

Cours Modélisation et vérification des systèmes informatiques

Exercices (avec les corrections)

Modélisation d'algorithmes en TLA⁺

Annotation, modélisation et vérification

par Dominique Méry

11 novembre 2019



annotation

Exercice 1 ✓

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

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

Cette condition s'écrit initialement :

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

mais on peut réduire en oubliant la variable pc.

<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> $\begin{aligned} \ell_1 : x = 10 \wedge y = z+x \wedge z = 2 \cdot x \\ y := z+x \\ \ell_2 : x = 10 \wedge y = x+2 \cdot 10 \end{aligned}$ </div> <p>— On suppose que p est un nombre premier :</p> <div style="border: 1px solid black; padding: 5px;"> $\begin{aligned} \ell_1 : x = 2^p \wedge y = 2^{p+1} \wedge x \cdot y = 2^{2 \cdot p+1} \\ x := y+x+2^x \\ \ell_2 : x = 5 \cdot 2^p \wedge y = 2^{p+1} \end{aligned}$ </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> $\begin{aligned} \ell_1 : x = 1 \wedge y = 12 \\ x := 2 \cdot y \\ \ell_2 : x = 1 \wedge y = 24 \end{aligned}$ </div> <p>—</p> <div style="border: 1px solid black; padding: 5px;"> $\begin{aligned} \ell_1 : x = 11 \wedge y = 13 \\ z := x; x := y; y := z; \\ \ell_2 : x = 26/2 \wedge y = 33/3 \end{aligned}$ </div>
--	--

Exercice 2 ✓

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

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

<p>— (1)</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> $\begin{aligned} \ell_1 : x = 9 \wedge y = z+x \\ y := x+9 \\ \ell_2 : x = 9 \wedge y = x+9 \end{aligned}$ </div> <p>— (2)</p> <div style="border: 1px solid black; padding: 5px;"> $\begin{aligned} \ell_1 : x = 1 \wedge y = 3 \wedge x+y = 12 \\ x := y+x \\ \ell_2 : x = 567 \wedge y = 34 \end{aligned}$ </div>	<p>—</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> $\begin{aligned} \ell_1 : x = 3 \wedge y = 3 \\ x := y+x \\ \ell_2 : x = 6 \wedge y = 3 \end{aligned}$ </div> <p>—</p> <div style="border: 1px solid black; padding: 5px;"> $\begin{aligned} \ell_1 : x = 1 \wedge y = 3 \\ z := x; x := y; y := z; \\ \ell_2 : x = 3 \wedge y = 1 \end{aligned}$ </div>
---	---

1. $c = \ell_1 \wedge x = 9 \wedge y = z+x \wedge \text{TRUE} \wedge (x', y', c') = (x, x+9, \ell_2) \Rightarrow c' = \ell_2 \wedge x' = 9 \wedge y' = x'+9 :$

(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. $c = \ell_1 \wedge x = 1 \wedge y = 3 \wedge x+y = 12 \wedge \text{TRUE} \wedge (x', y', c') = (y+x, y, \ell_2) \Rightarrow c' = \ell_2 \wedge x' = 567 \wedge y' = 34 :$

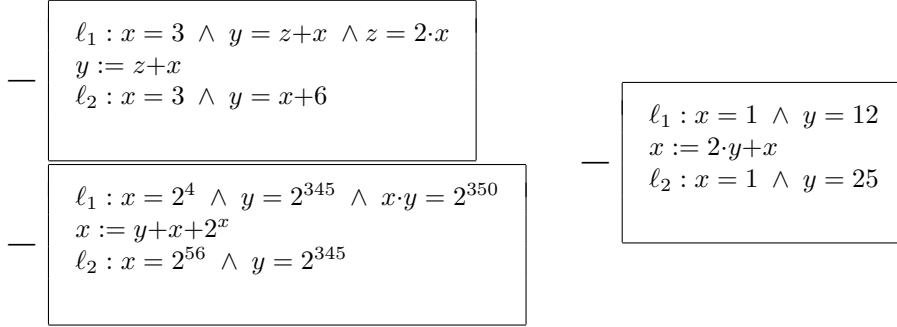
(a) $c = \ell_1 \wedge x = 1 \wedge y = 3 \wedge x+y = 12 \wedge (x', y', c') = (y+x, y, \ell_2) \Rightarrow c' = \ell_2 \wedge y+x = 567 \wedge y = 34$

