Cours Modélisation et Vérification des Systèmes Informatiques (MVSI)

Modélisation, spécification et vérification CM1

Dominique Méry
Telecom Nancy, Université de Lorraine

Année universitaire 2024-2025

Plan

Sommaire

Current Summary

Tryptique $(\mathcal{D}, \mathcal{S}, \mathcal{R})$

- $\mathcal{R}$: exigences du système.

Tryptique $(\mathcal{D}, \mathcal{S}, \mathcal{R})$

- $\mathcal{R}$: exigences du système.
- $\mathcal{D}$: domaine du problème.

Tryptique $(\mathcal{D}, \mathcal{S}, \mathcal{R})$

- $\mathcal{R}$: exigences du système.
- $\mathcal{D}$: domaine du problème.
- $\mathcal{S}$: système répondant aux spécifications.

- $\mathcal{R}$: exigences du système.
- $\mathcal{D}$: domaine du problème.
- $\mathcal{S}$: système répondant aux spécifications.

$$\mathcal{D}, \mathcal{S} \quad \text{SATISFAIT} \quad \mathcal{R}$$

Tryptique $(\mathcal{D}, \mathcal{S}, \mathcal{R})$

- $\mathcal{R}$: exigences du système.
- $\mathcal{D}$: domaine du problème.
- $\mathcal{S}$: système répondant aux spécifications.

$$\mathcal{D}, \mathcal{S} \quad \text{SATISFAIT} \quad \mathcal{R}$$

- $\mathcal{R}$: pre/post.
- $\mathcal{D}$: entiers, réels, ...
- $\mathcal{S}$: code, procédure, programme, ...

A program P *satisfies* a (pre,post) contract :

- P transforms a variable v from initial values v_0 and produces a final value v_f : $v_0 \xrightarrow{P} v_f$
- v_0 satisfies pre : pre(v_0) and v_f satisfies post : post(v_0, v_f)
- pre$(v_0) \wedge v_0 \xrightarrow{P} v_f \Rightarrow$ post(v_0, v_f)
- $\mathbb{D}$ est le domaine RTE de V

```
requires pre(v0)
ensures post(v0, vf)
variables X
    begin
    0 : P0(v0, v)
    instruction0
    ...
    i : Pi(v0, v)
    ...
    instruction(f-1)
    f : Pf(v0, v)
    end
```

- $pre(v_0) \wedge v = v_0 \Rightarrow P_0(v_0, v)$
- $pre(v_0) \wedge P_f(v_0, v) \Rightarrow post(v_0, v)$
- For any pair of labelsℓ, ℓ' such that $\ell \longrightarrow \ell'$, one verifies that, pour any values $v, v' \in \textsc{Memory}$
$$\left(\begin{array}{l} \left(\begin{array}{l} pre(v_0) \wedge P_\ell(v_0, v)) \\ \wedge cond_{\ell,\ell'}(v) \wedge v' = f_{\ell,\ell'}(v) \end{array} \right) \\ \Rightarrow P_{\ell'}(v_0, v') \end{array} \right),$$
- For any pair of labels m, n such taht $m \longrightarrow n$, one verifies that, $\forall v, v' \in \textsc{Memory}$: $pre(v_0) \wedge P_m(v_0, v) \Rightarrow \textbf{DOM}(m, n)(v)$

Example of an annotation

VARIABLES X
REQUIRES $\ldots$
ENSURES $\ldots$
WHILE $0 < X$ **DO**
  $X := X - 1;$
 OD;

Example of an annotation

VARIABLES X
REQUIRES $\ldots$
ENSURES $\ldots$
WHILE $0 < X$ **DO**
  $X := X-1;$
 OD;

$\longrightarrow$

Example of an annotation

VARIABLES X
REQUIRES $\ldots$
ENSURES $\ldots$
WHILE $0 < X$ **DO**
  $X := X-1;$
 OD;

$\longrightarrow \longrightarrow$

Example of an annotation

VARIABLES X
REQUIRES $\ldots$
ENSURES $\ldots$
WHILE $0 < X$ **DO**
  $X := X - 1;$
 OD;

