Spécifications Formelles

Pierre Roux (transparents de Xavier Thirioux)

2023-2024

Spécification Formelle

$$
\begin{tikzcd}
\text{Spécification} \arrow[r, bend left] & R_0 \arrow[l, bend left] \arrow[r, bend left] & \cdots \arrow[l, bend left] \arrow[r, bend left] & R_n \arrow[l, bend left] \arrow[r, bend left] & \text{Programme} \arrow[l, bend left]
\end{tikzcd}
$$

Abstraction et Raffinement

- Notions duales.
- Définie % un ensemble de propriétés E.
- Exprime une forme de conservation de ces propriétés.

Note : un système $\mathcal{S}'$ raffine un système $\mathcal{S}$ ssi $\mathcal{S}'$ conserve toutes les propriétés de E attachées à $\mathcal{S}$.

Différents avatars

On retrouve le raffinement sous différentes formes :

- raffinement (raffinage) de programmes séquentiels.
- raffinement par la relation d'héritage dans les langages objets.
- raffinement d'une spécification par une implantation : exemple des annotations JML JAVA qui constituent une spécification.
- raffinement de données : la structure de liste est un raffinement (implantation) de la notion d'ensemble.

Applications

Il existe des méthodes/outils industriels :

- La méthode Z.
- La méthode et les outils B (utilisés pour la ligne 14 météor).
- Outil *Zing* (laboratoires Microsoft) : correction d'implantations de protocoles de communication.

Plan du cours

1. Définition du cadre
2. Simulation/bisimulation faible/forte
3. Calcul de processus CCS
4. Raffinement de programmes

Première partie

Définition du cadre

Plan

Introduction

Le raffinement consiste à comparer 2 systèmes.

- On compare des langages/comportements.
- Par rapport à des événements observables.
- Les systèmes sont des systèmes de transitions étiquetés.

Note : les systèmes sont des boîtes noires, dont le fonctionnement n'est pas visible *a priori*.

Systèmes de transitions étiquetés

Un système de transitions étiqueté (S.T.E.) est un quadruplet $\langle S, L, I, R \rangle$.

- S est un ensemble d'états. Peut être fini ou infini.
- L est un alphabet (ensemble des étiquettes). Peut être fini ou infini.
- $I \subseteq S$ est l'ensemble des états initiaux.
- $R \subseteq S \times L \times S$ est une relation (de transitions) entre paires d'états. $(s, l, s') \in R$ signifie qu'il existe une transition faisant passer le système de l'état s à l'état s' par le biais d'un événement l.

Note : $(s, l, s') \in R$ se représente par : $s \xrightarrow{l} s'$.

Traces

Soit $\langle S, L, I, R \rangle$ un S.T.E.

On appelle trace finie maximale partant de s_0 un $\sigma \in L^\star$ tel que :

- $\sigma = l_0.l_1 \ldots l_{n-1}$
- Il existe une famille $\{s_i\}_{i \in [0..n]}$ tq :
  - $\forall i \in [0..n[\colon (s_i, l_i, s_{i+1}) \in R$
  - s_n est un état d'interblocage

On appelle trace infinie partant de s_0 un $\sigma \in L^\omega$ tel que :

- $\sigma = l_0.l_1.l_2 \ldots$
- Il existe une famille $\{s_i\}_{i \in \mathbb{N}}$ tq :
  - $\forall i \in \mathbb{N}.(s_i, l_i, s_{i+1}) \in R$

Les préfixes de trace, pour $\sigma \in L^\star \cup L^\omega$, sont définis comme suit :

$$Prefix(\sigma) \triangleq \{l_0. \ldots l_{n-1} \in L^\star \mid \sigma = l_0. \ldots l_{n-1} \ldots\}$$

Note : On ne s'intéresse qu'aux étiquettes, les états sont supposés inobservables.

Evénements observables

On considère souvent deux classes d'événements.

- Les événements dits observables représentés par une étiquette ordinaire.

→ entrée/sortie, modification de variable globale, etc

- Les événements internes non observables, représentés par l'étiquette spéciale τ. Cette étiquette permet donc de savoir que le système a fait quelque chose, non visible pour l'utilisateur.

