Cours Modélisation et vérification des systèmes informatiques
Exercices (avec les corrections)
Modélisation d'algorithmes en TLA$^+$
Annotation, modélisation et vérification
par Dominique Méry
1er octobre 2021

annotation

Exercice 1 ☑
Montrer que chaque annotation est correcte ou incorrecte selon les conditions de vérifications énoncées comme suit :
$\forall v, v'.P_\ell(v) \ \wedge \ cond_{\ell,\ell'}(v) \ \wedge \ v' = f_{\ell,\ell'}(v) \Rightarrow P_{\ell'}(v')$
Cette condition s'écrit initialement :
$\forall v, v', pc, pc'.pc = \ell \wedge P_\ell(v) \ \wedge \ pc = \ell \wedge \ cond_{\ell,\ell'}(v) \ \wedge \ v' = f_{\ell,\ell'}(v) \wedge \ pc' = \ell' \Rightarrow pc' = \ell' \wedge \ P_{\ell'}(v')$
mais on peut réduire en oubliant la variable pc.

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

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

— *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 2 ☑
Montrer que chaque annotation est correcte ou incorrecte selon les conditions de vérifications énoncées comme suit
$\forall x, y, , x', y'.P_\ell(x, y) \ \wedge \ cond_{\ell,\ell'}(x, y) \ \wedge \ (x', y') = f_{\ell,\ell'}(x, y) \Rightarrow P_{\ell'}(x', y')$

— *(1)* $\ell_1 : x = 9 \ \wedge \ y = z{+}x$
$y := x{+}9$
$\ell_2 : x = 9 \ \wedge \ y = x{+}9$

— $\ell_1 : x = 3 \ \wedge \ y = 3$
$x := y{+}x$
$\ell_2 : x = 6 \ \wedge \ y = 3$

— *(2)* $\ell_1 : x = 1 \ \wedge \ y = 3 \ \wedge \ x{+}y = 12$
$x := y{+}x$
$\ell_2 : x = 567 \ \wedge \ y = 34$

— $\ell_1 : x = 1 \ \wedge \ y = 3$
$z := x; x := y; y := z;$
$\ell_2 : x = 3 \ \wedge \ y = 1$

1. $c = \ell_1 \wedge x = 9 \ \wedge \ y = z{+}x \ \wedge \ \textbf{TRUE} \ \wedge \ (x', y', c') = (x, x{+}9, \ell_2) \Rightarrow c' = \ell_2 \wedge x' = 9 \ \wedge \ y' = x'{+}9$ *:*
   (a) $c = \ell_1 \wedge x = 9 \ \wedge \ y = z{+}x \ \wedge c' = \ell_2 \Rightarrow c' = \ell_2 \wedge x = 9 \ \wedge \ x{+}9 = x{+}9$
   (b) $c = \ell_1 \wedge x = 9 \ \wedge \ y = z{+}x \ \wedge c' = \ell_2 \Rightarrow x = 9 \ \wedge \ x{+}9 = x{+}9$
   (c) *CORRECT*
2. $c = \ell_1 \wedge x = 1 \ \wedge \ y = 3 \ \wedge \ x{+}y = 12 \ \wedge \textbf{TRUE} \ \wedge \ (x', y', c') = (y{+}x, y, \ell_2) \Rightarrow c' = \ell_2 \wedge x' = 567 \ \wedge \ y' = 34$ *:*
   (a) $c = \ell_1 \wedge x = 1 \ \wedge \ y = 3 \ \wedge \ x{+}y = 12 \ \ \wedge \ (x', y', c') = (y{+}x, y, \ell_2) \Rightarrow c' = \ell_2 \wedge y{+}x = 567 \ \wedge \ y = 34$

(b) $c = \ell_1 \wedge x = 1 \;\wedge\; y = 3 \;\wedge\; x{+}y = 12 \;\;\wedge\; c' = \ell_2 \Rightarrow c' = \ell_2 \wedge y{+}x = 567 \;\wedge\; y = 34$

(c) $c = \ell_1 \wedge x = 1 \;\wedge\; y = 3 \;\wedge\; x{+}y = 12 \;\;\wedge\; c' = \ell_2 \Rightarrow x{+}y = 4 \wedge x{+}y = 12$

(d) $c = \ell_1 \wedge x = 1 \;\wedge\; y = 3 \;\wedge\; x{+}y = 12 \;\;\wedge\; c' = \ell_2 \Rightarrow$ **FALSE**

(e) **FALSE** $\Rightarrow c' = \ell_2 \wedge y{+}x = 567 \;\wedge\; y = 34$

(f) *CORRECT*

3. $c = \ell_1 \wedge x = 3 \;\wedge\; y = 3 \wedge$ **TRUE** $\wedge\; (x', y', c') = (y{+}x, y, \ell_2) \Rightarrow c' = \ell_2 \wedge x' = 6 \;\wedge\; y' = 3$

   (a) $c = \ell_1 \wedge x = 3 \;\wedge\; y = 3 \;\wedge\; c' = \ell_2 \Rightarrow c' = \ell_2 \wedge y{+}x = 6 \;\wedge\; y = 3$

   (b) $c = \ell_1 \wedge x = 3 \;\wedge\; y = 3 \;\wedge\; c' = \ell_2 \Rightarrow c' = \ell_2 \wedge y{+}x = 6 \;\wedge\; y = 3$

   (c) $c = \ell_1 \wedge x = 3 \;\wedge\; y = 3 \;\wedge\; c' = \ell_2 \Rightarrow c' = \ell_2 \wedge 6 = 6 \;\wedge\; y = 3$

   (d) *CORRECT*

4. $c = \ell_1 \wedge x = 1 \;\wedge\; y = 3$ **TRUE** $\wedge\; (x', y', z', c') = (y, x, x, \ell_2) \Rightarrow c' = \ell_2 \wedge x' = 3 \;\wedge\; y' = 1$

   (a) $c = \ell_1 \wedge x = 1 \;\wedge\; y = 3$ **TRUE** $\wedge\; (x', y', z', c') = (y, x, x, \ell_2) \Rightarrow c' = \ell_2 \wedge y = 3 \;\wedge\; x = 1$

