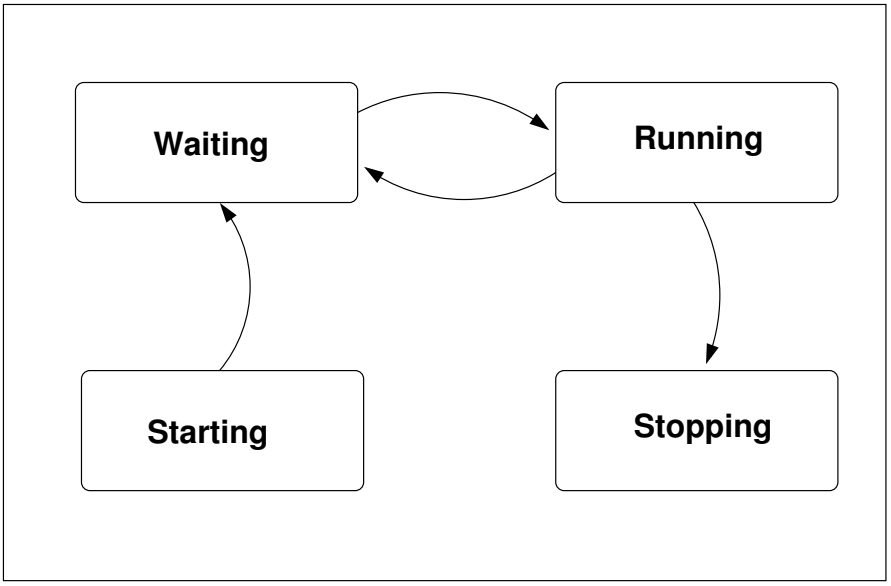
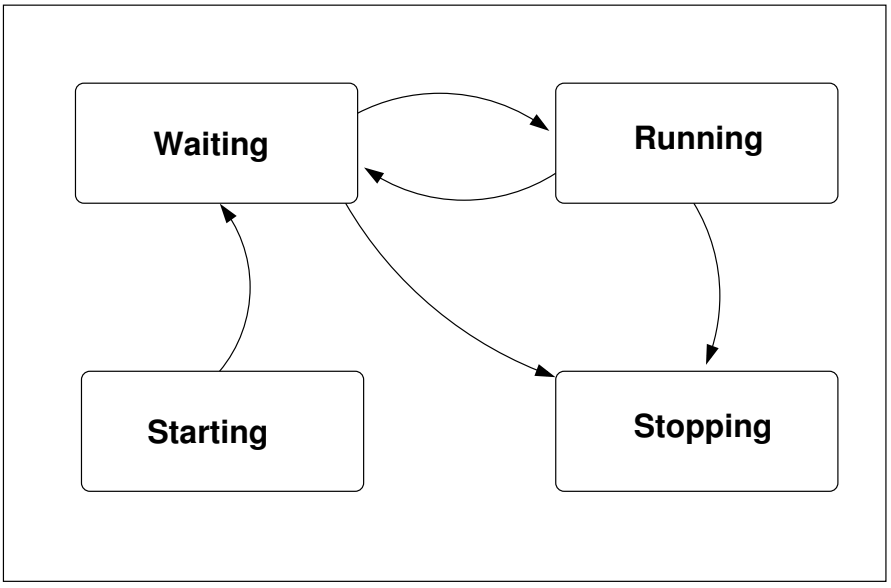
Cours MCFSI
...
Spécifier un problème en Event-B
Dominique Méry
1er octobre 2024

Exercice 1 *mcfsi1-ex1-tut1.zip*
Traduire cette machine sous la forme d'une machine Event-B.



Exercice 2 *mcfsi1-ex2-tut1.zip*
Traduire cette machine sous la forme d'une machine Event-B.



Exercice 3 *(mcfsi1-simple)*
Soient deux ensembles A et B qui sont des parties de U.

