Modélisation et vérification des systèmes informatiques

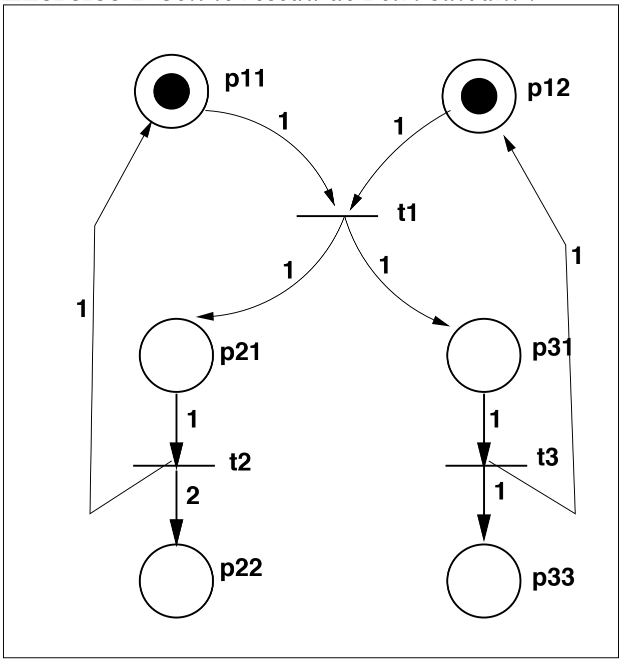
Dominique Méry

9 septembre 2015

Table des matières

1 Modélisation et vérification de systèmes

Exercice 1 *Soit le réseau de Petri suivant :*



Question 1.1 *Déterminer les conditions initiales.*

Question 1.2 *Déterminer les relations modélisant les transitions.*

Question 1.3 *Valider les propriétés et les hypothèses que vous pourrez faire sur ce réseau de Petri.*

precondition : $\left(\; a, b \in \mathbb{N} \wedge \; \right)$

postcondition : $\left(\begin{array}{l} z \in \mathbb{N} \wedge \\ z = gcd(a, b) \end{array} \right)$

local variables : $y1, y2 \in \mathbb{Z}$

$y1 := a$;
$y2 := b$;
while $y1 \neq y2$ **do**
  if *y1<y2* **then**
    $y2 := y2 - y1$;
  else
    $y1 := y1 - y2$;
  ;
;
$z := y1$;

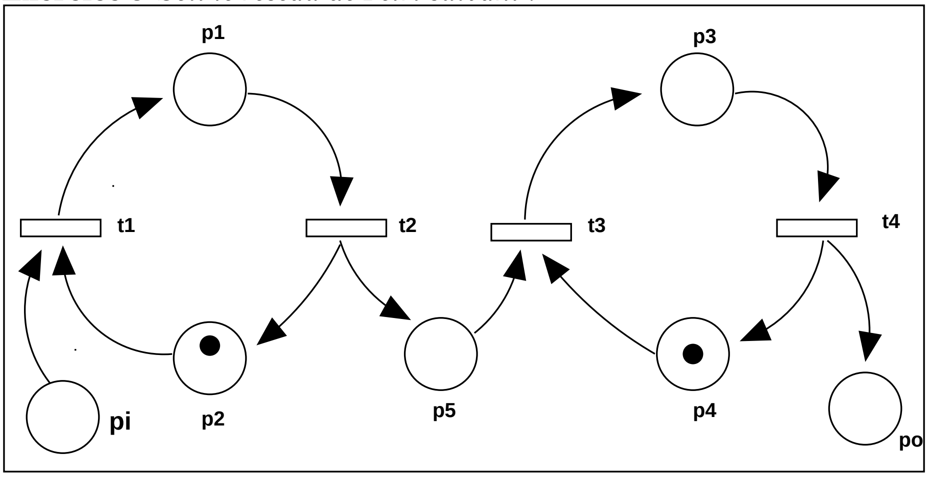
Algorithme 1: GCD

Exercice 2 Question 2.1 *Annoter l'algorithme 2 par des étiquettes correspondant à des points de contrôle.*

Question 2.2 *Construire un module TLA$^+$ modélisant les différents pas de calcul.*

