MVSI – Modelling and Verification of Computer Systems

# Annotation, contracts, modelling and verification

Tutorial

Dominique Méry – 20th November 2026

## ✎ Exercise 1 `appex3_1.tla`

Consider the following contract:

variables $int\ X, int\ Y, int\ Z$
requires $P(x_0,y_0,z_0)$
ensures $Q(x_0,y_0,z_0,x_f,y_f,z_f)$
begin
// $0 : R_0(x_0,y_0,z_0,x,y,z)$
$\mathtt{X} = \mathtt{g}(\mathtt{X},\mathtt{Y},\mathtt{Z})$
// $f : R_f(x_0,y_0,z_0,x,y,z)$
end

- g is an arithmetic function defined on the type of the integers `int`; it leads to a typing predicate $x,y,z \in \mathbb{Z}$ (or $\mathbb{Z}(x,y,z)$).
- $P(x_0,y_0,z_0)$ defines the precondition, that is, the conditions that the initial values of the variables `X,Y,Z` must satisfy.
- $Q(x_0,y_0,z_0,x_f,y_f,z_f)$ defines the postcondition, that is, the relation that the initial values and the final values of the variables `X,Y,Z` must satisfy.
- $R_0(x_0,y_0,z_0,x,y,z)$ and $R_f(x_0,y_0,z_0,x,y,z)$ define the conditions, or assertions, satisfied by the current values of the variables `X,Y,Z`.

**Question 1.1** We define

- $P(x_0,y_0,z_0) \stackrel{\text{def}}{=} x_0,y_0,z_0 \in \mathbb{Z}$
- $g \stackrel{\text{def}}{=} \lambda u,v,w.u+v+w$
- $Q(x_0,y_0,z_0,x,y,z) \stackrel{\text{def}}{=} x_0,y_0,z_0,x,y,z \in \mathbb{Z} \wedge x = x_0+y_0+z_0 \wedge y = y_0 \wedge z = z_0$
- $R_0(x_0,y_0,z_0,x,y,z) \stackrel{\text{def}}{=} x = x_0 \wedge y = y_0 \wedge z = z_0 \wedge \mathbb{Z}(x_0,y_0,z_0,x,y,z)$
- $R_f(x_0,y_0,z_0,x,y,z) \stackrel{\text{def}}{=} \mathbb{Z}(x_0,y_0,z_0,x,y,z) \wedge x = x_0+y_0+z_0 \wedge y = y_0 \wedge z = z_0$

Write the verification conditions of this contract and check that they hold.

**Question 1.2** We define

- $P(x_0,y_0,z_0) \stackrel{\text{def}}{=} x_0,y_0,z_0 \in \mathbb{Z}$
- $g \stackrel{\text{def}}{=} \lambda u,v,w.max(u,v,w)$
- $Q(x_0,y_0,z_0,x,y,z) \stackrel{\text{def}}{=} x_0,y_0,z_0 \in \mathbb{Z} \wedge x,y,z \in \mathbb{Z} \wedge x = max(x_0,y_0,z_0) \wedge y = y_0 \wedge z = z_0$
- $R_0(x_0,y_0,z_0,x,y,z) \stackrel{\text{def}}{=} \mathbb{Z}(x_0,y_0,z_0,x,y,z)$

Write the verification conditions of this contract and check that they hold. In particular, define a relation $R_f(x_0,y_0,z_0,x,y,z)$ that establishes correctness according to the rules.

## ✎ Exercise 2 `appex3_21.tla`

Show that each annotation is correct or incorrect, according to the verification conditions stated as follows:

$\forall v, v'.P_\ell(v) \wedge cond_{\ell,\ell'}(v) \wedge v' = f_{\ell,\ell'}(v) \Rightarrow P_{\ell'}(v')$
This condition is first written
$\forall v, v', pc, pc'.pc = \ell \wedge P_\ell(v) \wedge pc = \ell \wedge cond_{\ell,\ell'}(v) \wedge v' = f_{\ell,\ell'}(v) \wedge pc' = \ell' \Rightarrow pc' = \ell' \wedge P_{\ell'}(v')$
but it can be reduced by forgetting the variable $pc$. For each question, you will also write a TLA module that checks whether the annotation is correct.