- *Ecrire un modèle Event-B qui utilise deux variables v et w deux sous-ensembles de A et B*
- *Ajouter une fonction partielle de A dans B.*

— *Définir un événement e1 qui transfert un élément de A dans B s'il n'est pas dans A.*
— *Définir un événement e2 qui crée un lien entre un élément de A et un élément de B.*

Exercice 4 *(mcfsi1-variant)*
Un système permet de réaliser la somme de deux nombres x0 et y0 en ajoutant une unité à une variable z. Il comprend un événement incx2z qui décroit la valeur de x d'une unité et quio augmente la valeur de z de une unité et un événement incy2z qui décroît y d'une unité et qui augmente z d'une unité. Le processus global s'arrête quand les deux variables x et y sont nulles. Ecrire un modèle Event-B qui modélise ce système.

Exercice 5 *(mcfsi1-summation)*
Soit une suite de valeurs entières $v_1, \ldots, v_n$ où le nombre n est fixé. Ecrire une spécification événementielle décrivant le calcul de la somme des éléments de cette suite. Pour cela, vous devez décrire les données puis l'événement magique qui réalise ce calcul.

Exercice 6 *(mcsfi-ressources-pb1),*
Modéliser les problèmes suivants.

Question 6.1 *(mcsfi-ressources-pb1)*
On suppose disposer de ressources qui sont partagées par un ensemble de processus. Si un processus a besoin d'une ressource, il demande cette ressource et s'il n'a plus besoin de cette ressource, il la rend. Un processus peut utiliser plusieurs ressources à la fois mais une ressource ne peut pas être utilisée par deux processus à la fois.

Question 6.2 *(mcsfi-ressources-pb2)*
On suppose disposer de ressources qui sont partagées par un ensemble de processus. Si un processus a besoin d'une ressource, il demande cette ressource et s'il n'a plus besoin de cette ressource, il la rend. Un processus ne peut utiliser qu'une seule ressources à la fois et une ressource ne peut pas être utilisée par deux processus à la fois.

Exercice 7 *(mcfsi1-invariantssafety)*
Nous considérons le modèle suivant.

MACHINE $M1$
VARIABLES
x
INVARIANTS
...
EVENTS
EVENT *INITIALISATION*
BEGIN
$act1 : x := -10$
END
EVENT *evt1*
WHEN
$grd1 : x \geq -1$
THEN
$act1 : x := x+1$
END
EVENT *evt2*
WHEN
$grd1 : x \leq -1$
$grd2 : x \geq -44$
THEN
$act1 : x := x-1$
END
END

On considère plusieurs cas pour l'invariant.

Question 7.1 *(M1)*

$inv1 : x \in \mathbb{Z}$
$inv3 : x \leq -1$

Est-ce que toutes les conditions de vérification sont prouvées par le prouveur de l'application Rodin ? Expliquez clairement pourquoi elles sont prouvées ou non.

Question 7.2 *(M2)*

$inv1 : x \in \mathbb{Z}$
$inv3 : x \leq -3$

Est-ce que toutes les conditions de vérification sont prouvées par le prouveur de l'application Rodin ? Expliquez clairement pourquoi elles sont prouvées ou non.

Question 7.3 *(M3)*

$inv1 : x \in \mathbb{Z}$
$inv4 : -45 \leq x \wedge x \leq -10$

Est-ce que toutes les conditions de vérification sont prouvées par le prouveur de l'application Rodin ? Expliquez clairement pourquoi elles sont prouvées ou non.

Question 7.4 *(M4)*

$inv1 : x \in \mathbb{Z}$
$inv3 : x \leq -3$
$inv4 : -45 \leq x \wedge x \leq -10$
$inv2 : x \leq -1$

Est-ce que toutes les conditions de vérification sont prouvées par le prouveur de l'application Rodin ? Expliquez clairement pourquoi elles sont prouvées ou non.