   (b) *CORRECT*

Exercice 3 ☑

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

$\forall x, y, , x', y'. P_\ell(x, y) \;\wedge\; cond_{\ell,\ell'}(x, y) \;\wedge\; (x', y') = f_{\ell,\ell'}(x, y) \Rightarrow P_{\ell'}(x', y')$

—

$\ell_1 : x = 3 \;\wedge\; y = z{+}x \;\wedge z = 2{\cdot}x$
$y := z{+}x$
$\ell_2 : x = 3 \;\wedge\; y = x{+}6$

—

$\ell_1 : x = 1 \;\wedge\; y = 12$
$x := 2{\cdot}y{+}x$
$\ell_2 : x = 1 \;\wedge\; y = 25$

—

$\ell_1 : x = 2^4 \;\wedge\; y = 2^{345} \;\wedge\; x{\cdot}y = 2^{350}$
$x := y{+}x{+}2^x$
$\ell_2 : x = 2^{56} \;\wedge\; y = 2^{345}$

Exercice 4 ☑

Soit le petit algorithme annoté suivant :

$l0 : \{v = 3\}$
$v := v{+}2$
$l1 : \{v = 5\}$

Ecrire un module TLA$^+$ explicitant la relation de transition, les conditions initiales, l'invariant et la propriété de sûreté pour la correction partielle.

◇— **Solution de l'exercice 4**

MODULE an_1 ------

EXTENDS *Integers*, *TLC*

VARIABLES v, pc

$titi \triangleq pc = \text{"l0"} \wedge v = 3$

$skip \triangleq \text{UNCHANGED } \langle pc, v \rangle$
$skip_2 \triangleq pc' = pc \land v' = v$

$trans \triangleq pc = \text{"l0"} \land TRUE \land pc' = \text{"l1"} \land v' = v+2$
$trans_2 \triangleq pc = \text{"l0"} \land pc' = \text{"l1"} \land v' = v+2$
$trans_3 \triangleq$
$\quad \land pc = \text{"l0"} \land TRUE$
$\quad \land pc' = \text{"l1"}$
$\quad \land v' = v+2$

$toto \triangleq skip \lor trans$

$i \triangleq$
$\quad \land pc \in \{\text{"l0"}, \text{"l1"}\}$
$\quad \land pc = \text{"l0"} \Rightarrow v = 3$
$\quad \land pc = \text{"l1"} \Rightarrow v = 6$

$safety \triangleq pc = \text{"l1"} \Rightarrow v = 5$

Fin 4

Exercice 5 ☑
Définir les conditions de vérification de la correction partielle pour les structures suivantes. Définir un modèle TLA$^+$ pour vérifier la bonne annotation.

Question 5.1

$\ell 1 : \{P_{\ell 1}(x, y)\}$
$\quad x := x+y+7;$
$\ell 2 : \{P_{\ell 2}(x, y)\}$

⋄– **Solution de la question 5.1**

Conditions de vérification pour la correction partielle *pc désigne la variable de contrôle.*

$$pc = \ell 1 \land P_{\ell 1}(x, y) \land pc = \ell 1 \land pc' = \ell 2 \land x' = x+y+7 \land y' = y \Rightarrow pc' = \ell 2 \land P_{\ell 2}(x', y')$$

qui se simplifie en :

$$P_{\ell 1}(x, y) \land x' = x+y+7 \land y' = y \Rightarrow P_{\ell 2}(x', y')$$

qui se simplifie en :

$$P_{\ell 1}(x, y) \land x' = x+y+7 \Rightarrow P_{\ell 2}(x', y)$$

qui se simplifie en :

$$P_{\ell 1}(x, y) \Rightarrow P_{\ell 2}(x+y+7, y)$$

MODULE $an2$

Modèle TLA$^+$ pour vérifier la bonne annotation EXTENDS *Naturals*

VARIABLES x, y, pc

Define actions from the text of annotated algorithm

$al0l_1 \triangleq$

$\quad \land\ pc = \text{"l0"}$

$\quad \land\ pc' = \text{"l1"}$

$\quad \land\ x' = x+y+7$

$\quad \land\ y' = y$

Define the computation relation

$next \triangleq al0l1$

Define the initial conditions

$init \triangleq pc = \text{"l0"} \land x = 3 \land y = 8$

Define the invariant from the annotation

$i \triangleq$

$\quad \land\ pc = \text{"l0"} \Rightarrow x = 3 \land y = 8$

$\quad \land\ pc = \text{"l1"} \Rightarrow x = 6 \land y = 89$

Define the safety property to check namely the partial correctness

$safe \triangleq pc = \text{"l1"} \Rightarrow x = 7 \land y = 89$

Modification History

Last modified Tue Dec 15 17 :30 :19 CET 2015 by mery

Created Wed Sep 09 17 :02 :47 CEST 2015 by mery

MODULE $an2$

EXTENDS *Naturals*

VARIABLES x, y, pc

Define actions from the text of annotated algorithm

$al0l_1 \triangleq$

$\quad \land\ pc = \text{"l0"}$

$\quad \land\ pc' = \text{"l1"}$

$\quad \land\ x' = x+y+7$

$\quad \land\ y' = y$

Define the computation relation

$next \triangleq al0l1$

Define the initial conditions

$init \triangleq pc = \text{"l0"} \land x = 3 \land y = 8$

Define the invariant from the annotation

$i \triangleq$

$\wedge\, pc = \text{"l0"} \Rightarrow x = 3 \wedge y = 8$
$\wedge\, pc = \text{"l1"} \Rightarrow x = 18 \wedge y = 8$

Define the safety property to check namely the partial correctness

$safe \triangleq pc = \text{"l1"} \Rightarrow x = 18 \wedge y = 8$

$prop \triangleq i \Rightarrow safe$
$Init \triangleq init$
$Next \triangleq next$
$principe \triangleq init \Rightarrow \wedge\, prop$