Question 2.3 *Evaluer l'algorithme en posant des questions de sûreté.*

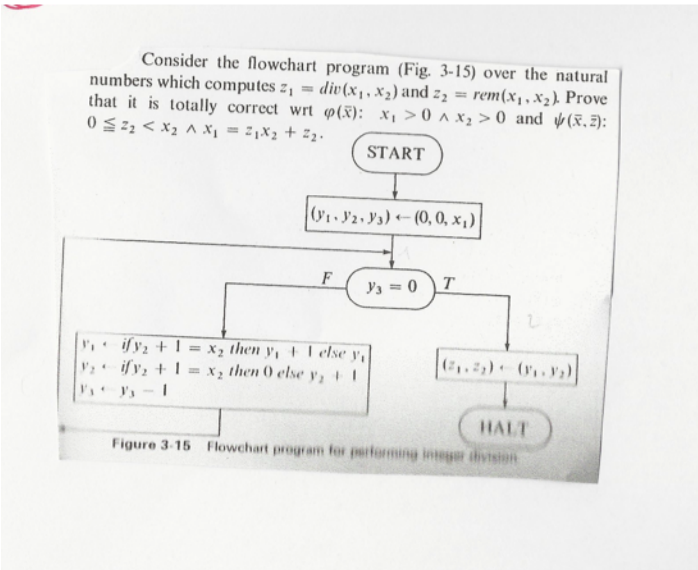
Exercice 3 *Soit le réseau de Petri suivant :*



Question 3.1 *Déterminer les conditions initiales.*

Question 3.2 *Déterminer les relations modélisant les transitions.*

Question 3.3 *Valider les propriétés et les hypothèses que vous pourrez faire sur ce réseau de Petri.*



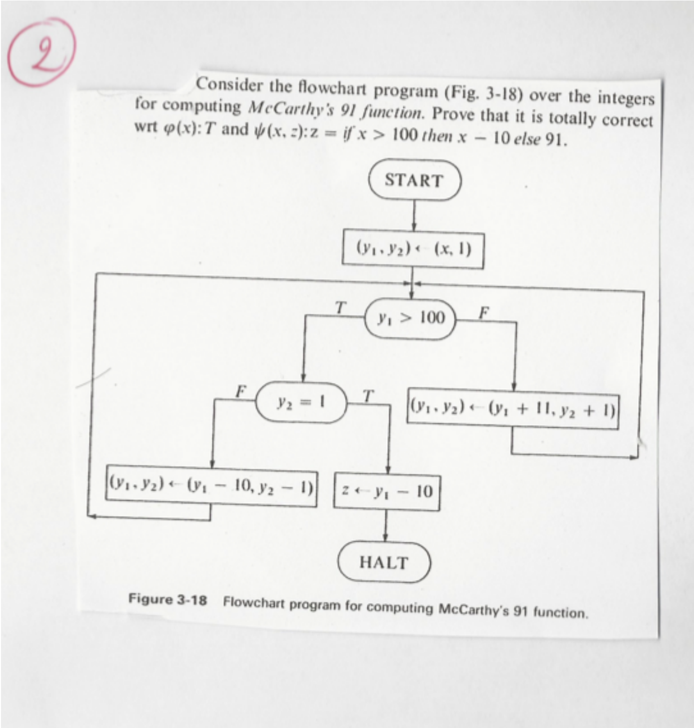
Consider the flowchart program (Fig. 3-15) over the natural numbers which computes $z_1 = div(x_1, x_2)$ and $z_2 = rem(x_1, x_2)$. Prove that it is totally correct wrt $\varphi(\bar{x})$: $x_1 > 0 \wedge x_2 > 0$ and $\psi(\bar{x}, \bar{z})$: $0 \leqq z_2 < x_2 \wedge x_1 = z_1 x_2 + z_2$.