- (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) $c = \ell_1 \wedge x = 1 \wedge y = 3 \wedge x+y = 12 \wedge c' = \ell_2 \Rightarrow \mathbf{FALSE}$
- (e) $\mathbf{FALSE} \Rightarrow c' = \ell_2 \wedge y+x = 567 \wedge y = 34$
- (f) **CORRECT**
3. $c = \ell_1 \wedge x = 3 \wedge y = 3 \wedge \mathbf{TRUE} \wedge (x', y', c') = (y+x, y, \ell_2) \Rightarrow c' = \ell_2 \wedge x' = 6 \wedge y' = 3$
- (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. $c = \ell_1 \wedge x = 1 \wedge y = 3 \wedge \mathbf{TRUE} \wedge (x', y', z', c') = (y, x, x, \ell_2) \Rightarrow c' = \ell_2 \wedge x' = 3 \wedge y' = 1$
- (a) $c = \ell_1 \wedge x = 1 \wedge y = 3 \wedge \mathbf{TRUE} \wedge (x', y', z', c') = (y, x, x, \ell_2) \Rightarrow c' = \ell_2 \wedge y = 3 \wedge x = 1$
- (b) **CORRECT**

Exercice 3 \square

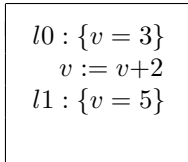
Montrer que chaque annotation est correcte ou incorrecte selon les conditions de vérifications énoncées comme suit

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



Exercice 4 \square

Soit le petit algorithme annoté suivant :



