MVSI – Modelling and Verification of Computer Systems

Modelling algorithms in PlusCal (I)

Tutorial

Dominique Méry – 20th November 2026

✎ Exercise 1

`appex4...`

We use the translation of a PlusCal algorithm into a TLA module to check algorithms. For each question, write a TLA module containing an expression of the algorithm, translate it into a module, and then analyse the resulting module with respect to partial correctness and the absence of run-time errors.

Question 1.1 `(appex4_1_1.tla)`

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

Question 1.2 We assume that p is a prime number:

$\ell_1 : x = 2^p \wedge y = 2^{p+1} \wedge x \cdot y = 2^{2 \cdot p+1}$
$x := y+x+2^x$
$\ell_2 : x = 5 \cdot 2^p \wedge y = 2^{p+1}$

Question 1.3

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

Question 1.4

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

✎ Exercise 2

Annotate each algorithm and translate it with the PlusCal to TLA$^+$ translator.

Question 2.1 `(pluscal_loop)`

Listing 1: Fragment

```
          variables x=x0;
{
 while (0<x)
  {
     x := x - 1;
   };
}
}
```

Question 2.2 `(pluscal_max)`

Listing 2: Fragment

```
  variables x=a,y = b,z;
{
 if ( x < y ) {
  z:=y;
  }
```

```
else
{
 z:= x;
 };
 }
}
```

✎ Exercise 3 appex4_3.tla

We consider the following algorithm:

START
$\{x_1 \geq 0 \wedge x_2 > 0\}$
$(y_1, y_2, y_3) \leftarrow (x_1, 0, x_2)$;
while $y_3 \leq y_1$ **do** $y_3 \leftarrow 2y_3$;
while $y_3 \neq x_2$ **do**
  begin $(y_2, y_3) \leftarrow (2y_2, y_3/2)$;
    if $y_3 \leq y_1$ **then** $(y_1, y_2) \leftarrow (y_1 - y_3, y_2 + 1)$
  end;
$(z_1, z_2) \leftarrow (y_1, y_2)$
$\{0 \leq z_1 < x_2 \wedge x_1 = z_2 x_2 + z_1\}$
HALT

Question 3.1 Show that this algorithm is partially correct with respect to its precondition and its postcondition, which you will state. To do so, translate this algorithm into a module from the PlusCal language.

Question 3.2 Show that it is free of run-time errors.

✎ Exercise 4 appex4_4.tla

We consider the following algorithm:

START
$\{x_1 > 0 \wedge x_2 > 0\}$
$(y_1, y_2, y_3, y_4) \leftarrow (x_1, x_2, x_2, 0)$;
while $y_1 \neq y_2$ **do**
  begin while $y_1 > y_2$ **do** $(y_1, y_4) \leftarrow (y_1 - y_2, y_4 + y_3)$;
    while $y_2 > y_1$ **do** $(y_2, y_3) \leftarrow (y_2 - y_1, y_3 + y_4)$
  end;
$(z_1, z_2) \leftarrow (y_1, y_3 + y_4)$
$\{z_1 = gcd(x_1, x_2) \wedge z_2 = scm(x_1, x_2)\}$
HALT

where $scm(x_1, x_2)$ is the smallest common multiple of x_1 and x_2, that is, their product divided by their greatest common divisor.

Question 4.1 Show that this algorithm is partially correct with respect to its precondition and its postcondition, which you will state. To do so, translate this algorithm into a module from the PlusCal language.

Question 4.2 Show that it is free of run-time errors.