$\longrightarrow \longrightarrow \longrightarrow$

Example of an annotation

VARIABLES X
REQUIRES $\ldots$
ENSURES $\ldots$
WHILE $0 < X$ **DO**
  $X := X-1;$
 OD;

$\longrightarrow \longrightarrow \longrightarrow \longrightarrow$

Example of an annotation

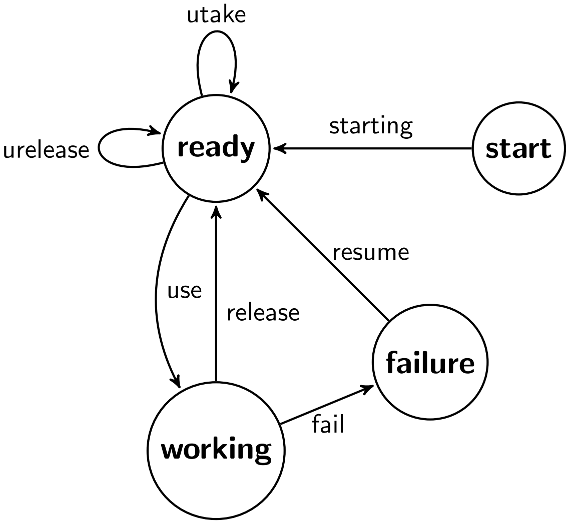
VARIABLES X
REQUIRES $\ldots$
ENSURES $\ldots$
WHILE $0 < X$ **DO**
  $X := X-1;$
 OD;

$\longrightarrow \longrightarrow \longrightarrow \longrightarrow$

CM3ONTRACT EX
VARIABLES $X(int)$
REQUIRES $x_0 \in \mathbb{N}$
ENSURES $x_f = 0$
  $\ell_0 : \{\ x = x_0 \wedge x_0 \in \mathbb{N}\}$
WHILE $0 < X$ **DO**
  $\ell_1 : \{0 < x \leq x_0 \wedge x_0 \in \mathbb{N}\}$
  $X := X-1;$
  $\ell_2 : \{0 \leq x \leq x_0 \wedge x_0 \in \mathbb{N}\}$
 OD;
  $\ell_3 : \{x = 0\}$

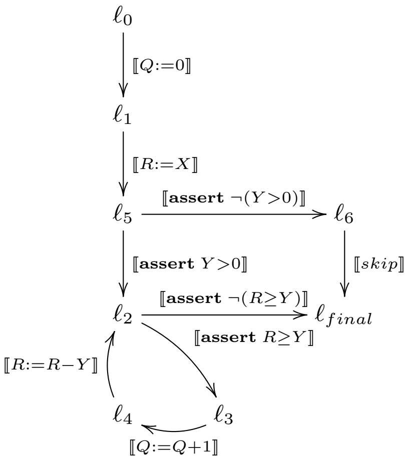
Current Summary

Un petit système en tant qu'automate



Programme en organigramme

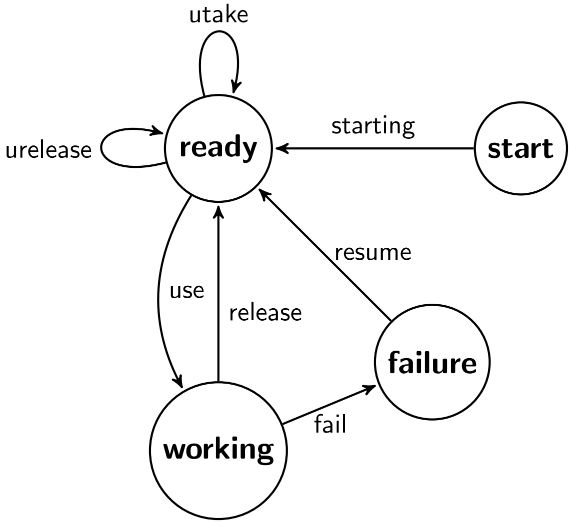
$\ell_0[Q := 0];$
$\ell_1[R := X];$
IF $\ell_5[Y > 0]$
WHILE $\ell_2[R \geq Y]$
$\ell_3[Q := Q{+}1];$
$\ell_4[R := R{-}Y]$
ENDWHILE
ELSE
$\ell_6[skip]$
ENDIF