Modification History

Last modified Wed Sep 21 13 :28 :17 CEST 2016 by mery

Created Wed Sep 09 17 :02 :47 CEST 2015 by mery

Fin 5.1

Question 5.2

$$\begin{array}{l} \ell : \{P_\ell(x,y)\} \\ \quad x, y := y, x; \\ \ell' : \{P_{\ell'}(x,y)\} \end{array}$$

◇— Solution de la question 5.2

Conditions de vérification pour la correction partielle *c désigne la variable de contrôle.*

$$c = \ell 1 \wedge P_{\ell 1}(x,y) \wedge c' = \ell 2 \wedge (x', y') = (y, x) \Rightarrow c' = \ell 2 \wedge P_{\ell 2}(x', y')$$

qui se simplifie en :

$$P_{\ell 1}(x,y) \wedge x' = y \wedge y' = x \Rightarrow P_{\ell 2}(x', y')$$

qui se simplifie en :

$$P_{\ell 1}(x,y) \Rightarrow P_{\ell 2}(y, x)$$

MODULE $an3$

Modèle TLA$^+$ pour vérifier la bonne annotation EXTENDS *Naturals*
CONSTANTS a, b
VARIABLES x, y, pc

Define actions from the text of annotated algorithm

$al1l_2 \triangleq$
$\wedge\, pc = \text{"l1"}$
$\wedge\, pc' = \text{"l2"}$
$\wedge\, x' = y \wedge y' = x$

$newaction \triangleq pc = \text{"l2"} \wedge pc' = \text{"l1"} \wedge x' = x \wedge y' = y$

Define the computation relation

$next \triangleq al1l2$

$newnext \triangleq al1l_2 \vee newaction$

Define the initial conditions

$init \triangleq pc = \text{"l1"} \wedge x = a \wedge y = b$

Define the invariant from the annotation

$i \triangleq$

$\quad \wedge pc = \text{"l1"} \Rightarrow x = a \wedge y = b$
$\quad \wedge pc = \text{"l2"} \Rightarrow x = b \wedge y = a$

Define the safety property to check namely the partial correctness

$safe \triangleq pc = \text{"l2"} \Rightarrow x = b \wedge y = a$

Fin 5.2

Exercice 6 ☑

Déterminer les conditions de vérification pour la structure de boucle bornée.
On suppose que S ne modifie pas i.

$\ell_1 : \{P_{\ell_1}(x)\}$
  FOR $i := 1$ ***TO*** n ***DO***
    $\ell_2 : \{P_{\ell_2}(i,x)\}$
      $S(x);$
    $\ell_3 : \{P_{\ell_3}(i,x)\}$
  ENDFOR
$\ell_4 : \{P_{\ell_4}(x)\}$

◇— **Solution de la question 6.0**

(1) $c = \ell_1 \wedge P_{\ell_1}(x) \wedge 1 \leq n \wedge c' = \ell_2 \wedge i' = 1 \wedge x' = x \Rightarrow c' = \ell_2 \wedge P_{\ell_2}(i', x')$
(2) $c = \ell_1 \wedge P_{\ell_1}(x) \wedge \neg(1 \leq n) \wedge c' = \ell_4 \wedge x' = x \Rightarrow c' = \ell_4 \wedge P_{\ell_4}(x')$
(3) $c = \ell_3 \wedge P_{\ell_3}(x,i) \wedge i{+}1 \leq n \wedge c' = \ell_2 \wedge i' = i{+}1 \wedge x' = x \Rightarrow c' = \ell_2 \wedge P_{\ell_2}(i', x')$
(4) $c = \ell_2 \wedge P_{\ell_3}(x,i) \wedge \neg(i{+}1 \leq n) \wedge c' = \ell_4 \wedge x' = x \wedge i' = i{+}1 \Rightarrow c' = \ell_4 \wedge P_{\ell_4}(x')$

Fin 6.0

Exercice 7 ☑

Question 7.1 *Compléter l'algorithme 7 en l'annotant.*

◇— **Solution de la question 7.1**

Annotation *L'annotation de cet algorithme est donnée à la référence d'algorithme 7 et la figure est placée au gré de LaTeX.*

Fin 7.1

Question 7.2 *Vérifier la bonne annotation*

◇— **Solution de la question 7.2**

Modèle TLA$^+$ pour vérifier la bonne annotation

```
-------------------------------- MODULE appex3_77 --------------------------------
EXTENDS Naturals, Integers
CONSTANTS x0,y0,z0
```

Variables : X,Y,Z
Requires : $x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z}$
Ensures : $z_f = max(x_0, y_0)$

$\ell_0 : \{\ldots\}$
if $X < Y$ **then**
  $\ell_1 : \{\ldots\}$
  $Z := Y;$
  $\ell_2 : \{\ldots\}$
else
  $\ell_3 : \{\ldots\}$
  $Z := X;$
  $\ell_4 : \{\ldots\}$
;
$\ell_5 : \{\ldots\}$

Algorithme 1: maximum de deux nombres non annotée

Variables : X,Y,Z
Requires : $x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z}$
Ensures : $z_f = max(x_0, y_0)$

$\ell_0 : \{x = x_0 \wedge y = y_0 \wedge z = z_0 \wedge x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z}\}$
if $X < Y$ **then**
  $\ell_1 : \{x < y \wedge x = x_0 \wedge y = y_0 \wedge z = z_0 \wedge x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z}\}$
  $Z := Y;$
  $\ell_2 : \{x < y \wedge x = x_0 \wedge y = y_0 \wedge x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z} \wedge z = y_0\}$
else
  $\ell_3 : \{x\ geqy \wedge x = x_0 \wedge y = y_0 \wedge z = z_0 \wedge x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z}\}$
  $Z := X;$
  $\ell_4 : \{x \geq y \wedge x = x_0 \wedge y = y_0 \wedge x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z} \wedge z = x_0\}$
;
$\ell_5 : \{z = max(x_0, y_0) \wedge x = x_0 \wedge y = y_0 \wedge x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z}\}$