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 an1
EXTENDS Integers, TLC
VARIABLES v, pc
titi  $\triangleq$  pc = "l0"  $\wedge$  v = 3

```

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

$trans \triangleq pc = "l0" \wedge TRUE \wedge pc' = "l1" \wedge v' = v+2$
 $trans_2 \triangleq pc = "l0" \wedge pc' = "l1" \wedge v' = v+2$
 $trans_3 \triangleq$
 $\quad \wedge pc = "l0" \wedge TRUE$
 $\quad \wedge pc' = "l1"$
 $\quad \wedge v' = v+2$

$toto \triangleq skip \vee trans$

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

$safety \triangleq pc = "l1" \Rightarrow v = 5$

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

$$\begin{aligned}
&\ell1 : \{P_{\ell1}(x, y)\} \\
&x := x+y+7; \\
&\ell2 : \{P_{\ell2}(x, y)\}
\end{aligned}$$

◇ Solution de la question 5.1

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

$$pc = \ell1 \wedge P_{\ell1}(x, y) \wedge pc = \ell1 \wedge pc' = \ell2 \wedge x' = x+y+7 \wedge y' = y \Rightarrow pc' = \ell2 \wedge P_{\ell2}(x', y')$$

qui se simplifie en :

$$P_{\ell1}(x, y) \wedge x' = x+y+7 \wedge y' = y \Rightarrow P_{\ell2}(x', y')$$

qui se simplifie en :

$$P_{\ell1}(x, y) \wedge x' = x+y+7 \Rightarrow P_{\ell2}(x', y)$$

qui se simplifie en :

$$P_{\ell1}(x, y) \Rightarrow P_{\ell2}(x+y+7, y)$$

Modèle TLA⁺ pour vérifier la bonne annotation EXTENDS *Naturals*
 VARIABLES x, y, pc

Define actions from the text of annotated algorithm

$al0l1 \triangleq$
 $\wedge pc = "l0"$
 $\wedge pc' = "l1"$
 $\wedge x' = x + y + 7$
 $\wedge y' = y$

Define the computation relation

$next \triangleq al0l1$

Define the initial conditions

$init \triangleq pc = "l0" \wedge x = 3 \wedge y = 8$

Define the invariant from the annotation

$i \triangleq$
 $\wedge pc = "l0" \Rightarrow x = 3 \wedge y = 8$
 $\wedge pc = "l1" \Rightarrow x = 6 \wedge y = 89$

Define the safety property to check namely the partial correctness

$safe \triangleq pc = "l1" \Rightarrow x = 7 \wedge y = 89$

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

EXTENDS *Naturals*
 VARIABLES x, y, pc

Define actions from the text of annotated algorithm

$al0l1 \triangleq$
 $\wedge pc = "l0"$
 $\wedge pc' = "l1"$
 $\wedge x' = x + y + 7$
 $\wedge y' = y$

Define the computation relation

$next \triangleq al0l1$

Define the initial conditions

$init \triangleq pc = "l0" \wedge x = 3 \wedge y = 8$

Define the invariant from the annotation

$i \triangleq$

$$\wedge pc = "l0" \Rightarrow x = 3 \wedge y = 8$$

$$\wedge pc = "l1" \Rightarrow x = 18 \wedge y = 8$$

Define the safety property to check namely the partial correctness

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

$$prop \triangleq i \Rightarrow safe$$

$$Init \triangleq init$$

$$Next \triangleq next$$

$$principe \triangleq init \Rightarrow \wedge prop$$

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

$$\begin{aligned} \ell &: \{P_\ell(x, y)\} \\ x, y &:= y, x; \\ \ell' &: \{P_{\ell'}(x, y)\} \end{aligned}$$

◇ Solution de la question 5.2

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

$$c = \ell1 \wedge P_{\ell1}(x, y) \wedge c' = \ell2 \wedge (x', y') = (y, x) \Rightarrow c' = \ell2 \wedge P_{\ell2}(x', y')$$

qui se simplifie en :

$$P_{\ell1}(x, y) \wedge x' = y \wedge y' = x \Rightarrow P_{\ell2}(x', y')$$

qui se simplifie en :

$$P_{\ell1}(x, y) \Rightarrow P_{\ell2}(y, x)$$

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

$$al1l2 \triangleq$$

$$\wedge pc = "l1"$$

$$\wedge pc' = "l2"$$

$$\wedge x' = y \wedge y' = x$$

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

Define the computation relation

$$next \triangleq al1l2$$

$newnext \triangleq all_2 \vee newaction$

Define the initial conditions

$init \triangleq pc = "I1" \wedge x = a \wedge y = b$

Define the invariant from the annotation

$i \triangleq$

$\wedge pc = "I1" \Rightarrow x = a \wedge y = b$

$\wedge pc = "I2" \Rightarrow x = b \wedge y = a$

Define the safety property to check namely the partial correctness

$safe \triangleq pc = "I2" \Rightarrow x = b \wedge y = a$

Fin 5.2

Exercice 6 \square

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 \square

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 $ex4$

EXTENDS *Naturals*

CONSTANTS a, b

precondition : $x = a \wedge y = b \wedge a, b \in \mathbb{N}$
postcondition : $z = \max(a, b)$

```

 $\ell_0 : \{\dots\}$ 
if  $x < y$  then
  |  $\ell_1 : \{\dots\}$ 
  |  $z := y;$ 
  |  $\ell_2 : \{\dots\}$ 
else
  |  $\ell_3 : \{\dots\}$ 
  |  $z := x;$ 
  |  $\ell_4 : \{\dots\}$ 
  |
;
 $\ell_5 : \{\dots\}$ 

```

Algorithme 1: maximum de deux nombres non annotée

precondition : $x = a \wedge y = b \wedge a, b \in \mathbb{N}$
postcondition : $z = \max(a, b)$