→ communication interne, modification de variable locale, etc

Note : $\tau \simeq \epsilon$, la transition spontanée des automates.

Relations utiles

Définition 1 (Relations dérivées de $\rightarrow$)

- On définit $s \rightarrow^\star s'$ comme la fermeture réflexive et transitive de $\rightarrow$.
- On définit $\Rightarrow$ comme la fermeture réflexive et transitive de $\rightarrow$ restreinte à τ, i.e. $s \Rightarrow s' \triangleq s \xrightarrow{\tau} s_1 \ldots \xrightarrow{\tau} s'$.
- On définit : $s \overset{a}{\Rightarrow} s' \triangleq s \Rightarrow s_1 \xrightarrow{a} s_2 \Rightarrow s'$.

Plan

Notion de Simulation

- Relation fondamentale entre systèmes.
- Plus forte que l'inclusion des langages.
- Peut s'appliquer au raffinement de programmes.
- Peut s'étendre à l'équivalence de systèmes.
- Différentes versions, selon le traitement de τ.

Simulation forte

Soient $\mathcal{S} = \langle S, L, I, R \rangle$ et $\mathcal{S}' = \langle S', L, I', R' \rangle$ deux S.T.E. définis sur un même alphabet d'événements $\mathcal{L}$.

Définition 2 (Relation de simulation forte)

On dit qu'une relation $Simu \subseteq S \times S'$ est une relation de simulation forte de S par S' ssi :

$$\begin{array}{l} \forall s_1, s_2 \in S, l \in L, s_1' \in S'. \\ \quad \langle s_1, s_1' \rangle \in Simu \;\wedge\; s_1 \xrightarrow{l} s_2 (\in R) \\ \quad \Longrightarrow \\ \quad \exists s_2'. \langle s_2, s_2' \rangle \in Simu \;\wedge\; s_1' \xrightarrow{l} s_2' (\in R') \end{array}$$

Simulation forte

Graphiquement parlant :

$$
\begin{tikzcd}
s_1 \arrow[r, dash, "Simu"] \arrow[d, "l"] & s_1' \arrow[d, dashed, "l"] \\
s_2 \arrow[r, dash, dashed, "Simu"] & \exists s_2'
\end{tikzcd}
$$

La relation *Simu* permet de relier les états de $\mathcal{S}$ aux états de $\mathcal{S}'$ qui font la même chose (ou plus), i.e. qui traitent au moins autant d'événements, et de la même façon. Cette définition de la simulation marche également pour $\mathcal{S}'{=}\mathcal{S}$.

Simulation forte entre S.T.E.

Simulation entre états

On dit que $s' \in S'$ simule fortement $s \in S$ ssi il existe une relation de simulation forte $Simu \subseteq S \times S'$ vérifiant la définition 2 et telle que :

$$\langle s, s' \rangle \in Simu$$

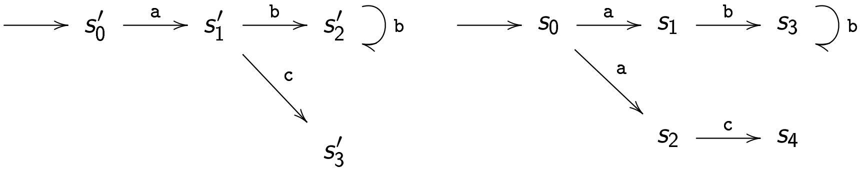
Simulation entre S.T.E.

On dit que $\mathcal{S}'$ simule fortement $\mathcal{S}$ ssi il existe une relation de simulation forte $Simu \subseteq S \times S'$ vérifiant la définition 2 et telle que :

$$\forall i \in I . \exists i' \in I' . \langle i, i' \rangle \in Simu$$

Note : $\mathcal{S}'$ simule $\mathcal{S} \equiv \mathcal{S}$ est une abstraction de $\mathcal{S}'$.

Exemple 1



Ici, $\mathcal{S}'$ simule $\mathcal{S}$, i.e. s'_0 simule s_0, car on peut vérifier que la relation R suivante :

$$R \triangleq \{\langle s_0, s'_0\rangle, \langle s_1, s'_1\rangle, \langle s_2, s'_1\rangle, \langle s_3, s'_2\rangle, \langle s_4, s'_3\rangle\}$$

- contient la paire $\langle s_0, s'_0\rangle$;
- est une relation de simulation suivant la définition 2.