Algorithme 2: maximum de deux nombres non annotée

```
VARIABLES  x,y,z,pc
ASSUME x0 \in Nat /\ y0 \in Nat
typeInt(u) == u \in Int
maxi(u,v) == IF u < v THEN v ELSE u
pre ==  x0 \in Nat /\ y0 \in Nat /\ z0 \in Int
---------------------------------------------------------------------
al0l1 ==
    /\ pc="l0"
    /\ pc'="l1"
    /\ x<y
    /\ z'=z /\ x'=x /\ y'=y
al1l2 ==
    /\ pc="l1"
    /\ pc'="l2"
    /\ z'=y
    /\ x'=x /\ y'=y
al2l5 ==
    /\ pc="l2"
    /\ pc'="l5"
    /\ z'=z /\ x'=x /\ y'=y
al0l3 ==
    /\ pc="l0"
    /\ pc'="l3"
    /\ x \geq y
    /\ z'=z /\ x'=x /\ y'=y
al3l4 ==
    /\ pc="l3"
    /\ pc'="l4"
    /\ z'=x
    /\ x'=x /\ y'=y
  al4l5 ==
    /\ pc="l4"
    /\ pc'="l5"
    /\ z'=z /\ x'=x /\ y'=y
---------------------
Next == al0l1 \/ al1l2 \/ al2l5  \/ al0l3 \/ al3l4 \/ al4l5 \/ UNCHANGED <<x,y,z,pc
Init == pc="l0" /\ x=x0 /\ y =y0 /\ z = z0
----------------------
i ==
    /\ typeInt(x) /\ typeInt(y) /\ typeInt(z)
    /\ pc="l0" =>  x=x0 /\ y=y0 /\ z=z0 /\ pre
    /\ pc="l1" => x<y /\ x=x0 /\ y=y0 /\ z=z0 /\ pre
    /\ pc="l2" => x<y /\ x=x0 /\ y=y0 /\ z=y0 /\ pre
    /\ pc="l3" => x \geq y /\ x=x0 /\ y=y0 /\ z=z0 /\ pre
    /\ pc="l4" => x \geq y /\ x=x0 /\ y=y0 /\ z=x0  /\ pre
    /\ pc="l5" => z = maxi(x0,y0) /\ x=x0 /\ y=y0  /\ pre
safe ==  pc="l5" =>  z = maxi(x0,y0)
safeab == x=x0 /\ y=y0
=============================================================================
\* Modification History
\* Last modified Wed Sep 29 20:32:22 CEST 2021 by mery
\* Created Wed Sep 09 18:19:08 CEST 2015 by mery
```

Fin 7.2

Question 7.3 *Enoncer et vérifier la correction partielle*

◇— **Solution de la question 7.3**

Il suffit de donner tout d'abord la précondition et la postcondition et de vérifier les conditoons de vérifications de la correction partielle.

Fin 7.3

Exercice 8 ☑

Il s'agit d'étudier et d'annoter le programme proposé en vu d'obtenir sa correction partielle (c'est-à-dire sans la preuve de terminaison). On appelle état un ensemble de valeurs précises (spécifié par un prédicat) des variables du programme, nous allons considérer une étiquette (ℓ) entre chaque instruction du programme considéré. On appelle une annotation le prédicat décrivant les valeurs possibles des variables pour un état du programme. Cette annotation est notée : $P_\ell(v)$ et exprime la propriété satisfaite par la variable v en ℓ.
On vous demande :

1. *de dessiner le graphe de transition entre les étiquettes*
2. *d'annoter toutes les étiquettes du programme*
3. *de proposer un modèle TLA$^+$ pour vérifier les annotations et la correction partielle*

Variables : X
Requires : $x_0 \in \mathbb{N}$
Ensures : $x_f = 0$

$\ell_0 : \{\ldots\}$
while $0 < X$ **do**
  $\ell_1 : \{\ldots\}$
  $X := X-1$;
  $\ell_2 : \{\ldots\}$
;
$\ell_3 : \{\ldots\}$

Algorithme 3: Exemple non annoté

◇— **Solution de l'exercice 8**

Annotation L'annotation (cf algorithme) est construite par propagation des assertions selon les instructions. Il faut ensuite vérifier que les conditions sont vraies.

MODULE $ex3$

Modèle TLA$^+$ pour vérifier la bonne annotation EXTENDS *Naturals*

CONSTANTS $x0$
VARIABLES x, pc

$al0l_1 \triangleq$
$\land\ pc = \text{"l0"}$
$\land\ pc' = \text{"l1"}$
$\land\ 0 < x$

Variables : X
Requires : $x_0 \in \mathbb{N}$
Ensures : $x_f = 0$

$\ell_0 : \{ \, x = x_0 \wedge x_0 \in \mathbb{N}\}$
while $0 < X$ **do**
  $\ell_1 : \{0 < x \leq x_0 \wedge x_0 \in \mathbb{N}\}$
  $X := X{-}1;$
  $\ell_2 : \{0 \leq x \leq x_0 \wedge x_0 \in \mathbb{N}\}$

;
$\ell_3 : \{x = 0\}$

Algorithme 4: exemple annoté

$\wedge \; x' = x$

$al0l_3 \;\triangleq$
$\wedge \; pc = \text{"l0"}$
$\wedge \; pc' = \text{"l3"}$
$\wedge \; x = 0$
$\wedge \; x' = x$

$al1l_2 \;\triangleq$
$\wedge \; pc = \text{"l1"}$
$\wedge \; pc' = \text{"l2"}$
$\wedge \; x' = x{-}1$

$al2l_1 \;\triangleq$
$\wedge \; pc = \text{"l2"}$
$\wedge \; pc' = \text{"l1"}$
$\wedge \; 0 < x$
$\wedge \; x' = x$

$al2l_3 \;\triangleq$
$\wedge \; pc = \text{"l2"}$
$\wedge \; pc' = \text{"l3"}$
$\wedge \; 0 = x$
$\wedge \; x' = x$

$\textit{next} \;\triangleq\; al0l_1 \;\vee\; al0l_3 \;\vee\; al1l_2 \;\vee\; al2l_1 \;\vee\; al2l3$