```

 $\ell_0 : \{x = a \wedge y = b\}$ 
if  $x < y$  then
  |  $\ell_1 : \{x < y \wedge x = a \wedge y = b \wedge a < b\}$ 
  |  $z := y;$ 
  |  $\ell_2 : \{x < y \wedge x = a \wedge y = b \wedge a < b \wedge z = b\}$ 
else
  |  $\ell_3 : \{x \geq y \wedge x = a \wedge y = b \wedge a \geq b\}$ 
  |  $z := x;$ 
  |  $\ell_4 : \{x \geq y \wedge x = a \wedge y = b \wedge a \geq b \wedge z = a\}$ 
  |
;
 $\ell_5 : \{z = \max(a, b) \wedge x = a \wedge y = b\}$ 

```

Algorithme 2: maximum de deux nombres annotée

VARIABLES x, y, z, pc

$$\begin{aligned} al0l_1 &\triangleq \\ &\wedge pc = "l0" \\ &\wedge pc' = "l1" \\ &\wedge x < y \\ &\wedge z' = z \wedge x' = x \wedge y' = y \\ al1l_2 &\triangleq \\ &\wedge pc = "l1" \\ &\wedge pc' = "l2" \\ &\wedge z' = y \\ &\wedge x' = x \wedge y' = y \\ al2l_5 &\triangleq \\ &\wedge pc = "l2" \\ &\wedge pc' = "l5" \\ &\wedge z' = z \wedge x' = x \wedge y' = y \\ al0l_3 &\triangleq \\ &\wedge pc = "l0" \\ &\wedge pc' = "l3" \\ &\wedge x \geq y \\ &\wedge z' = z \wedge x' = x \wedge y' = y \\ al3l_4 &\triangleq \\ &\wedge pc = "l3" \\ &\wedge pc' = "l4" \\ &\wedge z' = x \\ &\wedge x' = x \wedge y' = y \\ al4l_5 &\triangleq \\ &\wedge pc = "l4" \\ &\wedge pc' = "l5" \\ &\wedge z' = z \wedge x' = x \wedge y' = y \\ \text{next} &\triangleq al0l_1 \vee al1l_2 \vee al2l_5 \vee al0l_3 \vee al3l_4 \vee al4l_5 \\ \text{init} &\triangleq pc = "l0" \wedge x = a \wedge y = b \wedge z = a \end{aligned}$$

$$\begin{aligned} i &\triangleq \\ &\wedge pc = "l0" \Rightarrow x = a \wedge y = b \\ &\wedge pc = "l1" \Rightarrow x < y \wedge x = a \wedge y = b \\ &\wedge pc = "l2" \Rightarrow x < y \wedge x = a \wedge y = b \wedge z = b \\ &\wedge pc = "l3" \Rightarrow x \geq y \wedge x = a \wedge y = b \\ &\wedge pc = "l4" \Rightarrow x \geq y \wedge x = a \wedge y = b \wedge z = a \\ &\wedge pc = "l5" \Rightarrow ((a < b \wedge z = b) \vee (a \geq b \wedge z = a)) \\ \text{safe} &\triangleq pc = "l5" \Rightarrow ((a < b \wedge z = b) \vee (a \geq b \wedge z = a)) \\ \text{safeab} &\triangleq x = a \wedge y = b \end{aligned}$$

Modification History

Last modified Tue Dec 15 17:37:42 CET 2015 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 conditions 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

precondition : $x = x_0 \wedge x_0 \in \mathbb{N}$

postcondition : $x = 0$

$\ell_0 : \{\dots\}$

while $0 < x$ **do**

$\ell_1 : \{\dots\}$

$x := x-1;$

$\ell_2 : \{\dots\}$

;

$\ell_3 : \{\dots\}$

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.

precondition : $x = x_0 \wedge x_0 \in \mathbb{N}$

postcondition : $x = 0$

$\ell_0 : \{x = x_0 \wedge x_0 \in \mathbb{N}\}$

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

MODULE *ex3*

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

CONSTANTS x_0

VARIABLES x, pc

$$\begin{aligned} al0l_1 &\triangleq \\ &\wedge pc = "l0" \\ &\wedge pc' = "l1" \\ &\wedge 0 < x \\ &\wedge x' = x \end{aligned}$$

$$\begin{aligned} al0l_3 &\triangleq \\ &\wedge pc = "l0" \\ &\wedge pc' = "l3" \\ &\wedge x = 0 \\ &\wedge x' = x \end{aligned}$$

$$\begin{aligned} al1l_2 &\triangleq \\ &\wedge pc = "l1" \\ &\wedge pc' = "l2" \\ &\wedge x' = x - 1 \end{aligned}$$

$$\begin{aligned} al2l_1 &\triangleq \\ &\wedge pc = "l2" \\ &\wedge pc' = "l1" \\ &\wedge 0 < x \\ &\wedge x' = x \end{aligned}$$

$$\begin{aligned} al2l_3 &\triangleq \\ &\wedge pc = "l2" \\ &\wedge pc' = "l3" \\ &\wedge 0 = x \\ &\wedge x' = x \end{aligned}$$

$$next \triangleq al0l_1 \vee al0l_3 \vee al1l_2 \vee al2l_1 \vee al2l_3$$

$$init \triangleq pc = "l0" \wedge x = x_0$$

$$\begin{aligned} i &\triangleq \\ &\wedge pc = "l0" \Rightarrow x = x_0 \\ &\wedge pc = "l1" \Rightarrow 0 < x \wedge x \leq x_0 \\ &\wedge pc = "l2" \Rightarrow 0 \leq x \wedge x \leq x_0 \\ &\wedge pc = "l3" \Rightarrow x = 0 \end{aligned}$$

$$safe \triangleq pc = "l3" \Rightarrow x = 0$$

$$safeplus \triangleq x \geq 0$$

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 $\max(m, t, a, b) = \dots$ 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** —

$$\text{max}(m, t, a, b) \stackrel{\text{def}}{=} m \in \text{ran}(t) \wedge (\forall i. i \in a .. b \Rightarrow t(i) \leq m)$$

Fin 9

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

◊— **Solution de l'exercice 9** —

$$\text{trié}(t, a, b) \stackrel{\text{def}}{=} \forall i, j. ((i \in a .. b \wedge j \in a .. b \wedge i \leq j) \Rightarrow t(i) \leq t(j))$$

Fin 9

Exercice 10 Dans l'algorithme 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