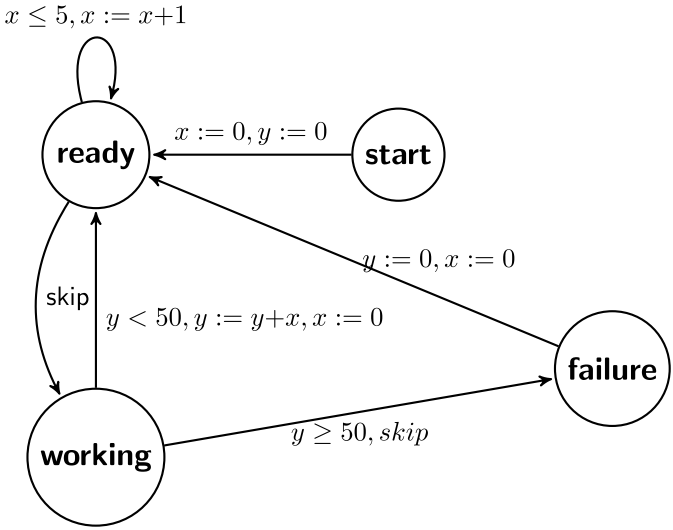
Observations

- Un automate a des états de contrôle : compteur ordinal d'un programme
- Un automate a des étiquettes : événements, actions, ...
- Un automate peut aussi avoir des variables explicites qui sont modifiées par des actions
- Un automate décrit des exécutions possibles qui sont des chemins suivant les informations de l'automate.