$\textit{init} \;\triangleq\; pc = \text{"l0"} \;\wedge\; x = x0$

$i \;\triangleq$
$\wedge \; pc = \text{"l0"} \;\Rightarrow\; x = x0$
$\wedge \; pc = \text{"l1"} \;\Rightarrow\; 0 < x \;\wedge\; x \;\leq\; x_0$
$\wedge \; pc = \text{"l2"} \;\Rightarrow\; 0 \;\leq\; x \;\wedge\; x \;\leq\; x0$
$\wedge \; pc = \text{"l3"} \;\Rightarrow\; x = 0$

$\textit{safe} \;\triangleq\; pc = \text{"l3"} \;\Rightarrow\; x = 0$

$\textit{safeplus} \;\triangleq\; x \;\geq\; 0$

Modification History

Last modified Thu Sep 10 09 :35 :48 CEST 2015 by mery

Created Wed Sep 09 18 :07 :50 CEST 2015 by mery

Fin 8

Exercice 9 Question 9.1 *Soit un tableau t (dans $\mathbb{N}$), donner un prédicat max$(m,t,a,b) = \ldots$ exprimant qu'un nombre $m \in \mathbb{N}$ est le maximum de ce tableau t dans l'intervalle $a \,..\, b$.*

◇— **Solution de l'exercice 9**

$$\textbf{\textit{max}}(m,t,a,b) \stackrel{def}{=} m \in \mathrm{ran}(t) \land (\forall i \cdot i \in a \,..\, b \Rightarrow t(i) \leq m)$$

Fin 9

Question 9.2 *De même pour trié(t,a,b), donnez un prédicat spécifiant que le tableau t est trié dans l'intervalle $a \,..\, b$.*

◇— **Solution de l'exercice 9**

$$\textbf{\textit{trié}}(t,a,b) \stackrel{def}{=} \forall i,j \cdot ((i \in a \,..\, b \land j \in a \,..\, b \land i \leq j) \Rightarrow t(i) \leq t(j))$$

Fin 9

Exercice 10 *Dans l'algorithm 10, on calcule le maximum d'une suite de valeurs entières. On vous demande :*

— *Définir la précondition et la postcondition.*
— *Annoter cet algorithme*
— *Vérifier les conditions de vérification pour la correction partielle*
— *Vérifier les conditions pour l'absence d'erreurs à l'exécution*

Variables : F,N,M,I

Requires : $\begin{pmatrix} n_0 \in \mathbb{N} \land \\ n_0 \neq 0 \land \\ f_0 \in 0 \,..\, n_0{-}1 \rightarrow \mathbb{N} \end{pmatrix}$

Ensures : $\begin{pmatrix} m_f \in \mathbb{N} \land \\ m_f \in ran(f_0) \land \\ (\forall j \cdot j \in 0 \,..\, n_0{-}1 \Rightarrow f_0(j) \leq m_f) \end{pmatrix}$

$M := F(0)$;
$I := 1$;
while $I < N$ **do**
  if *F(i)>M* **then**
    $M := F(I)$;
  ;
  $I{+}{+}$;
;

Algorithme 5: Algorithme du maximum d'une liste non annotée

◇— **Solution de l'exercice 10**

La solution de cette annotation est dans l'algorithme annoté.

MODULE *algo_maximum*

computing the maximum value of an array f

/* algorithme de calcul du maximum avec une boucle while de l'exercice 10 */
Variables : F,N,M,I

Requires : $\left(\begin{array}{l} n_0 \in \mathbb{N} \wedge \\ n_0 \neq 0 \wedge \\ f_0 \in 0 \,..\, n_0{-}1 \to \mathbb{N} \end{array}\right)$

Ensures : $\left(\begin{array}{l} m_f \in \mathbb{N} \wedge \\ m_f \in ran(f_0) \wedge \\ (\forall j \cdot j \in 0 \,..\, n_0{-}1 \Rightarrow f_0(j) \leq m_f) \end{array}\right)$

$\ell_0 : \{ \left(\begin{array}{l} n_0 \in \mathbb{N} \wedge \\ n_0 \neq 0 \wedge \\ f_0 \in 0 \,..\, n_0{-}1 \to \mathbb{N} \\ m_0, i_0 \in \mathbb{Z} \end{array}\right) \wedge n = n_\wedge f = f0 \wedge i = i_0 \wedge m = m_0 \}$

$M := F(0);$

$\ell_1 : \{ \left(\begin{array}{l} n_0 \in \mathbb{N} \wedge \\ n_0 \neq 0 \wedge \\ f_0 \in 0 \,..\, n_0{-}1 \to \mathbb{N} \\ m_0, i_0 \in \mathbb{Z} \end{array}\right) \wedge n = n_\wedge f = f0 \wedge i = i_0 \wedge m = f(0) \}$

$I := 1;$

$\ell_2 : \{ \left(\begin{array}{l} n_0 \in \mathbb{N} \wedge \\ n_0 \neq 0 \wedge \\ f_0 \in 0 \,..\, n_0{-}1 \to \mathbb{N} \\ m_0, i_0 \in \mathbb{Z} \\ n = n_0 \wedge f = f0 \end{array}\right) \wedge i = 1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i{-}1]) \wedge \\ (\forall j \cdot j \in 0 \,..\, i{-}1 \Rightarrow f(j) \leq m) \end{array}\right) \}$

while $I < N$ **do**

$\ell_3 : \{ \left(\begin{array}{l} n_0 \in \mathbb{N} \wedge \\ n_0 \neq 0 \wedge \\ f_0 \in 0 \,..\, n_0{-}1 \to \mathbb{N} \\ m_0, i_0 \in \mathbb{Z} \\ n = n_0 \wedge f = f0 \end{array}\right) \wedge i \in 1..n{-}1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i{-}1]) \wedge \\ (\forall j \cdot j \in 0 \,..\, i{-}1 \Rightarrow f(j) \leq m) \end{array}\right) \}$

if ***F(I)>M*** **then**

$\ell_4 : \{ \left(\begin{array}{l} n_0 \in \mathbb{N} \wedge \\ n_0 \neq 0 \wedge \\ f_0 \in 0 \,..\, n_0{-}1 \to \mathbb{N} \\ m_0, i_0 \in \mathbb{Z} \\ n = n_0 \wedge f = f0 \end{array}\right) \wedge i \in 1..n{-}1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i{-}1]) \wedge \\ (\forall j \cdot j \in 0 \,..\, i{-}1 \Rightarrow f(j) \leq m) \end{array}\right) \wedge$
$f(i) > m \}$