- (appex3_21.tla)

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

- $\ell_1 : x = 1 \wedge y = 12$
  $x := 2 \cdot y$
  $\ell_2 : x = 1 \wedge y = 24$
- We assume that $p$ is a prime number:

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

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

## ✎ Exercise 3

Show that each annotation is correct or incorrect, according to the verification conditions stated as follows:
$\forall x, y, x', y'.P_\ell(x,y) \wedge cond_{\ell,\ell'}(x,y) \wedge (x',y') = f_{\ell,\ell'}(x,y) \Rightarrow P_{\ell'}(x',y')$

- (1)

$\ell_1 : x = 9 \wedge y = z+x$
$y := x+9$
$\ell_2 : x = 9 \wedge y = x+9$

- (3)

$\ell_1 : x = 3 \wedge y = 3$
$x := y+x$
$\ell_2 : x = 6 \wedge y = 3$

- (2)

$\ell_1 : x = 1 \wedge y = 3 \wedge x+y = 12$
$x := y+x$
$\ell_2 : x = 567 \wedge y = 34$

- (4)

$\ell_1 : x = 1 \wedge y = 3$
$z := x; x := y; y := z;$
$\ell_2 : x = 3 \wedge y = 1$

## ✎ Exercise 4

Show that each annotation is correct or incorrect, according to the verification conditions stated as follows:
$\forall x, y, x', y'.P_\ell(x,y) \wedge cond_{\ell,\ell'}(x,y) \wedge (x',y') = f_{\ell,\ell'}(x,y) \Rightarrow P_{\ell'}(x',y')$

- $\ell_1 : x = 3 \wedge y = z+x \wedge z = 2 \cdot x$
  $y := z+x$
  $\ell_2 : x = 3 \wedge y = x+6$
- $\ell_1 : x = 2^4 \wedge y = 2^{345} \wedge x \cdot y = 2^{350}$
  $x := y+x+2^x$
  $\ell_2 : x = 2^{56} \wedge y = 2^{345}$
- $\ell_1 : x = 1 \wedge y = 12$
  $x := 2 \cdot y+x$
  $\ell_2 : x = 1 \wedge y = 25$

## ✎ Exercise 5 `appex3_4.tla`

Consider the following small annotated algorithm:

$l0 : \{v = 3\}$
$v := v+2$
$l1 : \{v = 5\}$

Write a TLA$^+$ module that makes explicit the transition relation, the initial conditions, the invariant and the safety property for partial correctness.

## ✎ Exercise 6

Define the verification conditions of partial correctness for the following structures. Define a TLA$^+$ model to check that the annotation is correct.

**Question 6.1**

$$\begin{array}{l} \ell 1 : \{P_{\ell 1}(x,y)\} \\ \quad x := x+y+7; \\ \ell 2 : \{P_{\ell 2}(x,y)\} \end{array}$$

**Question 6.2**

$$\begin{array}{l} \ell : \{P_{\ell}(x,y)\} \\ \quad x,y := y,x; \\ \ell' : \{P_{\ell'}(x,y)\} \end{array}$$

## ✎ Exercise 7

Determine the verification conditions for the bounded loop structure. We assume that $S$ does not modify $i$.

$$\begin{array}{l} \ell_1 : \{P_{\ell_1}(x)\} \\ \quad \textbf{FOR } i := 1 \textbf{ TO } n \textbf{ DO} \\ \quad\quad \ell_2 : \{P_{\ell_2}(i,x)\} \\ \quad\quad\quad S(x); \\ \quad\quad \ell_3 : \{P_{\ell_3}(i,x)\} \\ \quad \textbf{ENDFOR} \\ \ell_4 : \{P_{\ell_4}(x)\} \end{array}$$

## ✎ Exercise 8

`appex3_77.tla`

**Variables:** X,Y,Z
**Requires :** $x_0, y_0 \in \mathbb{N} \wedge z_0 \in \mathbb{Z}$
**Ensures :** $z_f = max(x_0, y_0)$

$\ell_0 : \{\ldots\}$
**if** $X < Y$ **then**
  $\ell_1 : \{\ldots\}$
  $Z := Y;$
  $\ell_2 : \{\ldots\}$
**else**
  $\ell_3 : \{\ldots\}$
  $Z := X;$
  $\ell_4 : \{\ldots\}$
;
$\ell_5 : \{\ldots\}$