Démonstration

$$
\begin{tikzcd}
s_0 \arrow[r, dash, "R"] \arrow[d, "\mathtt{a}"] & s'_0 \arrow[d, "\mathtt{a}"] & s_0 \arrow[r, dash, "R"] \arrow[d, "\mathtt{a}"] & s'_0 \arrow[d, "\mathtt{a}"] & s_1 \arrow[r, dash, "R"] \arrow[d, "\mathtt{b}"] & s'_1 \arrow[d, "\mathtt{b}"] \\
s_1 \arrow[r, dash, "R"] & \exists s'_1 & s_2 \arrow[r, dash, "R"] & \exists s'_1 & s_3 \arrow[r, dash, "R"] & \exists s'_2 \\
s_2 \arrow[r, dash, "R"] \arrow[d, "\mathtt{c}"] & s'_1 \arrow[d, "\mathtt{c}"] & s_3 \arrow[r, dash, "R"] \arrow[d, "\mathtt{b}"] & s'_2 \arrow[d, "\mathtt{b}"] & s_4 \arrow[r, dash, "R"] & s'_3 \\
s_4 \arrow[r, dash, "R"] & \exists s'_3 & s_3 \arrow[r, dash, "R"] & \exists s'_2 & &
\end{tikzcd}
$$

Attention : La relation R n'est pas la seule possible, ni la plus générale. On peut aussi ajouter la paire $\langle s_1, s'_2 \rangle$ entre autres.

Exemple 1

$$\begin{tikzcd}
{} \arrow[r] & s'_0 \arrow[r, "a"] & s'_1 \arrow[r, "b"] \arrow[dr, "c"] & s'_2 \arrow[loop right, "b"] \\
& & & s'_3
\end{tikzcd}
\qquad
\begin{tikzcd}
{} \arrow[r] & s_0 \arrow[r, "a"] \arrow[dr, "a"] & s_1 \arrow[r, "b"] & s_3 \arrow[loop right, "b"] \\
& & s_2 \arrow[r, "c"] & s_4
\end{tikzcd}$$

- **Par contre :** on ne peut pas construire une simulation sur $S' \times S$ qui contiendrait $\langle s'_0, s_0 \rangle$.

  Donc : $\mathcal{S}$ ne simule pas $\mathcal{S}'$.
- On peut néanmoins construire une relation qui prouverait que s_3 (ou même s_1) simule s'_2.

Structure algébrique

On considère des relations définies sur $S \times S$.

- vide : La relation $\emptyset$ est une simulation ;
- réflexivité : La relation identité Id est une simulation ;
- transitivité : Si R et R' sont deux simulations, alors $R; R' \triangleq \{\langle x, x'' \rangle \mid \exists x' \in S. \langle x, x' \rangle \in R \wedge \langle x', x'' \rangle \in R'\}$ est une simulation ;
- union : Si R et R' sont deux simulations, alors $R \cup R'$ est une simulation ;
- fermeture (étoile) de kleene : Si R est une simulation, alors $R^\star \triangleq \bigcup_{i \in \mathbb{N}} R^i$, avec $R^i \triangleq \underbrace{R; \ldots; R}_{i}$ est une simulation.

Structure algébrique

L'ensemble des simulations possède donc tous les opérateurs des langages réguliers :

- $\emptyset$ correspond au langage vide
- Id correspond à Λ
- $R; R'$ correspond à $e.e'$
- $R \cup R'$ correspond à $e + e'$
- $R^\star$ correspond à $e^\star$

$\rightarrow$ C'est une algèbre régulière.

- Reste à déterminer l'alphabet ?

Plus grande simulation forte

On peut définir la plus grande simulation de S par S', définis sur le même alphabet L. Cette relation existe car :

- L'union de 2 simulations est une simulation, plus grande ;
- Toute simulation est bornée par $S \times S'$.