$M := F(I);$

$\ell_5 : \{ \left(\begin{array}{l} n_0 \in \mathbb{N} \wedge \\ n_0 \neq 0 \wedge \\ f_0 \in 0 \,..\, n_0{-}1 \to \mathbb{N} \\ m_0, i_0 \in \mathbb{Z} \\ n = n_0 \wedge f = f0 \end{array}\right) \wedge i \in 1..n{-}1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i]) \wedge \\ (\forall j \cdot j \in 0 \,..\, i \Rightarrow f(j) \leq m) \end{array}\right) \}$

;

$\ell_6 : \{ \left(\begin{array}{l} n_0 \in \mathbb{N} \wedge \\ n_0 \neq 0 \wedge \\ f_0 \in 0 \,..\, n_0{-}1 \to \mathbb{N} \\ m_0, i_0 \in \mathbb{Z} \\ n = n_0 \wedge f = f0 \end{array}\right) \wedge i \in 1..n{-}1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i]) \wedge \\ (\forall j \cdot j \in 0 \,..\, i \Rightarrow f(j) \leq m) \end{array}\right) \}$

$I{+}{+};$

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

;

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

Algorithme 6: Algorithme du maximum d'une liste annoté

EXTENDS *Naturals*, *TLC*
CONSTANTS n
VARIABLES m, i, l

$f \triangleq [j \in 0..n-1 \mapsto j]$

$$\textit{Init} \triangleq \land\, i = 0$$
$$\land\, m = 0$$
$$\land\, l = \text{"l0"}$$

$$l0l_1 \triangleq \land\, l = \text{"l0"}$$
$$\land\, m' = f[0]$$
$$\land\, i' = i$$
$$\land\, l' = \text{"l1"}$$

$$l1l_2 \triangleq \land\, l = \text{"l1"}$$
$$\land\, m' = m$$
$$\land\, i' = 1$$
$$\land\, l' = \text{"l2"}$$

$$l2l_3 \triangleq \land\, l = \text{"l2"}$$
$$\land\, i < n$$
$$\land\, m' = m$$
$$\land\, i' = i$$
$$\land\, l' = \text{"l3"}$$

$$l2l_8 \triangleq \land\, l = \text{"l2"}$$
$$\land\, (i \geq n)$$
$$\land\, m' = m$$
$$\land\, i' = i$$
$$\land\, l' = \text{"l8"}$$

$$l3l_4 \triangleq \land\, l = \text{"l3"}$$
$$\land\, f[i] > m$$
$$\land\, m' = m$$
$$\land\, i' = i$$
$$\land\, l' = \text{"l4"}$$

$$l3l_6 \triangleq \land\, l = \text{"l3"}$$
$$\land\, (f[i] \leq m)$$
$$\land\, m' = m$$
$$\land\, i' = i$$
$$\land\, l' = \text{"l6"}$$

$$l4l_5 \triangleq \land\, l = \text{"l4"}$$
$$\land\, m' = f[i]$$
$$\land\, i' = i$$
$$\land\, l' = \text{"l5"}$$

$$l5l_6 \triangleq \land\, l = \text{"l5"}$$
$$\land\, m' = m$$
$$\land\, i' = i$$
$$\land\, l' = \text{"l6"}$$

$$l6l_7 \triangleq \land\, l = \text{"l6"}$$

$\wedge\ m' \ = \ m$
$\wedge\ i' \ = \ i+1$
$\wedge\ l' = $ "l7"

$l7l_2 \triangleq \wedge\ l \ = $ "l7"
$\wedge\ m' \ = \ m$
$\wedge\ i' \ = \ i$
$\wedge\ l' = $ "l2"

$Next \triangleq \vee\ l0l1$
$\vee\ l1l2$
$\vee\ l2l3$
$\vee\ l2l8$
$\vee\ l3l4$
$\vee\ l3l6$
$\vee\ l4l5$
$\vee\ l5l6$
$\vee\ l6l7$
$\vee\ l7l2$

$Safel_3 \triangleq l = \text{"l3"} \Rightarrow \wedge (i \in 1..n-1)$
$\wedge (\exists\, k \ : \ (k \in 0.i-1) \wedge f[k] = m)$
$\wedge (\forall\, j \ : \ j \in 0..i-1 \Rightarrow f[j] \leq m)$
$safety \triangleq l = \text{"l8"} \Rightarrow (\forall\, k \in 0..n-1 : m \geq f[k])$ partial correctnes
$Safety_2 \triangleq l \neq \text{"l8"}$

Fin 10

Exercice 11 *On considère l'algorithme* squareroot *11 calculant la racine carrée entière d'un nombre naturel $x \in \mathbb{N}$.*

Question 11.1 *Complétez cet algorithme en proposant trois assertions :*
— $P_{\ell_2}(z, y1, y2, y3)$
— $P_{\ell_4}(z, y1, y2, y3)$
— $P_{\ell_5}(z, y1, y2, y3)$

Question 11.2 *Pour chaque paire (ℓ, ℓ') d'étiquettes correspondant à un pas élémentaire ; on vérifie la propriété suivante :*
$\forall x, y, q, r, x', y', q', r'.P_\ell(y_1, y_2, y_3, z) \wedge cond_{\ell,\ell'}(y_1, y_2, y_3, z) \wedge (y'_1, y'_2, y'_3, z') = f_{\ell,\ell'}(y_1, y_2, y_3, z) \Rightarrow P_{\ell'}(y'_1, y'_2, y'_3, z')$
Enoncez et vérifiez cette propriété pour les paires d'étiquettes suivantes : (ℓ_1, ℓ_2) ;(ℓ_1, ℓ_4) ; (ℓ_2, ℓ_3) ; (ℓ_3, ℓ_2) ; (ℓ_3, ℓ_4) ; (ℓ_4, ℓ_5) ;