**Algorithm 1:** Maximum of two numbers, not annotated

**Question 8.1** Complete algorithm 1 by annotating it.

**Question 8.2** Check that the annotation is correct, by applying the rules of partial correctness.

**Question 8.3** Check that the annotation is correct, by translating the annotated algorithm into a TLA module.

**Question 8.4** State and check partial correctness.

## ✎ Exercise 9 `appex3_8.tla`

The aim is to study and annotate the proposed program in order to obtain its partial correctness (that is, without proving termination). A state is a set of precise values of the variables of the program (specified by a predicate); we consider a label ($\ell$) between each pair of statements of the program. An annotation is the predicate describing the possible values of the variables in a state of the program. This annotation is written $P_\ell(v)$ and expresses the property satisfied by the variable $v$ at $\ell$.

**Question 9.1** You are asked:

1. to annotate all the labels of the program;
2. to propose a TLA$^+$ model to check the annotations and partial correctness.

**Variables:** X
**Requires :** $x_0 \in \mathbb{N}$
**Ensures :** $x_f = 0$

$\ell_0 : \{\ldots\}$
**while** $0 < X$ **do**
  $\ell_1 : \{\ldots\}$
  $X := X - 1;$
  $\ell_2 : \{\ldots\}$
;
$\ell_3 : \{\ldots\}$

**Algorithm 2:** Example, not annotated

**Question 9.2** Check the conditions showing that the annotation is valid and that it establishes partial correctness.

**Question 9.3** Translate the annotated algorithm into a TLA module containing the definition of the algorithm as `Next` and `Init`, the definition of the invariant given by the annotations, and the definitions of partial correctness and of the absence of run-time errors.

## ✎ Exercise 10

**Question 10.1** Let $t$ be an array (with values in $\mathbb{N}$). Give a predicate $max(m,t,a,b) = \ldots$ expressing that a number $m \in \mathbb{N}$ is the maximum of the array $t$ on the interval $a\,..\,b$.

**Question 10.2** In the same way, give a predicate $sorted(t,a,b)$ specifying that the array $t$ is sorted on the interval $a\,..\,b$.

## ✎ Exercise 11 `appex3_10.tla`

Algorithm 3 computes the maximum of a sequence of integer values. You are asked:

- to define the precondition and the postcondition;
- to annotate this algorithm;
- to check the verification conditions for partial correctness;
- to check the conditions for the absence of run-time errors.

**Variables:** F,N,M,I

**Requires :** $\left(\begin{array}{l} n_0 \in \mathbb{N} \wedge \\ n_0 \neq 0 \wedge \\ f_0 \in 0\,..\,n_0{-}1 \to \mathbb{N} \end{array}\right)$

**Ensures :** $\left(\begin{array}{l} m_f \in \mathbb{N} \wedge \\ m_f \in ran(f_0) \wedge \\ (\forall j \cdot j \in 0\,..\,n_0{-}1 \Rightarrow f_0(j) \leq m_f) \end{array}\right)$

$M := F(0);$
$I := 1;$
**while** $I < N$ **do**
  **if** $F(I) > M$ **then**
    $M := F(I);$
  ;
  $I{+}{+};$
;

**Algorithm 3:** Maximum of a list, not annotated

## ✎ Exercise 12 `appex3_12.tla`

We consider the algorithm *squareroot* (algorithm 4), which computes the integer square root of a natural number $x \in \mathbb{N}$.

**Question 12.1** Complete this algorithm by proposing three assertions:

- $P_{\ell_2}(z, y1, y2, y3)$
- $P_{\ell_4}(z, y1, y2, y3)$
- $P_{\ell_5}(z, y1, y2, y3)$

**Question 12.2** For each pair $(\ell, \ell')$ of labels corresponding to an elementary step, we check the following property: $\forall y_1, y_2, y_3, z, y'_1, y'_2, y'_3, z' . P_\ell(y_1, y_2, y_3, z) \wedge cond_{\ell,\ell'}(y_1, y_2, y_3, z) \wedge (y'_1, y'_2, y'_3, z') = f_{\ell,\ell'}(y_1, y_2, y_3, z) \Rightarrow P_{\ell'}(y'_1, y'_2, y'_3, z')$. State and check this property for the following pairs of labels: $(\ell_1, \ell_2)$; $(\ell_1, \ell_4)$; $(\ell_2, \ell_3)$; $(\ell_3, \ell_2)$; $(\ell_3, \ell_4)$; $(\ell_4, \ell_5)$.

