

<

Cours Modélisation et vérification des systèmes informatiques

Exercices

Utilisation d'un environnement de vérification Frama-c (III)

par Dominique Méry

4 décembre 2025

Exercice 1 Utiliser frama-c pour vérifier ou non les annotations suivantes :**Question 1.1**

$$\begin{aligned}\ell_1 : & x = 10 \wedge y = z+x \wedge z = 2 \cdot x \\ & y := z+x \\ \ell_2 : & x = 10 \wedge y = x+2 \cdot 10\end{aligned}$$
Question 1.2

$$\begin{aligned}\ell_1 : & x = 1 \wedge y = 12 \\ & x := 2 \cdot y \\ \ell_2 : & x = 1 \wedge y = 24\end{aligned}$$
Question 1.3

$$\begin{aligned}\ell_1 : & x = 11 \wedge y = 13 \\ & z := x; x := y; y := z; \\ \ell_2 : & x = 26/2 \wedge y = 33/3\end{aligned}$$
Question 1.4

$$\begin{aligned}\ell_1 : & x = 3 \wedge y = z+x \wedge z = 2 \cdot x \\ & y := z+x \\ \ell_2 : & x = 3 \wedge y = x+6\end{aligned}$$
Question 1.5

$$\begin{aligned}\ell_1 : & x = 2^4 \wedge y = 2^{345} \wedge x \cdot y = 2^{350} \\ & x := y+x+2^x \\ \ell_2 : & x = 2^{56} \wedge y = 2^{345}\end{aligned}$$
Question 1.6

$$\begin{aligned}\ell_1 : & x = 1 \wedge y = 12 \\ & x := 2 \cdot y+x \\ \ell_2 : & x = 1 \wedge y = 25\end{aligned}$$
Exercice 2 Traduire ce contrat dans le langage ACSL et vérifier le contrat.

```

variables x
requires
   $x_0 \in \mathbb{N}$ 
ensures
   $x_f \in \mathbb{N}$ 
begin
   $\ell_0 : \{x = x_0 \wedge x_0 \in \mathbb{N}\}$ 
  While ( $0 < x$ )
     $\ell_1 : \{0 < x \leq x_0 \wedge x_0 \in \mathbb{N}\}$ 
     $x := x - 1;$ 
     $\ell_2 : \{0 \leq x \leq x_0 \wedge x_0 \in \mathbb{N}\}$ 
    od;
     $\ell_4 : \{x = 0\}$ 
end

```

Exercice 3 Utiliser *frama-c* pour vérifier le contrat suivant :

Variables : F,N,M,I Requires : $\left(\begin{array}{l} n_0 \in \mathbb{N} \wedge \\ n_0 \neq 0 \wedge \\ f_0 \in 0..n_0-1 \rightarrow \mathbb{N} \end{array} \right)$ Ensures : $\left(\begin{array}{l} m_f \in \mathbb{N} \wedge \\ m_f \in \text{ran}(f_0) \wedge \\ (\forall j \cdot j \in 0..n_0-1 \Rightarrow f_0(j) \leq m_f) \end{array} \right)$ $M := F(0);$ $I := 1;$ while $I < N$ do if $F(i) > M$ then $M := F(I);$; $I++;$; b

Algorithme 1: Algorithme du maximum d'une liste non annotée

Exercice 4

Utiliser *frama-c* pour vérifier ke contrat suivant :

Soit l'algorithme annoté suivant se trouvant à la page suivante et les pré et postconditions définies pour cet algorithme comme suit : On suppose que $x1$ et $x2$ sont des constantes.

Exercice 5 Soit la fonction suivante utilisée dans un programme

Listing 1 – mainpower.c

```

#include <stdio.h>
#include <limits.h>
int power(int x)
{int r, cz, cv, cu, cw, ct, k;
cz=0;cv=0;cw=1;ct=3;cu=0;k=0;
while (k<x)
{

```

Variables : X1,X2,Y1,Y2,Y3,Z

Requires : $x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0$

Ensures : $z_f = x_{10}^{x_{20}}$

$\ell_0 : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2, y_1, y_2, y_3, z) = (x_{10}, x_{20}, y_{10}, y_{20}, y_{30}, z_0)\}$

$(y_1, y_2, y_3) := (x_1, x_2, 1);$

$\ell_1 : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2, z) = (x_{10}, x_{20}, z_0) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2}\}$

while $y_2 \neq 0$ **do**

$\ell_2 : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2, z) = (x_{10}, x_{20}, z_0) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge 0 < y_2 \leq x_2\}$

if $impair(y_2)$ **then**

$\ell_3 : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2, z) = (x_{10}, x_{20}, z_0) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge 0 < y_2 \leq x_2 \wedge impair(y_2)\}$

$y_2 := y_2 - 1;$

$\ell_4 : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2, z) = (x_{10}, x_{20}, z_0) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge 0 \leq y_2 \leq x_2 \wedge pair(y_2)\}$

$y_3 := y_3 \cdot y_1;$

$\ell_5 : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2, z) = (x_{10}, x_{20}, z_0) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge 0 \leq y_2 \leq x_2 \wedge pair(y_2)\}$

;

$\ell_6 : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2, z) = (x_{10}, x_{20}, z_0) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge 0 \leq y_2 \leq x_2 \wedge pair(y_2)\}$

$y_1 := y_1 \cdot y_1;$

$\ell_7 : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2, z) = (x_{10}, x_{20}, z_0) \wedge y_3 \cdot y_1^{y_2} \text{ div } 2 = x_1^{x_2} \wedge 0 \leq y_2 \leq x_2 \wedge pair(y_2)\}$

$y_2 := y_2 \text{ div } 2;$

$\ell_8 : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2, z) = (x_{10}, x_{20}, z_0) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge 0 \leq y_2 \leq x_2\}$

;

$\ell_9 : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2, z) = (x_{10}, x_{20}, z_0) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_2 = 0\}$

$z := y_3;$

$\ell_{10} : \{x_{10} \in \mathbb{N} \wedge x_{20} \in \mathbb{N} \wedge x_{10} \neq 0 \wedge y_{10}, y_{20}, y_{30}, z_0 \in \mathbb{Z} \wedge (x_1, x_2) = (x_{10}, x_{20}) \wedge y_3 \cdot y_1^{y_2} = x_1^{x_2} \wedge y_2 = 0 \wedge z = x_1^{x_2}\}$

Algorithm 2: Algorithme de l'exponentiation indienne annoté

```

printf("%d %d %d %d\n", cu, cv, cw, cz, ct);
cz=cz+cv+cw;
cv=cv+ct;
ct=ct+6;
cw=cw+3;
cu=cu+1;
k=k+1;)

r=cz;
return(r);
}

int p(int x)
{
    int r;
    if (x==0)
    {
        r=0;

    }
    else
    {
        r= p(x-1)+3*(x-1)*(x-1) + 3*(x-1)+1;

    }
    return(r);
}

int check(int n){
    int r1,r2,r;
    r1 = power(n);
    r2 = p(n);
    if (r1 != r2)
    {
        r = 0;
    }
    else
    {
        r = 1;
    };
    return r;
}

int main ()
{
    int counter;
    for( counter=0; counter<5; counter++ ) {
        int v,r;
        printf("Enter a natural number:");
        scanf("%d", &v);
        r = power(v);
        printf ("Power : %d ----> %d\n", v, r);

    };
}

```

Question 5.1 *Compiler ce programme et tester son exécution afin d'en dégager ses fonctionnalités.*

Question 5.2 *Annoter les fonctions principales.*

Question 5.3 *Vérifiez sa correction partielle et totale.*