Cours MCFSI
. . .
ModÃ©lisation, v'erification et validation de modèles
Dominique Méry
21 octobre 2024

Exercice 1 *(alg-maxtwonumbers)*
Soit le contrat suivant annoté qui calcule le maximum de deux entiers naturels x_0 *et* y_0

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
  $\ell_3 : \{x \geq y \wedge x = x_0 \wedge y = y_0 \wedge z = z_0 \wedge x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z}\}$
  $Z := X;$
  $\ell_4 : \{x \geq y \wedge x = x_0 \wedge y = y_0 \wedge x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z} \wedge z = x_0\}$
;
$\ell_5 : \{z = max(x_0, y_0) \wedge x = x_0 \wedge y = y_0 \wedge x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z}\}$

Algorithme 1: maximum de deux nombres non annotée

Question 1.1 *Traduire l'automate de cet algorithme sous la forme d'une machine modifiant les variables* x, y, z, pc.

Question 1.2 *Valider la traduction en simulant quelques*

Question 1.3 *Ajouter les annotations et les pré et post conditions.*

Question 1.4 *Vérifier la correction partielle et l'absence d'erreurs à l'exécution.*

Exercice 2 *Show that each annotation is sound or unsound with respect to the proof obligations :*
$\forall x, y, , x', y'.P_\ell(x, y) \wedge cond_{\ell,\ell'}(x, y) \wedge (x', y') = f_{\ell,\ell'}(x, y) \Rightarrow P_{\ell'}(x', y')$
You will use a context and a machine for expressing these conditions.

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

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

— *We assume that* p *is a prime number.*
$\ell_1 : x = 2^p \wedge y = 2^{p+1} \wedge x\cdot y = 2^{2\cdot p+1}$
$x := y+x+2^x$
$\ell_2 : x = 5\cdot 2^p \wedge y = 2^{p+1}$

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

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**
  $\ell_1 : \{O < x \leq x_0 \wedge x_0 \in \mathbb{N}\}$
  $x := x{-}1;$
  $\ell_2 : \{0 \leq x \leq x_0 \wedge x_0 \in \mathbb{N}\}$
;
$\ell_3 : \{x = 0\}$

Algorithme 2: Exercice 3

Exercice 3 *(alg-simple)*
Let the following partially annotated algorithm :

Question 3.1 *Translate each transition ℓ, ℓ' into an event modifying the variables according to the statements.*

Question 3.2 *Define an invariant attaching to each label an assertion satisfied at the control point.*

Question 3.3 *Verify proof obligations and deduce that the algorithm is partially correct.*

Question 3.4 *Prove that the algorithm has no runtime error.*

Exercice 4 *(alg-squareroot)*
Let the following annotated invariant.

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 3: *squareroot* annotée Exercice 4

Question 4.1 *Translate each transition ℓ, ℓ' into an event modifying the variables according to the statements.*

Question 4.2 *Define an invariant attaching to each label an assertion satisfied at the control point.*

Question 4.3 *Verify proof obligations and deduce that the algorithm is partially correct.*

Question 4.4 *Prove that the algorithm has no runtime error.*

Exercice 5 *(alg-maximum)*
Soit l'algorithme suivant annoté partiellement :

Question 5.1 *Translate each transition ℓ, ℓ' into an event modifying the variables according to the statements.*

Question 5.2 *Define an invariant attaching to each label an assertion satisfied at the control point.*

Question 5.3 *Verify proof obligations and deduce that the algorithm is partially correct.*

Question 5.4 *Prove that the algorithm has no runtime error.*

Exercice 6 *()*
Cet exercice comprend plusieurs questions indépendantes. Il s'agit d'écrire un événement Event-B qui modélise une transformation décrite en langue naturelle.

Question 6.1 *Ecrire les conditions de vérification correspondant amcfsi4ux événements de la machine M du fichier mcfsi4-a1.pdf et du fichier mcfsi4-a2.pdf.*

Question 6.2 *Montrer que chaque annotation est correcte ou incorrecte selon les conditions de vérifications énoncées comme suit*
$\forall x, y, , x', y'. P_\ell(x, y) \ \wedge \ cond_{\ell,\ell'}(x, y) \ \wedge \ (x', y') = f_{\ell,\ell'}(x, y) \Rightarrow P_{\ell'}(x', y')$
mais utiliser une machine et un contexte pour faire le travail.