**Question 12.3** We assume that all the verification conditions associated with the pairs of successive labels of the algorithm hold. Which two conditions must be shown to deduce that the algorithm is partially correct with respect to the precondition and the postcondition? Give these conditions explicitly and explain why they hold.

**Question 12.4** Explain why this algorithm has no run-time errors, provided that the initial data belong to a domain, to be defined, included in the domain of computer integers, that is, the integers that can be coded on $n$ bits. The set of computer integers on $n$ bits is written $\mathbb{Z}_n$ and defined by $\{i \mid i \in \mathbb{Z} \wedge -2^{n-1} \leq i \wedge i \leq 2^{n-1}{-}1\}$.

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

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

**Algorithm 4:** *squareroot*, partly annotated

## ✎ Exercise 13

For each question, show that the annotation is correct or incorrect, according to the verification conditions stated as follows: $\forall v, v'.P_\ell(v) \wedge cond_{\ell,\ell'}(v) \wedge v' = f_{\ell,\ell'}(v) \Rightarrow P_{\ell'}(v')$. You must answer by stating and proving the *verification conditions*.

**Question 13.1**

$\ell_1 : x = 12 \wedge y = 2 \wedge z = 3 \cdot x$
$x := z + y$
$\ell_2 : x = 38 \wedge y = 2$

**Question 13.2**

$\ell_1 : x = 3 \wedge y = 9$
$x := 3 \cdot y$
$\ell_2 : x = 27 \wedge y = 9$

**Question 13.3** Let $p$ be a number that is not a multiple of 3, that is, different from 3, 6, 9, 12, ...

$\ell_1 : x = 3 + z \wedge y = 1 \wedge z = 3 \wedge x = y$
$x := p \cdot y$
$\ell_2 : x = z \wedge y = z \wedge z = 4 \cdot p$

**Question 13.4** Let $r$ be a cube, that is, a number of the form $r = q^3$.

$\ell_1 : x = r \wedge u = x^r \wedge z = 6 \wedge x = u$
$y := r \cdot r \cdot r$
$\ell_2 : x = z \wedge y = z \wedge z = 4 \cdot p$

## ✎ Exercise 14 `appex3_13.tla`

Consider the following annotated algorithm, with its precondition and postcondition. We assume that $x1$ and $x2$ are constants.

**Variables:** X1,X2,Y1,Y2,Y3,Z
**Requires :** $x1_0 \in \mathbb{N} \wedge x2_0 \in \mathbb{N} \wedge x1_0 \neq 0$
**Ensures :** $z_f = {x1_0}^{x2_0}$