```

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 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 |

|-----|

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

|-----|

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

|-----|

 $Init \triangleq \wedge i = 0$ 
 $\wedge m = 0$ 

```

/* algorithme de calcul du maximum avec une boucle while de l'exercice 10 */

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 : $i \in \mathbb{Z}$

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

$m := f(0);$

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

$i := 1;$

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

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**

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

$f(i) > m \}$

$m := f(i);$

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

;

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

$i++;$

$\ell_7 : \left\{ \left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0..n-1 \rightarrow \mathbb{N} \end{array} \right) \wedge i \in 2..n \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in \text{ran}(f[0..i-1]) \wedge \\ (\forall j. j \in 0..i-1 \Rightarrow f(j) \leq m) \end{array} \right) \right\}$

;

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

Algorithme 6: Algorithme du maximum d'une liste annoté

$$\wedge l = \text{"10"}$$

$$\begin{aligned} l0l_1 &\triangleq \wedge l = \text{"10"} \\ &\wedge m' = f[0] \\ &\wedge i' = i \\ &\wedge l' = \text{"11"} \end{aligned}$$

$$\begin{aligned} l1l_2 &\triangleq \wedge l = \text{"11"} \\ &\wedge m' = m \\ &\wedge i' = 1 \\ &\wedge l' = \text{"12"} \end{aligned}$$

$$\begin{aligned} l2l_3 &\triangleq \wedge l = \text{"12"} \\ &\wedge i < n \\ &\wedge m' = m \\ &\wedge i' = i \\ &\wedge l' = \text{"13"} \end{aligned}$$

$$\begin{aligned} l2l_8 &\triangleq \wedge l = \text{"12"} \\ &\wedge (i \geq n) \\ &\wedge m' = m \\ &\wedge i' = i \\ &\wedge l' = \text{"18"} \end{aligned}$$

$$\begin{aligned} l3l_4 &\triangleq \wedge l = \text{"13"} \\ &\wedge f[i] > m \\ &\wedge m' = m \\ &\wedge i' = i \\ &\wedge l' = \text{"14"} \end{aligned}$$

$$\begin{aligned} l3l_6 &\triangleq \wedge l = \text{"13"} \\ &\wedge (f[i] \leq m) \\ &\wedge m' = m \\ &\wedge i' = i \\ &\wedge l' = \text{"16"} \end{aligned}$$

$$\begin{aligned} l4l_5 &\triangleq \wedge l = \text{"14"} \\ &\wedge m' = f[i] \\ &\wedge i' = i \\ &\wedge l' = \text{"15"} \end{aligned}$$

$$\begin{aligned} l5l_6 &\triangleq \wedge l = \text{"15"} \\ &\wedge m' = m \\ &\wedge i' = i \\ &\wedge l' = \text{"16"} \end{aligned}$$

$$\begin{aligned} l6l_7 &\triangleq \wedge l = \text{"16"} \\ &\wedge m' = m \\ &\wedge i' = i + 1 \\ &\wedge l' = \text{"17"} \end{aligned}$$

$$\begin{aligned} l7l_2 &\triangleq \wedge l = \text{"17"} \\ &\wedge m' = m \\ &\wedge i' = i \\ &\wedge l' = \text{"12"} \end{aligned}$$

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

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

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 :

$\forall x, y, q, r, x', y', q', r'. P_{\ell}(y1, y2, y3, z) \wedge cond_{\ell, \ell'}(y1, y2, y3, z) \wedge (y'_1, y'_2, y'_3, z') = f_{\ell, \ell'}(y1, y2, y3, z) \Rightarrow P_{\ell'}(y'_1, y'_2, y'_3, z')$