Exercice 8 *(alg-maxtwonumbers)*
Soit le contrat suivant annoté qui calcule le maximum de deux entiers naturels x_0 *et* y_0

Question 8.1 *Traduire l'automate de cet algorithme sous la forme d'une machine modifiant les variables* x, y, z, pc.

Question 8.2 *Valider la traduction en simulant quelques*

Question 8.3 *Ajouter les annotations et les pré et post conditions.*

Question 8.4 *Vérifier la correction partielle et l'absence d'erreurs à l'exécution.*

Exercice 9 *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.

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

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

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

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

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

Exercice 10 *(alg-simple)*
Let the following partially annotated algorithm :

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 **??**

Question 10.1 *Translate each transition ℓ, ℓ' into an event modifying the variables according to the statements.*

Question 10.2 *Define an invariant attaching to each label an assertion satisfied at the control point.*

Question 10.3 *Verify proof obligations and deduce that the algorithm is partially correct.*

Question 10.4 *Prove that the algorithm has no runtime error.*

Exercice 11 *(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 **??**

Question 11.1 *Translate each transition ℓ, ℓ' into an event modifying the variables according to the statements.*

Question 11.2 *Define an invariant attaching to each label an assertion satisfied at the control point.*

Question 11.3 *Verify proof obligations and deduce that the algorithm is partially correct.*

Question 11.4 *Prove that the algorithm has no runtime error.*

Exercice 12 *(alg-maximum)*
Soit l'algorithme suivant annoté partiellement :

Question 12.1 *Translate each transition ℓ, ℓ' into an event modifying the variables according to the statements.*

Question 12.2 *Define an invariant attaching to each label an assertion satisfied at the control point.*

Question 12.3 *Verify proof obligations and deduce that the algorithm is partially correct.*

Question 12.4 *Prove that the algorithm has no runtime error.*

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

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

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 : $i \in \mathbb{Z}$

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

$m := f(0)$;

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

$i := 1$;

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

while $i < n$ **do**

- $\ell_3 : \{\left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0\,..\,n{-}1 \to \mathbb{N} \end{array}\right) \wedge i \in 1..n{-}1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i{-}1]) \wedge \\ (\forall j \cdot j \in 0\,..\,i{-}1 \Rightarrow f(j) \leq m) \end{array}\right)\}$
- **if** *f(i)>m* **then**
  - $\ell_4 : \{\left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0\,..\,n{-}1 \to \mathbb{N} \end{array}\right) \wedge i \in 1..n{-}1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i{-}1]) \wedge \\ (\forall j \cdot j \in 0\,..\,i{-}1 \Rightarrow f(j) \leq m) \end{array}\right) \wedge f(i) > m\}$
  - $m := f(i)$;
  - $\ell_5 : \{\left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0\,..\,n{-}1 \to \mathbb{N} \end{array}\right) \wedge i \in 1..n{-}1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i]) \wedge \\ (\forall j \cdot j \in 0\,..\,i \Rightarrow f(j) \leq m) \end{array}\right)\}$
- ;
- $\ell_6 : \{\left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0\,..\,n{-}1 \to \mathbb{N} \end{array}\right) \wedge i \in \mathbb{Z} \wedge \wedge i \in 1..n{-}1 \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i]) \wedge \\ (\forall j \cdot j \in 0\,..\,i \Rightarrow f(j) \leq m) \end{array}\right)\}$
- $i{+}{+}$;
- $\ell_7 : \{\left(\begin{array}{l} n \in \mathbb{N} \wedge \\ n \neq 0 \wedge \\ f \in 0\,..\,n{-}1 \to \mathbb{N} \end{array}\right) \wedge i \in 2..n \wedge \left(\begin{array}{l} m \in \mathbb{N} \wedge \\ m \in ran(f[0..i{-}1]) \wedge \\ (\forall j \cdot j \in 0\,..\,i{-}1 \Rightarrow f(j) \leq m) \end{array}\right)\}$

;

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

Algorithme 4: Algorithme du manimum d'une liste annoté Exercice **??**