$\ell_0 : \{x1_0 \in \mathbb{N} \wedge x2_0 \in \mathbb{N} \wedge x1_0 \neq 0 \wedge y1_0, y2_0, y3_0, z_0 \in \mathbb{Z} \wedge (x1, x2, y1, y2, y3, z) = (x1_0, x2_0, y1_0, y2_0, y3_0, z_0)\}$
$(y_1, y_2, y_3) := (x_1, x_2, 1);$
$\ell_1 : \{x1_0 \in \mathbb{N} \wedge x2_0 \in \mathbb{N} \wedge x1_0 \neq 0 \wedge y1_0, y2_0, y3_0, z_0 \in \mathbb{Z} \wedge (x1, x2, z) = (x1_0, x2_0, z_0) \wedge y_3 \cdot {y_1}^{y_2} = {x_1}^{x_2}\}$
**while** $y_2 \neq 0$ **do**
  $\ell_2 : \{x1_0 \in \mathbb{N} \wedge x2_0 \in \mathbb{N} \wedge x1_0 \neq 0 \wedge y1_0, y2_0, y3_0, z_0 \in \mathbb{Z} \wedge (x1, x2, z) = (x1_0, x2_0, z_0) \wedge y_3 \cdot {y_1}^{y_2} = {x_1}^{x_2} \wedge 0 < y2 \leq x2\}$
  **if** $odd(y_2)$ **then**
    $\ell_3 : \{\ldots\}$
    $y_2 := y_2 - 1;$
    $\ell_4 : \{x1_0 \in \mathbb{N} \wedge x2_0 \in \mathbb{N} \wedge x1_0 \neq 0 \wedge y1_0, y2_0, y3_0, z_0 \in \mathbb{Z} \wedge (x1, x2, z) = (x1_0, x2_0, z_0) \wedge y_3 \cdot y1 \cdot {y_1}^{y_2} = {x_1}^{x_2} \wedge 0 \leq y2 \leq x2 \wedge even(y2)\}$
    $y_3 := y_3 \cdot y_1;$
    $\ell_5 : \{x1_0 \in \mathbb{N} \wedge x2_0 \in \mathbb{N} \wedge x1_0 \neq 0 \wedge y1_0, y2_0, y3_0, z_0 \in \mathbb{Z} \wedge (x1, x2, z) = (x1_0, x2_0, z_0) \wedge y_3 \cdot {y_1}^{y_2} = {x_1}^{x_2} \wedge 0 \leq y2 \leq x2 \wedge even(y2)\}$
  ;
  $\ell_6 : \{\ldots\}$
  $y_1 := y_1 \cdot y_1;$
  $\ell_7 : \{x1_0 \in \mathbb{N} \wedge x2_0 \in \mathbb{N} \wedge x1_0 \neq 0 \wedge y1_0, y2_0, y3_0, z_0 \in \mathbb{Z} \wedge (x1, x2, z) = (x1_0, x2_0, z_0) \wedge y_3 \cdot {y_1}^{y_2\ div\ 2} = {x_1}^{x_2} \wedge 0 \leq y2 \leq x2 \wedge even(y2)\}$
  $y_2 := y_2\ div\ 2;$
  $\ell_8 : \{\ldots\}$
;
$\ell_9 : \{\ldots\}$
$z := y_3;$
$\ell_{10} : \{x1_0 \in \mathbb{N} \wedge x2_0 \in \mathbb{N} \wedge x1_0 \neq 0 \wedge y1_0, y2_0, y3_0, z_0 \in \mathbb{Z} \wedge (x1, x2) = (x1_0, x2_0) \wedge y_3 \cdot {y_1}^{y_2} = {x_1}^{x_2} \wedge y2 = 0 \wedge z = {x1}^{x2}\}$

**Algorithm 5:** Indian exponentiation, partly annotated

**Question 14.1** Complete the annotations associated with each label $\ell \in \{\ell_3, \ell_6, \ell_8, \ell_9\}$. You must write the complete annotation of each requested control point.

**Question 14.2** For each pair $(\ell, \ell')$ of labels corresponding to an elementary step, we check the following property: $\forall y_1, y_2, y_3, z, y'_1, y'_2, y'_3, z'. P_\ell(y_1, y_2, y_3, z) \wedge cond_{\ell,\ell'}(y_1, y_2, y_3, z) \wedge (y'_1, y'_2, y'_3, z') = f_{\ell,\ell'}(y_1, y_2, y_3, z) \Rightarrow P_{\ell'}(y'_1, y'_2, y'_3, z')$. State and check this property for the following pairs of labels: $(\ell_0, \ell_1)$; $(\ell_1, \ell_2)$; $(\ell_3, \ell_4)$; $(\ell_6, \ell_7)$; $(\ell_7, \ell_8)$; $(\ell_1, \ell_9)$; $(\ell_9, \ell_{10})$. This check also confirms the annotations completed in the previous question.

**Question 14.3** We assume that all the verification conditions associated with the pairs of successive labels of the algorithm hold. Which two conditions must be shown to deduce that the algorithm is partially correct with respect to the precondition and the postcondition? Give these conditions explicitly and explain why they hold.

**Question 14.4** In mathematics, the power $x_1^{x_2}$ is defined here for a non-zero value $x_1$, which is why the precondition requires $x_1$ to be different from 0. However, if a zero value of $x_1$ is used, the algorithm still runs and returns a value. One day, a mathematician applied this algorithm without checking whether $x_1$ was zero, and lost his temper! He accuses you of not having supplied the right algorithm for his requirements, and claims damages. Explain briefly why the text of the algorithm and its proof of correctness are enough to clear you, by making clear the role of the precondition and of the postcondition.