Énoncez et vérifiez cette propriété pour les paires d'étiquettes suivantes : $(\ell_1, \ell_2); (\ell_1, \ell_4); (\ell_2, \ell_3); (\ell_3, \ell_2); (\ell_3, \ell_4); (\ell_4, \ell_5);$

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 $\{i | i \in \mathbb{Z} \wedge -2^{n-1} \leq i \wedge i \leq 2^{n-1}-1\}$.

L'algorithme annoté est décrit par l'algorithme 11

MODULE *algo_squareroot*

EXTENDS *Integers, TLC*

CONSTANTS x x is the input

VARIABLES $pc, y1, y2, y3, z$

$vars \triangleq \langle pc, y1, y2, y3, z \rangle$

precondition : $x \in \mathbb{N}$
postcondition : $z^2 \leq x \wedge x < (z+1)^2$
local variables : $y_1, y_2, y_3 \in \mathbb{N}$
 $\ell_0 : \{x \in \mathbb{N} \wedge z \in \mathbb{Z} \wedge y_1 \in \mathbb{Z} \wedge y_2 \in \mathbb{Z} \wedge y_3 \in \mathbb{Z}\}$
 $(y_1, y_2, y_3) := (0, 1, 1);$
 $\ell_1 : \{y_2 = (y_1+1) \cdot (y_1+1) \wedge y_3 = 2 \cdot y_1+1 \wedge y_1 \cdot y_1 \leq x\}$
while $y_2 \leq x$ **do**
 $\ell_2 : \{\dots\}$
 $(y_1, y_2, y_3) := (y_1+1, y_2+y_3+2, y_3+2);$
 $\ell_3 : \{y_2 = (y_1+1) \cdot (y_1+1) \wedge y_3 = 2 \cdot y_1+1 \wedge y_1 \cdot y_1 \leq x\}$

;
 $\ell_4 : \{\dots\}$
 $z := y_1;$
 $\ell_5 : \{\dots\}$

Algorithme 7: *squareroot* partiellement annotée

precondition : $x \in \mathbb{N}$
postcondition : $z^2 \leq x \wedge x < (z+1)^2$
local variables : $y_1, y_2, y_3 \in \mathbb{N}$
 $pre : \{x \in \mathbb{N}\}$
 $post : \{z \cdot z \leq x \wedge x < (z+1) \cdot (z+1)\}$
 $\ell_0 : \{x \in \mathbb{N} \wedge z \in \mathbb{Z} \wedge y_1 \in \mathbb{Z} \wedge y_2 \in \mathbb{Z} \wedge y_3 \in \mathbb{Z}\}$
 $(y_1, y_2, y_3) := (0, 1, 1);$
 $\ell_1 : \{y_2 = (y_1+1) \cdot (y_1+1) \wedge y_3 = 2 \cdot y_1+1 \wedge y_1 \cdot y_1 \leq x\}$
while $y_2 \leq x$ **do**
 $\ell_2 : \{y_2 = (y_1+1) \cdot (y_1+1) \wedge y_3 = 2 \cdot y_1+1 \wedge y_2 \leq x\}$
 $(y_1, y_2, y_3) := (y_1+1, y_2+y_3+2, y_3+2);$
 $\ell_3 : \{y_2 = (y_1+1) \cdot (y_1+1) \wedge y_3 = 2 \cdot y_1+1 \wedge y_1 \cdot y_1 \leq x\}$

;
 $\ell_4 : \{y_2 = (y_1+1) \cdot (y_1+1) \wedge y_3 = 2 \cdot y_1+1 \wedge y_1 \cdot y_1 \leq x \wedge x < y_2\}$
 $z := y_1;$
 $\ell_5 : \{y_2 = (y_1+1) \cdot (y_1+1) \wedge y_3 = 2 \cdot y_1+1 \wedge y_1 \cdot y_1 \leq x \wedge x < y_2 \wedge z = y_1 \wedge z \cdot z \leq x \wedge x < (z+1) \cdot (z+1)\}$

Algorithme 8: *squareroot* annotée

$al0l_1 \triangleq pc = "l0" \wedge y'_1 = 0 \wedge y'_2 = 1 \wedge y'_3 = 1 \wedge pc' = "l1" \wedge z' = z$
 $al1l_2 \triangleq pc = "l1" \wedge y_2 \leq x \wedge pc' = "l2" \wedge \text{UNCHANGED } \langle y_1, y_2, y_3, z \rangle$
 $al1l_4 \triangleq pc = "l1" \wedge y_2 > x \wedge pc' = "l4" \wedge \text{UNCHANGED } \langle y_1, y_2, y_3, z \rangle$
 $al2l_3 \triangleq pc = "l2" \wedge y'_1 = y_1 + 1 \wedge y'_2 = y_2 + y_3 + 2 \wedge y'_3 = y_3 + 2 \wedge pc' = "l3" \wedge z' = z$
 $al3l_2 \triangleq pc = "l3" \wedge y_2 \leq x \wedge pc' = "l2" \wedge \text{UNCHANGED } \langle y_1, y_2, y_3, z \rangle$
 $al3l_4 \triangleq pc = "l3" \wedge y_2 > x \wedge pc' = "l4" \wedge \text{UNCHANGED } \langle y_1, y_2, y_3, z \rangle$
 $al4l_5 \triangleq pc = "l4" \wedge z' = y_1 \wedge pc' = "l5" \wedge \text{UNCHANGED } \langle y_1, y_2, y_3 \rangle$
 $Init \triangleq y_1 = 0 \wedge y_2 = 0 \wedge y_3 = 0 \wedge z = 0 \wedge pc = "l0"$