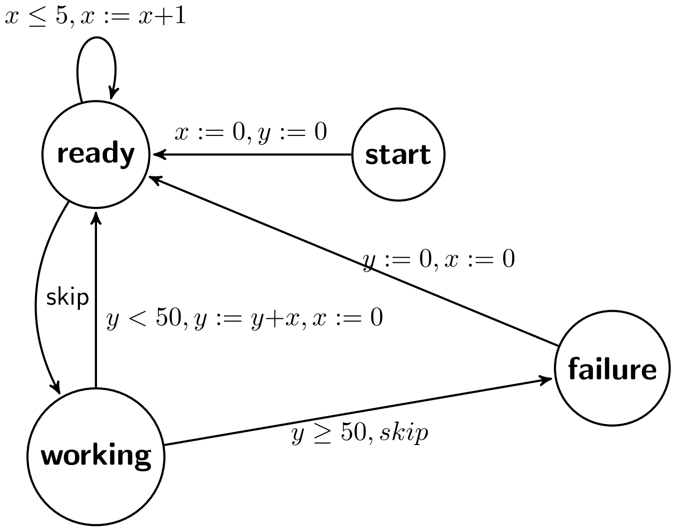
Un petit système en tant qu'automate



Un petit système en tant qu'automate

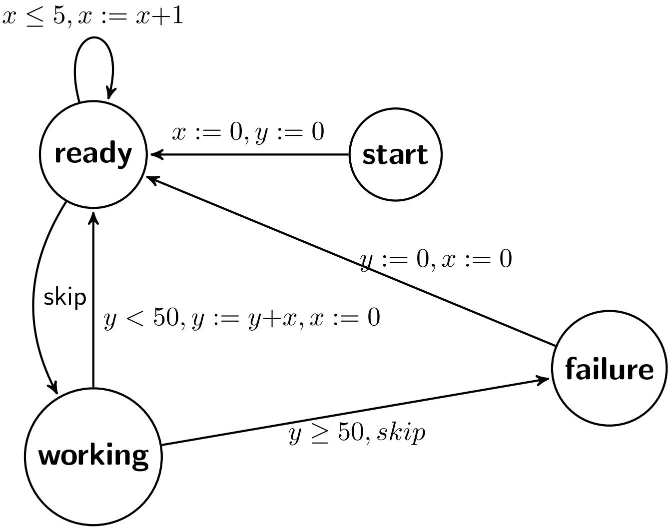


Un petit système en tant qu'automate



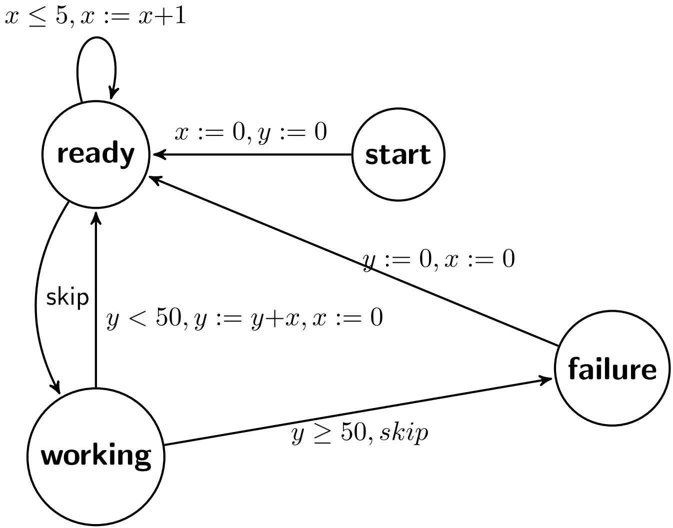
- safety1 : $0 \leq x \leq 5$

Un petit système en tant qu'automate



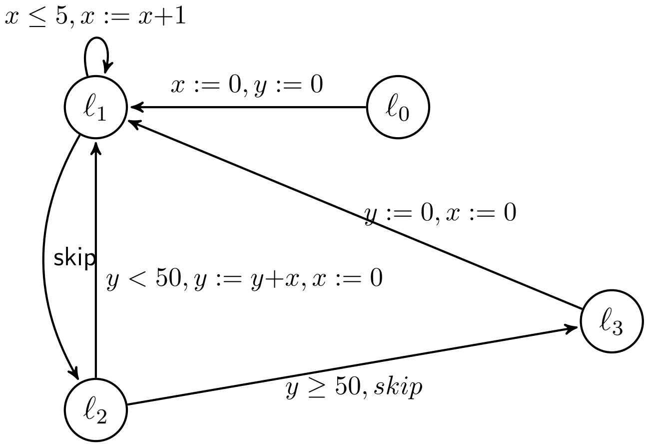
- safety1 : $0 \leq x \leq 5$ et ...

Un petit système en tant qu'automate

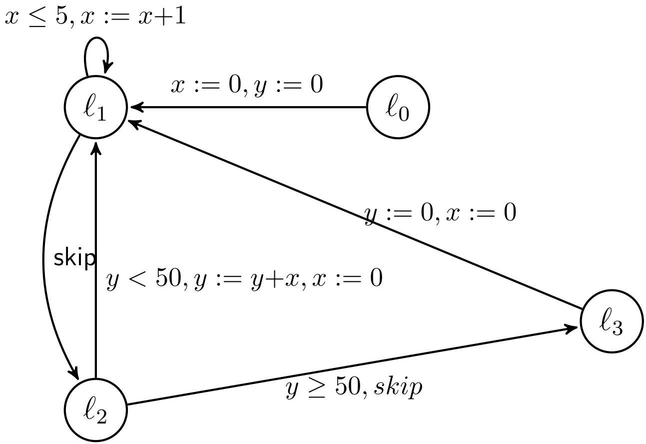


- safety1 : $0 \leq x \leq 5$ et ... safety2 : $0 \leq y \leq 56$

Un petit système en tant qu'automate

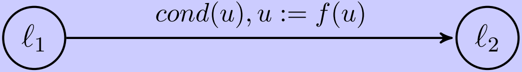


Un petit système en tant qu'automate



- safety1 : $0 \leq x \leq 5$ et safety2 : $0 \leq y \leq 56$
- $skip = x := x, y := y$
- skip $= TRUE, x := x, y := y = TRUE, skip$

