



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

- 1 Prolégomènes
- 2 Modélisation de programmes et de systèmes logiciels
- 3 Modélisation relationnelle



Sommaire

1 Prolégomènes

2 Modélisation de programmes et de systèmes logiciels

3 Modélisation relationnelle



Current Summary

1 Prolégomènes

2 Modélisation de programmes et de systèmes logiciels

3 Modélisation relationnelle

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

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

 $ightharpoonup \mathcal{D}$: domaine du problème.

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

 $ightharpoonup \mathcal{D}$: domaine du problème.

 $ightharpoonup \mathcal{S}$: système répondant aux spécifications.

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

 $ightharpoonup \mathcal{D}$: domaine du problème.

 $ightharpoonup \mathcal{S}$: système répondant aux spécifications.

 \mathcal{D}, \mathcal{S} satisfait \mathcal{R}

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

\mathcal{D}, \mathcal{S} SATISFAIT \mathcal{R}

- $ightharpoonup \mathcal{R}$: pre/post.
- $ightharpoonup \mathcal{D}$: entiers, réels, . . .
- $ightharpoonup \mathcal{S}$: 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 $v_f: v_0 \xrightarrow{P} v_f$
- ightharpoonup v $_0$ satisfies pre : $\mathsf{pre}(v_0)$ and v $_f$ satisfies post : $\mathsf{post}(v_0,v_f)$
- ▶ D est le domaine RTE de V

```
requires pre(v_0)
ensures post(v_0, v_f)
variables X
            \begin{aligned} & \text{begin} \\ & 0: P_0(v_0, v) \\ & \text{instruction}_0 \end{aligned}
           i: P_i(v_0, v)
...
             instruction_{f-1}
             f: P_f(v_0, v)
```

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

```
 \begin{array}{l} \textbf{VARIABLES} \ X \\ \textbf{REQUIRES} \ \dots \\ \textbf{ENSURES} \ \dots \\ \textbf{WHILE} \ 0 < X \ \textbf{DO} \\ X := X - 1; \\ \textbf{OD}; \end{array}
```

```
 \begin{array}{c} \textbf{VARIABLES} \ X \\ \textbf{REQUIRES} \ \dots \\ \textbf{ENSURES} \ \dots \\ \textbf{WHILE} \ 0 < X \ \textbf{DO} \\ X := X - 1; \\ \textbf{OD}; \end{array}
```

 $\begin{array}{c} \textbf{VARIABLES} \ X \\ \textbf{REQUIRES} \ \dots \\ \textbf{ENSURES} \ \dots \\ \textbf{WHILE} \ 0 < X \ \textbf{DO} \\ X := X - 1; \\ \textbf{OD}; \end{array} \longrightarrow \longrightarrow$

 $\begin{array}{c} \textbf{VARIABLES} \ X \\ \textbf{REQUIRES} \ \dots \\ \textbf{ENSURES} \ \dots \\ \textbf{WHILE} \ 0 < X \ \textbf{DO} \\ X := X - 1; \\ \textbf{OD}; \end{array} \longrightarrow \longrightarrow \longrightarrow$

 $\begin{array}{c} \textbf{VARIABLES} \ X \\ \textbf{REQUIRES} \ \dots \\ \textbf{ENSURES} \ \dots \\ \textbf{WHILE} \ 0 < X \ \textbf{DO} \\ X := X - 1; \\ \textbf{OD}; \end{array} \longrightarrow \longrightarrow \longrightarrow$

```
VARIABLES X
REQUIRES ...
ENSURES ...
WHILE 0 < X DO X := X-1;
OD;
```

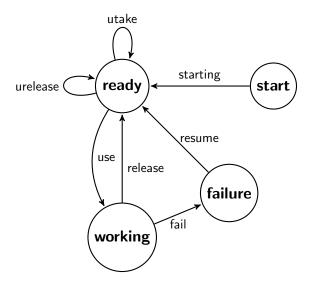
```
CM3ONTRACT EX
VARIABLES X(int)
REQUIRES x_0 \in \mathbb{N}
ENSURES x_f = 0
  \ell_0: \{ \ x = x_0 \land x_0 \in \mathbb{N} \}
WHILE 0 < X DO
  \ell_1 : \{0 < x \le x_0 \land x_0 \in \mathbb{N}\}
  X := X - 1;
  \ell_2 : \{0 \le x \le x_0 \land x_0 \in \mathbb{N}\}
 OD:
  \ell_3: \{x=0\}
```

Current Summary

1 Prolégomènes

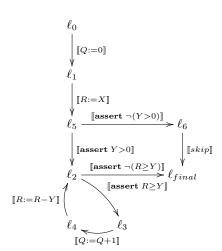
2 Modélisation de programmes et de systèmes logiciels