 $Next \triangleq al0l_1 \vee al1l_2 \vee al1l_4 \vee al2l_3 \vee al3l_2 \vee al3l_4 \vee al4l_5$

$MAX \triangleq 32768$ 16 bits

$D \triangleq 0..32768$

$x \leq 32760$

$Safety_absence \triangleq (y_1 \in D) \wedge (y_2 \in D) \wedge (y_3 \in D) \wedge (z \in D)$

$i \triangleq$

$\wedge pc = "l0" \Rightarrow y_1 \in D \wedge y_2 \in D \wedge y_3 \in D \wedge z \in D$
 $\wedge pc = "l1" \Rightarrow y_2 = (y_1 + 1) \cdot (y_1 + 1) \wedge y_3 = 2 \cdot y_1 + 1 \wedge y_1 \cdot y_1 \leq x \wedge Safety_absence$
 $\wedge pc = "l2" \Rightarrow y_2 = (y_1 + 1) \cdot (y_1 + 1) \wedge y_3 = 2 \cdot y_1 + 1 \wedge y_1 \cdot y_1 \leq x \wedge y_2 \leq x \wedge Safety_absence$
 $\wedge pc = "l3" \Rightarrow y_2 = (y_1 + 1) \cdot (y_1 + 1) \wedge y_3 = 2 \cdot y_1 + 1 \wedge y_1 \cdot y_1 \leq x \wedge Safety_absence$
 $\wedge pc = "l4" \Rightarrow y_2 = (y_1 + 1) \cdot (y_1 + 1) \wedge y_3 = 2 \cdot y_1 + 1 \wedge y_1 \cdot y_1 \leq x \wedge x < y_2 \wedge Safety_absence$
 $\wedge pc = "l5" \Rightarrow y_2 = (y_1 + 1) \cdot (y_1 + 1) \wedge y_3 = 2 \cdot y_1 + 1 \wedge z \cdot z \leq x \wedge x < (z + 1) \cdot (z + 1) \wedge Safety_absence$

$Safety_partialcorrectness \triangleq pc = "l5" \Rightarrow \wedge y_2 = (y_1 + 1) \cdot (y_1 + 1)$
 $\wedge y_3 = 2 \cdot y_1 + 1$
 $\wedge z \cdot z \leq x \wedge x < (z + 1) \cdot (z + 1)$

Exercice 12

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

$\forall v, v'. P_\ell(v) \wedge cond_{\ell, \ell'}(v) \wedge v' = f_{\ell, \ell'}(v) \Rightarrow P_{\ell'}(v')$. Vous devez répondre en énonçant et en démontrant les Conditions de vérification.

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

On suppose que x_1 et x_2 sont des constantes.

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

Algorithme 9: Algorithme de l'exponentiation indienne annoté

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 :

$$\forall x, y, q, r, x', y', q', r'. P_\ell(y_1, y_2, y_3, z) \wedge \text{cond}_{\ell, \ell'}(y_1, y_2, y_3, z) \wedge (y'_1, y'_2, y'_3, z') = f_{\ell, \ell'}(y_1, y_2, y_3, z) \Rightarrow P_{\ell'}(y'_1, y'_2, y'_3, z')$$

Enoncer et vérifier cette propriété pour les paires d'étiquettes suivantes : $(\ell_0, \ell_1); (\ell_1, \ell_2); (\ell_3, \ell_4); (\ell_6, \ell_7); (\ell_7, \ell_8); (\ell_1, \ell_9); (\ell_9, \ell_{10})$.

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 postcondition.

Exercice 14 La numération en base 10 repose sur la décomposition d'un nombre entier naturel n selon les puissances de dix. Ainsi, si $n = \sum_{i=0}^p a_i \cdot 10^i$ où pour toutes les valeurs possibles de $i \in 0..p$, $a_i \in 0..9$ et si $n \neq 0$, alors $a_p \neq 0$ et le nombre n s'écrit alors $a_p \dots a_0$.

Par exemple, on écrira :

$$\text{— } 23 = 2 \cdot 10^1 + 3 \cdot 10^0$$

$$\text{— } 456 = 4 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0$$

Pour chaque entier naturel n , on définit alors la dimension de n par la valeur $p+1$ et on notera cette fonction $\text{dim} \in \mathbb{N} \rightarrow \mathbb{N}$:

$$\text{— } \text{dim}(23) = 2$$

$$\text{— } \text{dim}(456) = 3$$

Puisque $0 = 0 \cdot 10^0$, alors $\text{dim}(0) = 1$. 0 est un cas particulier qu'il faudra gérer correctement. Pour calculer cette fonction, le programme C suivant est proposé :