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

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

Exercice 7 *Soit l'algorithme suivant annoté partiellement :*

Question 7.1 *Traduire chaque transition ℓ, ℓ' par un événement transformant les variables.*

Question 7.2 *Définir un invariant associant à chaque étiquette une assertion satisfaite à ce point de contrôle.*

Question 7.3 *Vérifier toutes les conditions et en déduire que l'algorithme est partiellement correct.*

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

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

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

$m := f(0);$

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

$i := 1;$

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

while $i < n$ **do**

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

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

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

$m := f(i);$

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

;

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

$i{+}{+};$

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

;

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

Algorithme 4: Algorithme du manimum d'une liste annoté Exercice 5

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 5: exemple annoté

Variables : X,Y1,Y2,Y3,Z
Requires : $x_0 \in \mathbb{N}$
Ensures : $z_f^2 \leq x_0 \wedge x_0 < (z_f{+}1)^2$

$\ell_0 : \{x_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z} \wedge y1_0 \in \mathbb{Z} \wedge y2_0 \in \mathbb{Z} \wedge y3_0 \in \mathbb{Z} \wedge (x, y1, y2, y3, z) = (x_0, y1_0, y2_0, y3_0, z_0)\}$
$(y_1, y_2, y_3) := (0, 1, 1)$;
$\ell_1 : \{x_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z} \wedge y1_0 \in \mathbb{Z} \wedge y2_0 \in \mathbb{Z} \wedge y3_0 \in \mathbb{Z} \wedge y2 = (y1{+}1)\cdot(y1{+}1) \wedge y3 = 2\cdot y1{+}1 \wedge y1\cdot y1 \leq x \wedge (x, z) = (x_0, , z_0)\}$
while $y_2 \leq x$ **do**
  $\ell_2 : \{\ldots\}$
  $(y_1, y_2, y_3) := (y_1{+}1, y_2{+}y_3{+}2, y_3{+}2)$;
  $\ell_3 : \{\ldots\}$
;
$\ell_4 : \{\ldots\}$
$z := y_1$;
$\ell_5 : \{\ldots\}$

Algorithme 6: *squareroot* partiellement annotée

Question 7.4 *Démontrer que cet algorithme est sans erreurs à l'exécution en prenant soin de chosir des ensembles informatiques.*

Exercice 8 *Soit l'algorithme suivant annoté partiellement :*

Question 8.1 *Traduire chaque transition ℓ, ℓ' par un événement transformant les variables.*

Question 8.2 *Définir un invariant associant à chaque étiquette une assertion satisfaite à ce point de contrôle.*

Question 8.3 *Vérifier toutes les conditions et en déduire que l'algorithme est partiellement correct.*

Exercice 9 *On considère le problème du contrôle d'accès et l'archive du projet sys-accesscontrol.zip. L'objectif est de valider le modèle obtenu.*

Question 9.1 *Modifier les données abstraites du problèmes en les instanciant par des valeurs : P, B, A, ...*

Question 9.2 *Utiliser ProB pour vérifier que les invariants et les propriétés de sûreté sont satisfaites sur ces instances. En particulier, on montrera l'absence de blocage.*

Question 9.3 *En raffinant une fois de plus le modèle control5, introduire la notion de timer pour prendre en compte les deux cas de 2 secondes et de trente secondes.*

Les exercices complémentaires sont donnés dans la suite et sont corrigés sous forme d'archives.

Exercice 10 *Soit l'algorithme suivant annoté partiellement :*

Question 10.1 *Traduire chaque transition ℓ, ℓ' par un événement transformant les variables.*

Question 10.2 *Définir un invariant associant à chaque étiquette une assertion satisfaite à ce point de contrôle.*

Question 10.3 *Vérifier toutes les conditions et en déduire que l'algorithme est partiellement correct.*

You will find a list of exercices that you can try to solve but they will not be solved during the tutorials.