Figure 3-15 Flowchart program for performing integer division

Exercice 4

Question 4.1 *Construire un module TLA$^+$ modélisant les différents pas de calcul.*

Question 4.2 *Evaluer l'algorithme en posant des questions de sûreté.*



2

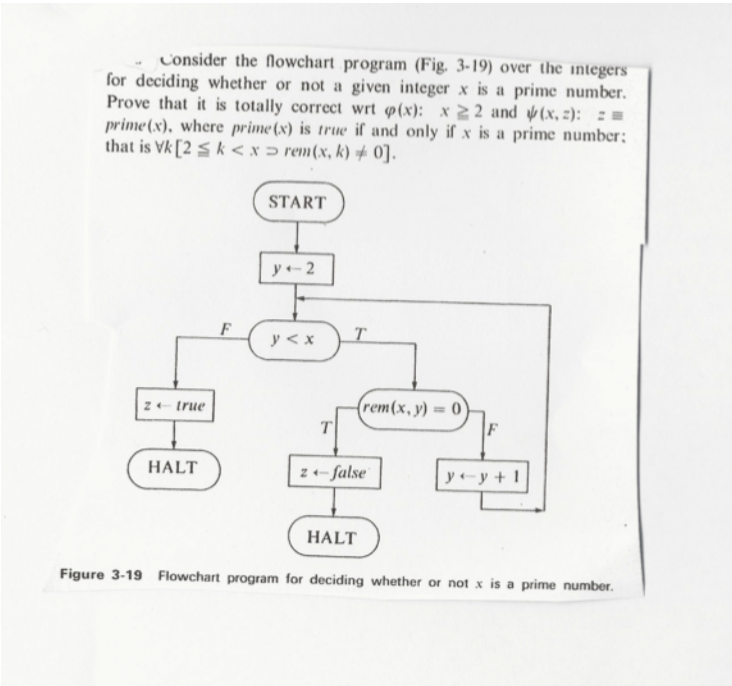
Consider the flowchart program (Fig. 3-18) over the integers for computing *McCarthy's 91 function*. Prove that it is totally correct wrt $\varphi(x)$: T and $\psi(x, z)$: $z = $ *if* $x > 100$ *then* $x - 10$ *else* 91.

Figure 3-18 Flowchart program for computing McCarthy's 91 function.

Exercice 5

Question 5.1 *Construire un module TLA$^+$ modélisant les différents pas de calcul.*

Question 5.2 *Evaluer l'algorithme en posant des questions de sûreté.*



Consider the flowchart program (Fig. 3-19) over the integers for deciding whether or not a given integer x is a prime number. Prove that it is totally correct wrt $\varphi(x)$: $x \geq 2$ and $\psi(x, z)$: $z \equiv prime(x)$, where $prime(x)$ is *true* if and only if x is a prime number; that is $\forall k[2 \leq k < x \supset rem(x, k) \neq 0]$.

Figure 3-19 Flowchart program for deciding whether or not x is a prime number.

Exercice 6

Question 6.1 *Annoter l'algorithme 2 par des étiquettes correspondant à des points de contrôle.*

Question 6.2 *Evaluer l'algorithme en posant des questions de sûreté.*

Exercice 7

$$\begin{array}{l} \ell_1 : x = 10 \ \wedge \ y = z{+}x \ \wedge z = 2{\cdot}x \\ y := z{+}x \\ \ell_2 : x = 10 \ \wedge \ y = x{+}2{\cdot}10 \end{array}$$

Ecrire un module TLA$^+$ pour vérifier l'annotation ci)dessus si elle est bien définie.

Exercice 8

$$\begin{array}{l} \ell_1 : x = 1 \ \wedge \ y = 12 \\ x := 2{\cdot}y \\ \ell_2 : x = 1 \ \wedge \ y = 24 \end{array}$$

Ecrire un module TLA$^+$ pour vérifier l'annotation ci)dessus si elle est bien définie.

Exercice 9

$$\begin{array}{l} \ell_1 : x = 11 \ \wedge \ y = 13 \\ z := x; x := y; y := z; \\ \ell_2 : x = 26/2 \ \wedge \ y = 33/3 \end{array}$$

Ecrire un module TLA$^+$ pour vérifier l'annotation ci)dessus si elle est bien définie.

