<

Cours Modélisation et vérification des systèmes informatiques
Exercices
Modélisation d'algorithmes en PlusCal (II)
par Dominique Méry
4 décembre 2025

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** : $\begin{pmatrix} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix}$

postcondition : $\begin{pmatrix} m \in \mathbb{N} \wedge \\ m \in ran(f) \wedge \\ (\forall j \cdot j \in 0 \mathinner{..} n{-}1 \Rightarrow f(j) \leq m) \end{pmatrix}$

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 :* $x_1 \in \mathbb{N} \wedge x_2 \wedge \mathbb{N}$
- *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 : $\begin{pmatrix} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix}$

postcondition : $\begin{pmatrix} m \in \mathbb{N} \wedge \\ m \in ran(f) \wedge \\ (\forall j \cdot j \in 0 \mathinner{..} n{-}1 \Rightarrow f(j) \leq m) \end{pmatrix}$

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

$\ell_0 : \{ \begin{pmatrix} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in \mathbb{Z} \wedge i \in \mathbb{Z} \wedge ...\}$

$m := f(0);$

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

$i := 1;$

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

while $i < n$ **do**

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

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

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

$m := f(i);$

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

;

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

$i{+}{+};$

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

;

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

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

Algorithme 3: Version solution annotée

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

Question 3.1 *Montrer que cet algorithme est aprtiellement 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
$\forall x, y, , x', y'.P_\ell(x, y) \wedge cond_{\ell,\ell'}(x, y) \wedge (x', y') = f_{\ell,\ell'}(x, y) \Rightarrow P_{\ell'}(x', y')$
Pour cela, on utlisera une machine et un ciontexte Event-B.

— $\ell_1 : x = 10 \wedge y = z{+}x \wedge z = 2{\cdot}x$
  $y := z{+}x$
  $\ell_2 : x = 10 \wedge y = x{+}2{\cdot}10$

— $\ell_1 : x = 1 \wedge y = 12$
  $x := 2{\cdot}y$
  $\ell_2 : x = 1 \wedge y = 24$

— *On suppose que p est un nombre premier :*
  $\ell_1 : x = 2^p \wedge y = 2^{p+1} \wedge x{\cdot}y = 2^{2 \cdot p+1}$
  $x := y{+}x{+}2^x$
  $\ell_2 : x = 5{\cdot}2^p \wedge y = 2^{p+1}$

— $\ell_1 : x = 11 \wedge y = 13$
  $z := x; x := y; y := z;$
  $\ell_2 : x = 26/2 \wedge y = 33/3$

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

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

```
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

- $x0 \in \mathbb{N}$
- $y10 \in Int$
- $y20 \in Int$
- $y30 \in Int$
- $z0 \in Int$

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 : $\begin{pmatrix} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0 \,..\, n{-}1 \rightarrow \mathbb{N} \end{pmatrix}$

postcondition : $\begin{pmatrix} m \in \mathbb{N} \wedge \\ m \in ran(f) \wedge \\ (\forall j \cdot j \in 0 \,..\, n{-}1 \Rightarrow f(j) \leq m) \end{pmatrix}$

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

$\ell_0 : \{ \begin{pmatrix} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0 \,..\, n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in \mathbb{Z} \wedge i \in \mathbb{Z} \wedge ...\}$

$m := f(0);$

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

$i := 1;$

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

while $i < n$ **do**

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

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

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

$m := f(i);$

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

;

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

$i{+}{+};$

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

;

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

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
$/\cdot @assert; \cdot/$
$x = x+1;$
$/\cdot @assert; \cdot/$
$y = x+y+2;$
$/\cdot @assert; \cdot/$
$z = x+y;$
$/\cdot @assert; \cdot/$
end

La version ACSL est la suivante

Listing 1 – td51.c

```
struct data {
    unsigned x;
    unsigned y;
    unsigned z;
};

/*@
  @ ensures \result.z == 100 && \result.x+\result.y == 12 && \result.x + x0==4;
  */


struct data  exemple(int x0,int y0,int z0)
{
  int x=x0;
  int y=y0;
  int z=z0;
//@  assert  x == x0;
  x = x + 1;
  y=x+y+2;                                           \
  z = x +y;
  struct data r;
  r.x = x;r.y=y;r.z=z;
  return r;
}
```

Question 7.2

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

La version ACSL est la suivante

Listing 2 – td52.c

```
/*@ requires x0 >= 0 && y0 >= 0;
  @ ensures (\result == x0 || \result == y0)  && \result >= x0 && \result >= y0;
  */

 exemple(int x0,int y0)
{
  int x=x0;
  int y=y0;
  int z;
//@  assert  x == x0  && y == y0;
  if ( x < y )
    {
    z = y;
    }
  else
    {
    z=x;
    };
   return z;
}
```

Exercice 8 *td58.c*
On suppose que val est une valeur entière. Vérifier l'annotation suivante :

Listing 3 – td51.c

```
#define v 3
/*@ requires  val == v;
 */

int  exemple(int val)
{
  int c = val ;
  //@ assert  c == 2;
  int x;
  //@ assert  c == 2;
  x = 3 * c ;
```

```
  //@ assert x == 6;
  return(0);
}
```

Exercice 9 *td59.c*
Vérifier l'annotation suivante :

Listing 4 – td59.c

```
int  exemple()
{
  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 :*

Listing 5 – td510.c

```
int main()
{
  int z;
  int a = 4;
  //@ assert a == 4 ;
  int b = 3;
  //@ assert  b == 3 && a == 4;
  int c = a+b;
  //@ assert b == 3 && c == 7 && a == 4 ;
  a += c;
  b += a;
  //@ assert a == 11 && b == 14 && c == 7 ;
  //@ assert a +b  == 25 ;
  z = a*b;
  //@ assert  a == 11 && b == 14 && c == 7 && z == 154;
  return(0);
}
```

Exercice 11 *Vérifier l'annotation suivante :*

Listing 6 – td511.c

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