Question 11.3 *On suppose que toutes les conditions de vérifications associées aux paires d'étiquettes successives de l'algorithme sont vérifiées. Quelles sont les deux conditions à montrer pour déduire que l'algorithme est partiellement correct par rapport aux pré et post conditions ? Vous donnerez explicitement les conditions et vous expliquerez pourquoi elles sont correctes.*

Question 11.4 *Expliquer que cet algorithme est sans erreurs à l'exécution, si les données initiales sont dans un domaine à définir inclus dans le domaine des entiers informatiques c'est-à-dire les entiers codables sur n bits. L'ensemble des entiers informatiques sur n bits est l'ensemble noté $\mathbb{Z}_n$ et défini par $\{i | i \in \mathbb{Z} \wedge -2^{n-1} \leq i \wedge i \leq 2^{n-1}-1\}$.*

precondition : $x \in \mathbb{N}$
postcondition : $z^2 \leq x \wedge x < (z{+}1)^2$
local variables : $y_1, y_2, y_3 \in \mathbb{N}$

$\ell_0 : \{x \in \mathbb{N} \wedge z \in \mathbb{Z} \wedge y1 \in \mathbb{Z} \wedge y2 \in \mathbb{Z} \wedge y3 \in \mathbb{Z}\}$
$(y_1, y_2, y_3) := (0, 1, 1);$
$\ell_1 : \{y2 = (y1{+}1)\cdot(y1{+}1) \wedge y3 = 2\cdot y1{+}1 \wedge y1\cdot y1 \leq x\}$
while $y_2 \leq x$ **do**
    $\ell_2 : \{\ldots\}$
    $(y_1, y_2, y_3) := (y_1{+}1, y_2{+}y_3{+}2, y_3{+}2);$
    $\ell_3 : \{y2 = (y1{+}1)\cdot(y1{+}1) \wedge y3 = 2\cdot y1{+}1 \wedge y1\cdot y1 \leq x\}$
;
$\ell_4 : \{\ldots\}$
$z := y_1;$
$\ell_5 : \{\ldots\}$

Algorithme 7: *squareroot* partiellement annotée

precondition : $x \in \mathbb{N}$
postcondition : $z^2 \leq x \wedge x < (z{+}1)^2$
local variables : $y_1, y_2, y_3 \in \mathbb{N}$

$pre : \{x \in \mathbb{N}\}$
$post : \{z\cdot z \leq x \wedge x < (z{+}1)\cdot(z{+}1)\}$
$\ell_0 : \{x \in \mathbb{N} \wedge z \in \mathbb{Z} \wedge y1 \in \mathbb{Z} \wedge y2 \in \mathbb{Z} \wedge y3 \in \mathbb{Z}\}$
$(y_1, y_2, y_3) := (0, 1, 1);$
$\ell_1 : \{y2 = (y1{+}1)\cdot(y1{+}1) \wedge y3 = 2\cdot y1{+}1 \wedge y1\cdot y1 \leq x\}$
while $y_2 \leq x$ **do**
    $\ell_2 : \{y2 = (y1{+}1)\cdot(y1{+}1) \wedge y3 = 2\cdot y1{+}1 \wedge y2 \leq x\}$
    $(y_1, y_2, y_3) := (y_1{+}1, y_2{+}y_3{+}2, y_3{+}2);$
    $\ell_3 : \{y2 = (y1{+}1)\cdot(y1{+}1) \wedge y3 = 2\cdot y1{+}1 \wedge y1\cdot y1 \leq x\}$
;
$\ell_4 : \{y2 = (y1{+}1)\cdot(y1{+}1) \wedge y3 = 2\cdot y1{+}1 \wedge y1\cdot y1 \leq x \wedge x < y2\}$
$z := y_1;$
$\ell_5 : \{y2 = (y1{+}1)\cdot(y1{+}1) \wedge y3 = 2\cdot y1{+}1 \wedge y1\cdot y1 \leq x \wedge x < y2 \wedge z = y1 \wedge z\cdot z \leq x \wedge x < (z{+}1)\cdot(z{+}1)\}$

Algorithme 8: *squareroot* annotée

L'algorithme annoté est décrit par l'algorithme 11

MODULE *algo_squareroot*

EXTENDS *Integers*, *TLC*
CONSTANTS x *x is the input*
VARIABLES $pc, y_1, y2, y_3, z$

$vars \triangleq \langle pc, y_1, y2, y_3, z \rangle$
$al0l_1 \triangleq pc = \text{"l0"} \land y_1' = 0 \land y_2' = 1 \land y_3' = 1 \land pc' = \text{"l1"} \land z' = z$

$al1l_2 \triangleq pc = \text{"l1"} \land y_2 \leq x \land pc' = \text{"l2"} \land \text{UNCHANGED } \langle y_1, y2, y_3, z \rangle$

$al1l_4 \triangleq pc = \text{"l1"} \land y_2 > x \land pc' = \text{"l4"} \land \text{UNCHANGED } \langle y_1, y2, y_3, z \rangle$

$al2l_3 \triangleq pc = \text{"l2"} \land y_1' = y_1{+}1 \land y_2' = y_2{+}y3{+}2 \land y_3' = y_3{+}2 \land pc' = \text{"l3"} \land z' = z$

$al3l_2 \triangleq pc = \text{"l3"} \land y_2 \leq x \land pc' = \text{"l2"} \land \text{UNCHANGED } \langle y_1, y2, y_3, z \rangle$

$al3l_4 \triangleq pc = \text{"l3"} \land y_2 > x \land pc' = \text{"l4"} \land \text{UNCHANGED } \langle y_1, y2, y_3, z \rangle$

$al4l_5 \triangleq pc = \text{"l4"} \land z' = y_1 \land pc' = \text{"l5"} \land \text{UNCHANGED } \langle y_1, y2, y_3 \rangle$