Transition entre deux états de contrôle



Quelques formes de transitions

Transition entre deux états de contrôle

$$\ell_1 \xrightarrow{\;cond(u), u := f(u)\;} \ell_2$$

Transition entre deux états de contrôle

$$pc = \ell_1 \xrightarrow{\left(\begin{array}{l} cond(u) \\ \wedge\ pc = \ell_1 \end{array}\right), \begin{array}{l} u := f(u) \\ pc := \ell_2 \end{array},} pc = \ell_2$$

Quelques formes de transitions

Transition entre deux états de contrôle

$$\ell_1 \xrightarrow{cond(u), u := f(u)} \ell_2$$

Transition entre deux états de contrôle

$$pc = \ell_1 \xrightarrow{\left(\begin{array}{l} cond(u) \\ \wedge\ pc = \ell_1 \end{array}\right),\ \begin{array}{l} u := f(u) \\ pc := \ell_2 \end{array},} pc = \ell_2$$

Transition entre deux prédicats

$$P_1(u) \xrightarrow{\left(\begin{array}{l} cond(u) \\ \wedge\ pc = \ell_1 \end{array}\right),\ \begin{array}{l} u := f(u) \\ pc := \ell_2 \end{array},} P_2(u)$$

Current Summary

Modèle relationnel d'un système

Un modèle relationnel $\mathcal{MS}$ pour un système $\mathcal{S}$ est une structure

$$(Th(s,c), \mathsf{X}, \textsc{Vals}, \mathsf{Init}(x), \{r_0, \ldots, r_n\})$$

où

- $Th(s,c)$ est une théorie définissant les ensembles, les constantes et les propriétés statiques de ces éléments.
- X est une liste de variables flexibles.
- $\textsc{Vals}$ est un ensemble de valeurs possibles pour X.
- $\{r_0, \ldots, r_n\}$ est un ensemble fini de relations reliant les valeurs avant x et les valeurs après x'.
- $\mathsf{Init}(x)$ définit l'ensemble des valeurs initiales de X.
- la relation r_0 est la relation $Id[\textsc{Vals}]$, identité sur $\textsc{Vals}$.

$\boxtimes$ Definition

Soit $(Th(s,c), \mathsf{X}, \textsc{Vals}, \mathsf{Init}(x), \{r_0, \ldots, r_n\})$ un modèle relationnel d'un système $\mathcal{S}$. La relation Next associée à ce modèle est définie par la disjonction des relations r_i :

Next $\stackrel{def}{=} r_0 \vee \ldots \vee r_n$

pour une variable x, nous définissons les valeurs suivantes :

- x est la valeur courante de la variable X.
- x' est la valeur suivante de la variable X.
- x_0 ou $\underline{x}$ sont la valeur initiale de la variable X.
- $\overline{x}$ ou x_f est la valeur finale de la variable X, quand cette notion a du sens.

⊠ Definition(assertion)

Soit $(Th(s,c), \mathsf{X}, \textsc{Vals}, \mathsf{Init}(x), \{r_0, \ldots, r_n\})$ un modèle relationnel M d'un système $\mathcal{S}$. Une propriété A est une propriété assertionnelle de sûreté pour le système $\mathcal{S}$, si

$$\forall x_0, x \in \textsc{Vals}.Init(x_0) \wedge \mathsf{Next}^\star(x_0, x) \Rightarrow A(x).$$

⊠ Definition(relation)

Soit $(Th(s,c), \mathsf{X}, \textsc{Vals}, \mathsf{Init}(x), \{r_0, \ldots, r_n\})$ un modèle relationnel M d'un système $\mathcal{S}$. Une propriété R est une propriété relationnelle de sûreté pour le système $\mathcal{S}$, si

$$\forall x_0, x \in \textsc{Vals}.Init(x_0) \wedge \mathsf{Next}^\star(x_0, x) \Rightarrow R(x_0, x).$$

Assertion versus relation

- P. et R. Cousot développent une étude complète des propriétés d'invariance et de sûreté en mettant en évidence correspondances entre les différentes méthodes ou systèmes proposées par Turing, Floyd, Hoare, Wegbreit, Manna ... et reformulent les principes d'induction utilisés pour définir ces méthodes de preuve (voir les deux cubes des 16 principes).