrow f(j) \leq m) \end{array} \right)$

local variables : $i \in \mathbb{Z}$

$m := f(0);$

$i := 1;$

while $i < n$ **do**

if $f(i) > m$ **then**

$m := f(i);$

 ;

$i++;$

;

Algorithme 4: Algorithme du maximum d'une liste non annotée

variables X, Y_1, Y_2, Y_3, Z

requires

$x0 \in \mathbb{N}$

$y10 \in \text{Int}$

$y20 \in \text{Int}$

$y30 \in \text{Int}$

$z0 \in \text{Int}$

ensures

$zf \cdot zf \leq x < (zf+1) \cdot (zf+1)$

$xf = x0$

$zf = y1f$

$y2f = y1f+1$

$y3f = 2 \cdot y1f+1$

begin

$\ell_0 : \{x \in \mathbb{N} \wedge z \in \mathbb{Z} \wedge y1 \in \mathbb{Z} \wedge y2 \in \mathbb{Z} \wedge y3 \in \mathbb{Z}\}$

$(Y_1, Y_2, Y_3) := (0, 1, 1);$

$\ell_1 : \{y2 = (y1+1) \cdot (y1+1) \wedge y3 = 2 \cdot y1+1 \wedge y1 \cdot y1 \leq x\}$

While $(Y_2 \leq X)$

$\ell_2 : \{y2 = (y1+1) \cdot (y1+1) \wedge y3 = 2 \cdot y1+1 \wedge y2 \leq x\}$

$(Y_1, Y_2, Y_3) := (Y_1+1, Y_2+Y_3+2, Y_3+2);$

$\ell_3 : \{y2 = (y1+1) \cdot (y1+1) \wedge y3 = 2 \cdot y1+1 \wedge y1 \cdot y1 \leq x\}$

od;

$\ell_4 : \{y2 = (y1+1) \cdot (y1+1) \wedge y3 = 2 \cdot y1+1 \wedge y1 \cdot y1 \leq x \wedge x < y2\}$

$Z := Y_1;$

$\ell_5 : \{y2 = (y1+1) \cdot (y1+1) \wedge y3 = 2 \cdot y1+1 \wedge y1 \cdot y1 \leq x \wedge x < y2 \wedge z = y1 \wedge z \cdot z \leq x \wedge x < (z+1) \cdot (z+1)\}$

end

Exercice 7 Soient les contrats suivants. Pour chaque contrat, évaluer sa validité avec le calcul des wps.

/* algorithme de calcul du maximum avec une boucle while de l'exercice ?? */

precondition : $\left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right)$

postcondition : $\left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f) \wedge \\ (\forall j. j \in 0..n-1 \Rightarrow f(j) \leq m) \end{array} \right)$

local variables : $i \in \mathbb{Z}$

$\ell_0 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in \mathbb{Z} \wedge i \in \mathbb{Z} \wedge \dots \right\}$

$m := f(0);$

$\ell_1 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in \mathbb{Z} \wedge m = f(0) \right\}$

$i := 1;$

$\ell_2 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i = 1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i-1]) \wedge \\ (\forall j. j \in 0..i-1 \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

while $i < n$ **do**

$\ell_3 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in 1..n-1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i-1]) \wedge \\ (\forall j. j \in 0..i-1 \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

if $f(i) > m$ **then**

$\ell_4 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in 1..n-1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i-1]) \wedge \\ (\forall j. j \in 0..i-1 \Rightarrow f(j) \leq m) \end{array} \right) \wedge \right.$

$f(i) > m \}$

$m := f(i);$

$\ell_5 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in 1..n-1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i]) \wedge \\ (\forall j. j \in 0..i \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

;

$\ell_6 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in \mathbb{Z} \wedge i \in 1..n-1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i]) \wedge \\ (\forall j. j \in 0..i \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

$i++;$

$\ell_7 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in 1..n-1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i-1]) \wedge \\ (\forall j. j \in 0..i-1 \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

;

$\ell_8 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i = n \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f) \wedge \\ (\forall j. j \in 0..n-1 \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

Algorithme 5: Algorithme du maximum d'une liste annoté

Question 7.1

```

requires ...
ensures  $z_f = 100 \wedge x_f + y_f = 12 \wedge x_f + x_0 = 4$ ;
variables  $x, y, z$ 
begin
  /·@assert;·/
   $x = x + 1$ ;
  /·@assert;·/
   $y = x + y + 2$ ;
  /·@assert;·/
   $z = x + y$ ;
  /·@assert;·/
end

```

Question 7.2

```

requires  $x_0, y_0 \in \mathbb{N}$ 
ensures  $z_f = \max(x_0, y_0)$ 
variables  $x, y, z$ 
begin
  /·@assert;·/
  IF  $x < y$  THEN
    /·@assert;·/
     $z := y$ ;
    /·@assert;·/
  ELSE
    /·@assert;·/
     $z := x$ ;
    /·@assert;·/
  FI;
  /·@assert;·/
end

```

Exercice 8 *framac/ex2.c*

On suppose que *val* est une valeur entière. Vérifier l'annotation suivante :

```

int q1() {
  int c = val ;
  l1:  $x == 2$ ;
  int x;
  l2:  $c == 2$ ;
   $x = 3 * c$  ;
  l3:  $x == 6$ ;
  return (0);
}

```

Exercice 9 Vérifier l'annotation suivante :

```

int main()
{
  int a = 42; int b = 37;
  int c = a+b;
  l1:  $b == 37$  ;
   $a -= c$ ;
   $b += a$ ;
  l2:  $b == 0 \ \&\& \ c == 79$ ;
  return (0);
}

```

Exercice 10 Vérifier l'annotation suivante :

```
int main()
{
    int z;
    int a = 4;
    l1: a == 4 ;
    int b = 3;
    l2: b == 3 && a == 4;
    int c = a+b;
    l3: b == 3 && c == 7 && a == 4 ; */
    a += c;
    b += a;
    l4: a == 11 && b == 14 && c == 7 ;
    l5: a +b == 25 ;
    z = a*b;
    l6: a == 11 && b == 14 && c == 7 && z == 154;
    return(0);
}
```

Exercice 11 Vérifier l'annotation suivante :

```
int main()
{
    int a = 4;
    int b = 3;
    int c = a+b;
    a += c;
    b += a;
    l: a == 11 && b == 14 && c == 7 ;
    return(0);
}
```