$Init \triangleq y_1 = 0 \land y_2 = 0 \land y_3 = 0 \land z = 0 \land pc = \text{"l0"}$

$Next \triangleq al0l_1 \lor al1l_2 \lor al1l_4 \lor al2l_3 \lor al3l_2 \lor al3l_4 \lor al4l_5$

$MAX \triangleq 32768$ *16 bits*
$D \triangleq 0..32768$
x \leq 32760

$Safety\_absence \triangleq (y_1 \in D) \land (y_2 \in D) \land (y_3 \in D) \land (z \in D)$

$i \triangleq$
$\land\, pc = \text{"l0"} \Rightarrow y_1 \in D \land y_2 \in D \land y_3 \in D \land z \in D$
$\land\, pc = \text{"l1"} \Rightarrow y_2 = (y_1{+}1) \cdot (y_1{+}1) \land y_3 = 2 \cdot y_1{+}1 \land y_1 \cdot y1 \leq x \land Safety\_absence$
$\land\, pc = \text{"l2"} \Rightarrow y_2 = (y_1{+}1) \cdot (y_1{+}1) \land y_3 = 2 \cdot y_1{+}1 \land y_1 \cdot y1 \leq x \land y_2 \leq x \land Safety\_absence$

$\land\, pc = \text{"l3"} \Rightarrow y_2 = (y_1{+}1) \cdot (y_1{+}1) \land y_3 = 2 \cdot y_1{+}1 \land y_1 \cdot y1 \leq x \land Safety\_absence$
$\land\, pc = \text{"l4"} \Rightarrow y_2 = (y_1{+}1) \cdot (y_1{+}1) \land y_3 = 2 \cdot y_1{+}1 \land y_1 \cdot y1 \leq x \land x < y_2 \land Safety\_absence$

$\land\, pc = \text{"l5"} \Rightarrow y_2 = (y_1{+}1) \cdot (y_1{+}1) \land y_3 = 2 \cdot y_1{+}1 \land z \cdot z \leq x \land x < (z{+}1) \cdot (z{+}1) \land Safety\_absence$

$Safety\_partialcorrectness \triangleq pc = \text{"l5"} \Rightarrow \land\, y_2 = (y_1{+}1) \cdot (y_1{+}1)$
$\qquad \land\, y_3 = 2 \cdot y_1{+}1$
$\qquad \land\, z \cdot z \leq x \land x < (z{+}1) \cdot (z{+}1)$

Exercice 12

Montrer, pour chaque question, que chaque annotation est correcte ou incorrecte selon les conditions de vérifications énoncées comme suit
$\forall v, v'. P_\ell(v) \land cond_{\ell,\ell'}(v) \land v' = f_{\ell,\ell'}(v) \Rightarrow P_{\ell'}(v')$. *Vous devez répondre en énonçant et en démontrant les Conditions de vérification.*

Question 12.1

$\ell_1 : x = 12 \ \wedge \ y = 2 \ \wedge z = 3 \cdot x$
$x := z + y$
$\ell_2 : x = 38 \ \wedge \ y = 2$

Question 12.2

$\ell_1 : x = 3 \ \wedge \ y = 9$
$x := 3 \cdot y$
$\ell_2 : x = 27 \ \wedge \ y = 9$

Question 12.3 *Soit p un nombre différent d'une puissance de 3 c'est-à-dire différent de 3, 6, 9, 12, ...*

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

Question 12.4 *Soit r un nombre cubique c'est-à-dire de la forme $p = q^3$.*

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

Exercice 13

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.

Question 13.1 *Compléter les annotations associées à chaque étiquette $\ell \in \{\ell_3, \ell_6, \ell_8, \ell_9\}$. Vous devez écrire les annotations complètes de chaque point de contrôle demandé.*

Question 13.2 *Pour chaque paire (ℓ, ℓ') d'étiquettes correspondant à un pas élémentaire ; on vérifie la propriété suivante :*
$\forall x, y, q, r, x', y', q', r'. P_\ell(y_1, y_2, y_3, z) \ \wedge \ cond_{\ell,\ell'}(y_1, y_2, y_3, z) \ \wedge \ (y'_1, y'_2, y'_3, z') = f_{\ell,\ell'}(y_1, y_2, y_3, z) \Rightarrow P_{\ell'}(y'_1, y'_2, y'_3, z')$
Enoncer et vérifier cette propriété pour les paires d'étiquettes suivantes : (ℓ_0, ℓ_1) ; (ℓ_1, ℓ_2) ; (ℓ_3, ℓ_4) ; (ℓ_6, ℓ_7) ; (ℓ_7, ℓ_8) ; (ℓ_1, ℓ_9) ; (ℓ_9, ℓ_{10}).
Il est clair que cette vérification confirmera les complétions réalisées dans la question précédente.

Question 13.3 *On suppose que toutes les conditions de vérifications associées aux paires d'étiquettes successives de l'algorithme sont vérifiées. Quelles sont les deux conditions à montrer pour déduire que l'algorithme est partiellement correct par rapport aux pré et post conditions ? Vous donnerez explicitement les conditions et vous expliquerez pourquoi elles sont correctes.*

Question 13.4 *Selon la définition mathématique de la puissance $x_1^{x_2}$ est définie pour une valeur x_1 non nulle et c'est pour cela que la précondition indique que x_1 est différent de 0. Cependant, si on utilise une valeur de x_1 nulle, l'algorithme fonctionne et renvoie une valeur. Un jour, un mathématicien a appliqué cet algorithme sans veiller à ce que la valeur de x_1 soit nulle ou non nulle et il 'est emporté !... Il vous accuse de ne pas lui avoir fourni le bon algorithme répondant à son cahier des charges et il vous demande des dommages et intérêts. Expliquer de manière courte que le texte de l'algorithme et sa preuve de correction suffisent pour vous sauver, en expliquant clairement le rôle de la précondition et de la postcndition.*

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

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

Algorithme 9: Algorithme de l'exponentitaion indienne annoté