Exercice 10 *Soit l'algorithme annoté 10 calculant la maximum d'un tableau f.*

Question 10.1 *Traduire cet algorithme sous forme d'un module TLA$^+$.*

Question 10.2 *Poser une question de sûreté pour la correction partielle.*

Question 10.3 *Poser une question de sûreté pour l'absence d'erreurs à l'exécution.*

Question 10.4 *Dégager un processus général de traduction d'un algorithme vers un module TLA$^+$.*

/* algorithme de calcul du maximum avec une boucle while de l'exercice **??** */

precondition : $\begin{pmatrix} n \in \mathbb{N}\wedge \\ n \neq 0\wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix}$

postcondition : $\begin{pmatrix} m \in \mathbb{N}\wedge \\ m \in ran(f)\wedge \\ (\forall j \cdot j \in 0 \mathinner{..} n{-}1 \Rightarrow f(j) \leq m) \end{pmatrix}$

local variables : $i \in \mathbb{Z}$

$\ell_0 : \{ \begin{pmatrix} n \in \mathbb{N}\wedge \\ n \neq 0\wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in \mathbb{Z} \wedge i \in \mathbb{Z} \wedge ...\}$

$m := f(0);$

$\ell_1 : \{ \begin{pmatrix} n \in \mathbb{N}\wedge \\ n \neq 0\wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in \mathbb{Z} \wedge m = f(0)\}$

$i := 1;$

$\ell_2 : \{ \begin{pmatrix} n \in \mathbb{N}\wedge \\ n \neq 0\wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i = 1 \wedge \begin{pmatrix} m \in \mathbb{N}\wedge \\ m \in ran(f[0..i{-}1])\wedge \\ (\forall j \cdot j \in 0 \mathinner{..} i{-}1 \Rightarrow f(j) \leq m) \end{pmatrix}\}$

while $i < n$ **do**

$\ell_3 : \{ \begin{pmatrix} n \in \mathbb{N}\wedge \\ n \neq 0\wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in 1..n{-}1 \wedge \begin{pmatrix} m \in \mathbb{N}\wedge \\ m \in ran(f[0..i{-}1])\wedge \\ (\forall j \cdot j \in 0 \mathinner{..} i{-}1 \Rightarrow f(j) \leq m) \end{pmatrix}\}$

if *f(i)>m* **then**

$\ell_4 : \{ \begin{pmatrix} n \in \mathbb{N}\wedge \\ n \neq 0\wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in 1..n{-}1 \wedge \begin{pmatrix} m \in \mathbb{N}\wedge \\ m \in ran(f[0..i{-}1])\wedge \\ (\forall j \cdot j \in 0 \mathinner{..} i{-}1 \Rightarrow f(j) \leq m) \end{pmatrix} \wedge$
$f(i) > m\}$

$m := f(i);$

$\ell_5 : \{ \begin{pmatrix} n \in \mathbb{N}\wedge \\ n \neq 0\wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in 1..n{-}1 \wedge \begin{pmatrix} m \in \mathbb{N}\wedge \\ m \in ran(f[0..i])\wedge \\ (\forall j \cdot j \in 0 \mathinner{..} i \Rightarrow f(j) \leq m) \end{pmatrix}\}$

;

$\ell_6 : \{ \begin{pmatrix} n \in \mathbb{N}\wedge \\ n \neq 0\wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in \mathbb{Z} \wedge \wedge i \in 1..n{-}1 \wedge \begin{pmatrix} m \in \mathbb{N}\wedge \\ m \in ran(f[0..i])\wedge \\ (\forall j \cdot j \in 0 \mathinner{..} i \Rightarrow f(j) \leq m) \end{pmatrix}\}$

$i{+}{+};$

$\ell_7 : \{ \begin{pmatrix} n \in \mathbb{N}\wedge \\ n \neq 0\wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i \in 1..n{-}1 \wedge \begin{pmatrix} m \in \mathbb{N}\wedge \\ m \in ran(f[0..i{-}1])\wedge \\ (\forall j \cdot j \in 0 \mathinner{..} i{-}1 \Rightarrow f(j) \leq m) \end{pmatrix}\}$

;

$\ell_8 : \{ \begin{pmatrix} n \in \mathbb{N}\wedge \\ n \neq 0\wedge \\ f \in 0 \mathinner{..} n{-}1 \rightarrow \mathbb{N} \end{pmatrix} \wedge i = n \wedge \begin{pmatrix} m \in \mathbb{N}\wedge \\ m \in ran(f)\wedge \\ (\forall j \cdot j \in 0 \mathinner{..} n{-}1 \Rightarrow f(j) \leq m) \end{pmatrix}\}$

Algorithme 2: $MAXIMUM$ annotée