```
#include <stdio.h>
int main()
{
    int n, count=0;
    printf("Enter_an_integer:_");
    scanf("%d", &n);
    while(n!=0)
    {
        n/=10;
        ++count;
    }
    printf("Number_of_digits:_%d", count);
}
```

Ce programme C peut être simplifié sous la forme de la version algorithmique suivante :

```
—algorithm count{
variables n, count=0;
{
/*l0:
n:=input;
```

```

while (n#0)
{
  n:= n \div 10;
  count := count +1;
};
print <<"Number of digits: %d",count>>;
/*lf:
}}}}

```

Question 14.1 Donner la valeur calculée pour $n = 0$, $n = 1$, $n = 10$ et $n = 123$.

Question 14.2 Donner la précondition et la postcondition de cet algorithme en vue de montrer sa correction partielle.

Question 14.3 Deux étiquettes ont été placées dans le texte de cet algorithme et désignent respectivement l'entrée (l0) et la sortie (lf). On décide de compléter cette annotation et de produire un module TLA^+ modélisant l'exécution de cet algorithme. La variable de contrôle est pc. Donner la méthode pour montrer avec l'outil de TLA^+ que cet algorithme est partiellement correct.

On ne vous demande pas d'écrire le module TLA^+ mais de considérer que vous avez Init et Next.

Question 14.4 Donner la méthode pour montrer avec l'outil de TLA^+ que cet algorithme est sans erreur à l'exécution. En particulier, on prendra soin de préciser par une analyse une borne possible pour le cas où les entiers sont codés sur b bits avec b non nul.