3 Modélisation relationnelle



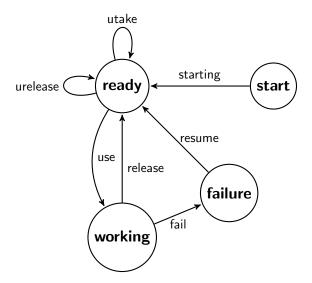
Programme en organigramme

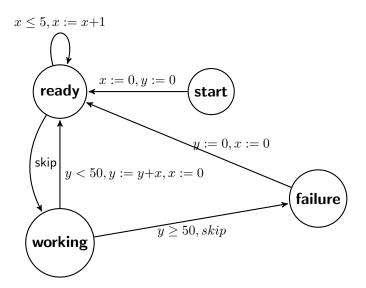
```
\begin{array}{l} \ell_0[Q:=0];\\ \ell_1[R:=X];\\ \textbf{IF}\ \ell_5[Y>0]\\ \textbf{WHILE}\ \ell_2[R\geq Y]\\ \ell_3[Q:=Q+1];\\ \ell_4[R:=R-Y]\\ \textbf{ENDWHILE}\\ \textbf{ELSE}\\ \ell_6[skip]\\ \textbf{ENDIF} \end{array}
```

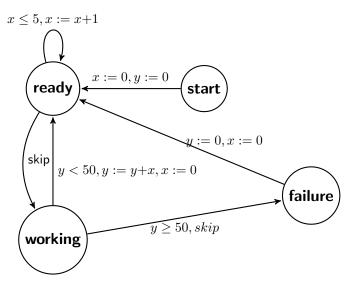


Observations

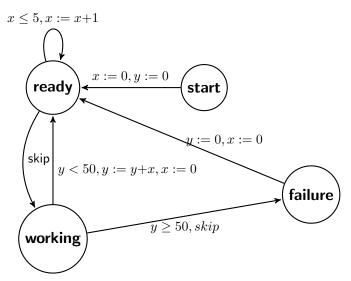
- Un automate a des états de contrôle : compteur ordinal d'un programme
- ▶ Un automate a des étiquettes : événements, actions, . . .
- Un automate peut aussi avoir des variables explicites qui sont modifiées par des actions
- Un automate décrit des exécutions possibles qui sont des chemins suivant les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



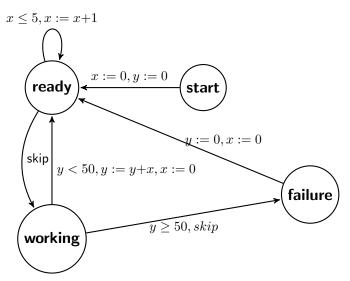




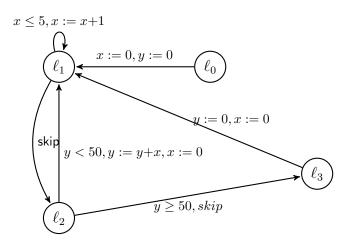
ightharpoonup safety1 : $0 \le x \le 5$

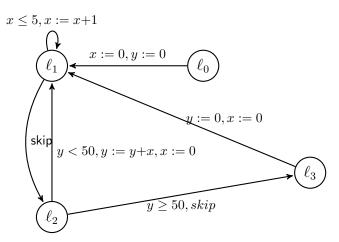


ightharpoonup safety1 : $0 \le x \le 5$ et . . .



▶ safety1 : $0 \le x \le 5$ et . . . safety2 : $0 \le y \le 56$

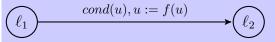




- ▶ safety1 : $0 \le x \le 5$ et safety2 : $0 \le y \le 56$
- ightharpoonup skip = x := x, y := y
- ightharpoonup skip = TRUE, x := x, y := y = TRUE, skip

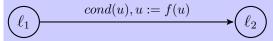
Quelques formes de transitions

Transition entre deux états de contrôle

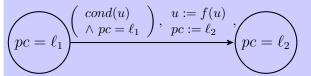


Quelques formes de transitions

Transition entre deux états de contrôle

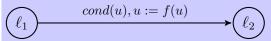


Transition entre deux états de contrôle



Quelques formes de transitions

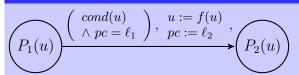
Transition entre deux états de contrôle



Transition entre deux états de contrôle

$$(pc = \ell_1) \xrightarrow{\begin{pmatrix} cond(u) \\ \land pc = \ell_1 \end{pmatrix}}, \begin{array}{c} u := f(u) \\ pc := \ell_2 \end{array}, \\ pc = \ell_2)$$

Transition entre deux prédicats



Current Summary

1 Prolégomènes

2 Modélisation de programmes et de systèmes logiciels

3 Modélisation relationnelle

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

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

οù

- ightharpoonup 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.
- ightharpoonup VALS est un ensemble de valeurs possibles pour X.
- $rackleft \{r_0, \dots, r_n\}$ est un ensemble fini de relations reliant les valeurs avant x et les valeurs après x'.
- ▶ Init(x) définit l'ensemble des valeurs initiales de X.
- ▶ la relation r_0 est la relation Id[Vals], identité sur Vals.

.....

□ Definition

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

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

The second secon

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.
- $ightharpoonup x_0$ 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.

Propriétés de sûreté et d'invariance dans un modèle relationnel

......

□ Definition(assertion)

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

$$\forall x_0, x \in \text{VALS}.Init(x_0) \land \mathsf{Next}^{\star}(x_0, x) \Rightarrow A(x).$$

.....

□ Definition(relation)

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

$$\forall x_0, x \in \text{VALS}.Init(x_0) \land \text{Next}^*(x_0, x) \Rightarrow R(x_0, x).$$

.....

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).