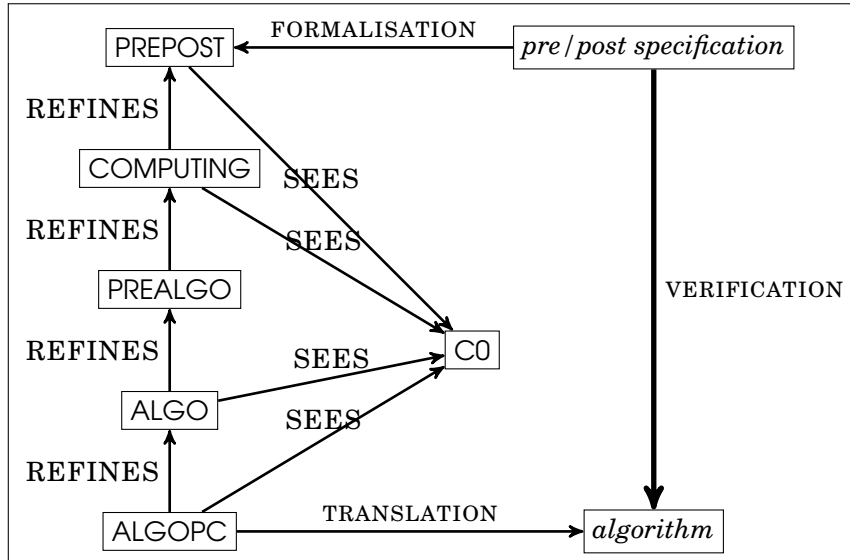Tutorial Modelling Software-based Systems
· · ·
Tutorial 2 : Designing and verifying sequential algorithms using the Event-B modelling
language
Dominique Méry
24 octobre 2025

We apply the following pattern for developing sequential algorithms from a contract.

Exercice 1 *fx1-tut2.zip*
We consider a finite sequence of integers $v_1, \ldots, v_n$ where n is the length of the sequence and is supposed to be fixed. Write an Event B specification modelling the computation of the value of the summation of the sequence v. You should define cerafully v, n and the summation of a finite sequence of integers.

Exercice 2 *fx2-tut2.zip*
The Rodin archive is on Arche repository as m iterativepattern.zip.

Question 2.1 *Apply the pattern for computing the value n^2 using the sequence $(n+1)^2 = n^2+n+n+1$.*

Question 2.2 *Write a C function with annotation that you will check with Frama-c.*

Exercice 3 *fx3-tut2.zip*
Develop an algorithmic solution with the boss for the problem of finding the number of occurrences of a value v satisfying a condition CO in a table t of dimension n. We assume that the table is a value in a set V and that CO is a subset of V.

Exercice 4 *fx4-tut2.zip*
Apply this pattern to find the index i of t such that $t(i) = v$.

Exercice 5 *fx5-tut2.zip*
Apply the pattern for the case of calculating x^3 by using $(i+1)^3 = i^3+3i^2+3i+1$. We are actually using these sequences :
   — $z_0 = 0$ *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 6 *fx6-tut2.zip*

The primitive recursive functions are defined by initial functions (the 0-place zero function ζ, the k-place projection function $\pi_i{}^k$, the successor function σ) and by two combining rules, namely the composition rule and the primitive recursive rule. More precisely, we give the definition of functions and rules :

— $\zeta() = 0$
— $\forall i \in \{1, \ldots, k\} : \forall x_1, \ldots, x_k \in \mathbb{N} : \pi_i{}^k(x_1, \ldots, x_k) = x_i$
— $\forall x \in \mathbb{N} : \sigma(n) = n+1$
— *If g is a l-place function, if $h_1, \ldots, h_l$ are n-place functions and if the function f is defined by :*
  $\forall x_1, \ldots, x_n \in \mathbb{N} : f(x_1, \ldots, x_n) = g(h_1(x_1, \ldots, x_n), \ldots, h_l(x_l, \ldots, x_n))$,
  then f is obtained from g and $h_1, \ldots, h_l$ by composition.
— *If g is a l-place function, if h is a $(l+2)$-place function and if the function f is defined by :* $\forall x_1, \ldots, x_l, x \in \mathbb{N}!$,
  $$\begin{cases} f(x_1, \ldots, x_l, 0) & = & g(x_1, \ldots, x_l) \\ f(x_1, \ldots, x_l, x+1) & = & h(x_1, \ldots, x_l, x, f(x_1, \ldots, x_l, x)) \end{cases},$$

  then f is obtained from g and h by primitive recursion.

A function f is primitive recursive, if it is an initial function or can be generated from initial functions by some finite sequence of the operations of composition and primitive recursion.
We summarize the equations defining these functions :
$$\begin{cases} f(x, 0) = g(x) \\ f(x, suc(y)) = h(x, y, f(x, y)) \end{cases}$$
We suppose that g and h are two functions defined as primitive recursive.

Question 6.1 *Write a first model stating the computation of f for given data.*

Question 6.2 *Propose a refinement of the model of the previous question using the two functions g and h.*

Question 6.3 *Apply the iterative paradigm for deriving a recursive algorithm.*