Tutorial Modelling Software-based Systems
...
Tutorial 5 : Developing systems by refinement
Dominique Méry
20 novembre 2024

Exercice 1 *(sys-swp,sys-abp))*

Question 1.1 *Develop the alternating bit protocol using the Event-B modelling language and methodology.*

Question 1.2 *The sliding window protocol is generalizing the ABP protocol by using a window of size 1 or more and it communicates through the window the data from a file. Develop a full Event-B model for deriving the sliding window protocol.*

Exercice 2 *Two agents A and B communicate data through a communication network. The communication network should support the transfer of a file f of size n.*

Question 2.1 *Model the service of the protocol.*

Question 2.2 *Using the refinement, develop a solution which is reliable in an environment which is loosing messages.*

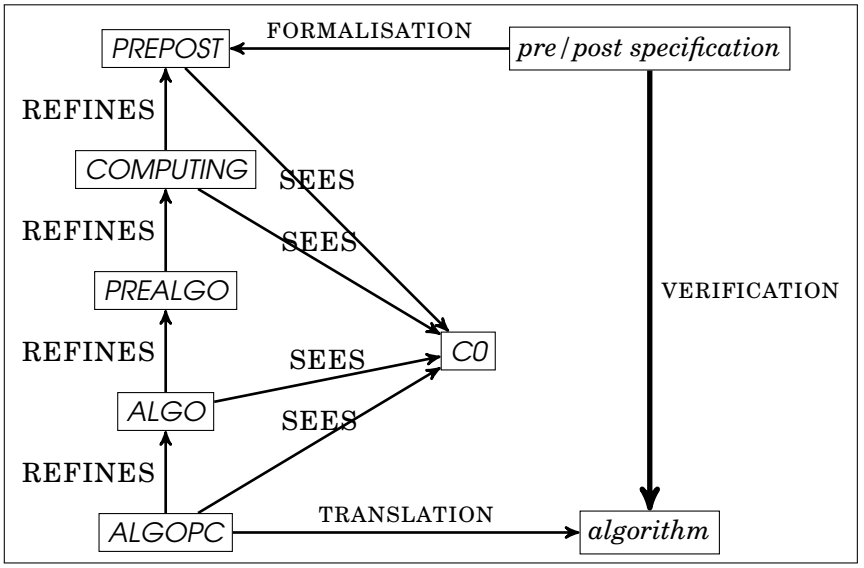
Exercice 3 *ex-ccordinationcomputing*
Le modèle de coordination est un modèle de programmation parallèle qui est fondé sur un espace de partage d'informations appelé tuple space et sur des primitives de communication via ce tuple space ; deux types de primitives sont fournies dans ce modèle, d'une part le dépôt d'un tuple dans l'espace tuple space et d'autre part le retrait d'un tuple du tuple space avec soit un nre ou la consultation de l'espace des tuples. Dans ce modèle un programme est une structure constituée de processus qui communiquent via le tuple space et qui sont écrits dans un langage de programmation donné comme C, C++, ML etc Dans ce modèle de programmation par coordination, on distingue donc deux langages de processus :

- *le langage des processus de calculs ou tâches*
- *le langage de coordination des tâches.*

Soit une suite t de n valeurs de type T1 et soit une fonction $f \in T1 \longrightarrow T2$. Développer un modèle fondé sur le modèle de coordination. Pour cela, on définira un contexte pour les données et une machine de spécification puis un raffinement introduisant le modèle de coordination.

Exercice 4 *ex-ccordinationmatrix*
Dans le cadre du modèle de programmation par coordination, développer une solution du calcul du produit de matrices. On rappelle que le produit de deux matrices est défini comme suit :
$\forall i, j.i \in 1..n \wedge j \in 1..m \Rightarrow \sum_{k=0}^{k=m} A(i,k) \cdot B(k,j)$

Exercice 5 *Patron*



L'archive est sur le site arche sous le nom iterativepattern.zip.

Question 5.1 *Appliquer ce patron pour calculer la valeur de n^2 avec la suite $(n+1)^2 = n^2+n+n+1$. Ecrire une fonction C que vous vérifierez avec Frama-c.*

Question 5.2 *Appliquer ce patron pour rechercher l'indice i de t tel que $t(i) = v$. Ecrire une fonction C que vous vérifierez avec Frama-c.*

Exercice 6 *Appliquer le patron pour le cas du calcul de x^3 en utilisant $(i+1)^3 = i^3+3i^2+3i+1$. Nous utilisons en fait ces suites :*

- $z_0 = 0$ *et* $\forall n \in \mathbb{N} : z_{n+1} = z_n+v_n+w_n$
- $v_0 = 0$ *et* $\forall n \in \mathbb{N} : v_{n+1} = v_n+t_n$
- $t_0 = 3$ *et* $\forall n \in \mathbb{N} : t_{n+1} = t_n+6$
- $w_0 = 1$ *et* $\forall n \in \mathbb{N} : w_{n+1} = w_n+3$
- $u_0 = 0$ *et* $\forall n \in \mathbb{N} : u_{n+1} = u_n+1$

$$\begin{pmatrix} z(i+1) \\ v(i+1) \\ t(i+1) \\ w(i+1) \\ u(i+1) \end{pmatrix} = \begin{pmatrix} z_i+v_i+w_i \\ v(i)+t(i) \\ t(i)+6 \\ w(i)+3 \\ u(i)+1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 01 & 0 & \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} z(i) \\ v(i \\ t(i) \\ w(i) \\ u(i) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 6 \\ 3 \\ 1 \end{pmatrix}$$

Ecrire une fonction C que vous vérifierez avec Frama-c.

Exercice 7 *(sys-ga)*
We consider the general problem of access control with the administration of access rights.

1. *Model the access control problem in the case of the access of persons in buildings. We assume that the rights are given and are not modified.*
2. *Model the access control problem by adding specific actions for administrating access rights.*

Exercice 8 *(alg-squareroot)*
Let the following annotated invariant.

Question 8.1 *Translate each transition ℓ, ℓ' into an event modifying the variables according to the statements.*

Question 8.2 *Define an invariant attaching to each label an assertion satisfied at the control point.*

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 1: $squareroot$ annotée Exercice 8

Question 8.3 *Verify proof obligations and deduce that the algorithm is partially correct.*

Question 8.4 *Prove that the algorithm has no runtime error.*

Exercice 9 Question 9.1 *Modéliser l'accès en lecture ou écriture à des documents classifiés selon les catégories suivantes* $\{public, prive, secret\}$.

Question 9.2 *Modéliser l'accès à des fichiers selon le modèle UNIX.*