Exercice 11 *A set of users have a controlled access to resources and they can or they can not use a resource in different modes. For instance, in a system you can use a memory according to two modes at least read or write. Model a access control system of a set of users to a set of resources where each resource is possibly used in a given mode. When a user has not get the access in a mode, he or she can not use the resource. We assume that the access rights are possibly modified by request to an authority. The following events may be observed on the system :*

- *access to a authorized resource by a user*
- *adding an authorization to a user for a given resource*
- *removing an authorization to a user for a given resource*

/* 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 : $i \in \mathbb{Z}$

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

$m := f(0);$

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

$i := 1;$

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

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**
>
> > $\ell_4 : \{ \begin{pmatrix} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0 \,..\, n-1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in 1..n-1 \wedge \begin{pmatrix} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i-1]) \wedge \\ (\forall j \cdot j \in 0 \,..\, i-1 \Rightarrow f(j) \leq m) \end{pmatrix} \wedge$
> > $f(i) > m \}$
> >
> > $m := f(i);$
> >
> > $\ell_5 : \{ \begin{pmatrix} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0 \,..\, n-1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in 1..n-1 \wedge \begin{pmatrix} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i]) \wedge \\ (\forall j \cdot j \in 0 \,..\, i \Rightarrow f(j) \leq m) \end{pmatrix} \}$
>
> ;
>
> $\ell_6 : \{ \begin{pmatrix} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0 \,..\, n-1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in \mathbb{Z} \wedge \wedge i \in 1..n-1 \wedge \begin{pmatrix} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i]) \wedge \\ (\forall j \cdot j \in 0 \,..\, i \Rightarrow f(j) \leq m) \end{pmatrix} \}$
>
> $i{+}{+};$
>
> $\ell_7 : \{ \begin{pmatrix} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0 \,..\, n-1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in 2..n \wedge \begin{pmatrix} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i-1]) \wedge \\ (\forall j \cdot j \in 0 \,..\, i-1 \Rightarrow f(j) \leq m) \end{pmatrix} \}$

;

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

Algorithme 7: Algorithme du manimum d'une liste annoté

Exercice 12 *Soient les trois variables x, y, z entières. On se donne une relation $x+y+z = N$ où N est une constante entière. On note $I(x, y, z)$ la propriété suivante :*

$$\begin{array}{l} x \in \mathbb{Z} \\ y \in \mathbb{Z} \\ z \in \mathbb{Z} \\ x+y+z = N \end{array}$$

Question 12.1 *On définit l'événement e1 définit comme suit :*

EVENT *e1*
  WHEN
    $x \leq -5$
  THEN
    $x := x+2$
    $y, z : |(x+y'+z'+2 = N)$
  END

Ecrire la condition de vérification exprimant la préservation de l'invariant I(x,y,z) par l'événement e1

Question 12.2 *On définit l'événement e2 définit comme suit :*

EVENT *e2*
  WHEN
    $y \geq -5$
    $y \leq 5$
  THEN
    $x, y, z : |(y' = y+1 \wedge x'+y'+z' = N)$
  END

Ecrire la condition de vérification exprimant la préservation de l'invariant I(x,y,z) par l'événement e2.

Exercice 13 *()*
Soit une table $t0$ de $n0$ valeurs entières. Ecrire une spécification événementielle décrivant le calcul du nombre de valeurs supérieures à une valeur donnée B.

Question 13.1 *Ecrire un contrat caractérisant ce calcul.*

Question 13.2 *Ecrire le contrat en Event-B.*

Exercice 14 *()*
Soit le contrat suivant :

requires $pre(v_0)$
ensures $post(v_0, v_f)$
variables V
constantes C
  begin
  $\ell_1 : Q_1(v_0, v)$
  $V := F(V, C)$
  $\ell_2 : Q_2(v_0, v)$
  end

Question 14.1 *On suppose que V est de type Int et que C est une constante de type Int satisfaisant une condition* $R(C)$ *Rappeler la liste des conditions de vérification.*

Question 14.2 *Traduire ce contrat sous la forme d'un contexte et d'une machine Event-B.*

Question 14.3