De plus, si S et S' sont finis :

- La plus grande simulation est calculable ;
- Toute question "s' simule-t'il s ?" peut se résoudre en considérant la plus grande simulation.

Note : On représente cette relation par $\mathcal{S} \prec \mathcal{S}'$

Autres propriétés

Soient 2 S.T.E. $\mathcal{S}$ et $\mathcal{S}'$, définis sur le même alphabet L, tels que $\mathcal{S}'$ simule fortement $\mathcal{S}$.

Compositionalité

Tout système $\mathcal{G}$ contenant le sous-système $\mathcal{S}$, noté $\mathcal{G}[\mathcal{S}]$ est simulé par $\mathcal{G}[\mathcal{S}']$.

Autres propriétés

Soient 2 S.T.E. $\mathcal{S}$ et $\mathcal{S}'$, définis sur le même alphabet L, tels que $\mathcal{S}'$ simule fortement $\mathcal{S}$.

Conservation des traces

- $Prefix(Exec(\mathcal{S})) \subseteq Prefix(Exec(\mathcal{S}'))$.
- Pour tout s simulé par s',
  $Prefix(Traces(s)) \subseteq Prefix(Traces(s'))$.

Autres propriétés

Soient 2 S.T.E. $\mathcal{S}$ et $\mathcal{S}'$, définis sur le même alphabet L, tels que $\mathcal{S}'$ simule fortement $\mathcal{S}$.

Conservation des propriétés temporelles

Toute propriété temporelle LTL de sûreté (par ex. tout invariant) vérifiée par $\mathcal{S}'$ est conservée dans $\mathcal{S}$.

Autres propriétés

Soient $\mathcal{A} = \langle \mathcal{S}_\mathcal{A}, \mathcal{L}, \mathcal{I}_\mathcal{A}, \mathcal{R}_\mathcal{A}, \mathcal{F}_\mathcal{A} \rangle$ un automate et $\mathcal{D} = \langle \mathcal{S}_\mathcal{D}, \mathcal{L}, \mathcal{I}_\mathcal{D}, \mathcal{R}_\mathcal{D}, \mathcal{F}_\mathcal{D} \rangle$ le résultat de la déterminisation de $\mathcal{A}$.

Déterminisation des automates

- $\mathcal{D}$ simule fortement $\mathcal{A}$.
- $\exists$ une simulation R qui respecte le caractère terminal des états, i.e.

$$R \subseteq (F_\mathcal{A} \times F_\mathcal{D}) \cup ((S_\mathcal{A} \backslash F_\mathcal{A}) \times (S_\mathcal{D} \backslash F_\mathcal{D}))$$

Comment savoir si $\mathcal{S}$ est simulé par $\mathcal{S}'$?

Le problème général est de décider si un système ou un état donné en simule un autre.

- Pour les systèmes infinis, il faut :
  1. Exhiber une relation.
  2. Prouver qu'il s'agit d'une simulation.
- Pour les systèmes finis, on peut :
  1. Calculer la plus grande simulation possible de $\mathcal{S}$ par $\mathcal{S}'$.
  2. Vérifier qu'elle contient les paires d'états initiaux.

Calcul de la plus grande simulation forte

Soient 2 S.T.E. $\mathcal{S}$ et $\mathcal{S}'$, définis sur le même alphabet L.

Définition 3 (Plus grande simulation de $\mathcal{S}$ par $\mathcal{S}'$)

La plus grande simulation est la limite de la suite (décroissante) $(R_i)_{i\in\mathbb{N}}$:

$$\begin{cases} R_0 & \triangleq & S \times S' \\ R_{i+1} & \triangleq & R_i \cap \mathcal{F}(R_i) \end{cases}$$

Avec $\mathcal{F}(R)$ la fonctionnelle suivante :

$$\begin{aligned} \mathcal{F}(R) \triangleq \quad & \{\langle s_1, s_1' \rangle \mid \forall l \in L. \forall s_2 \in S. \\ & \qquad (s_1 \xrightarrow{l} s_2 \Rightarrow \exists s_2' \in S'. s_1' \xrightarrow{l} s_2' \wedge \langle s_2, s_2' \rangle \in R)\} \end{aligned}$$

Calcul pour l'exemple 1

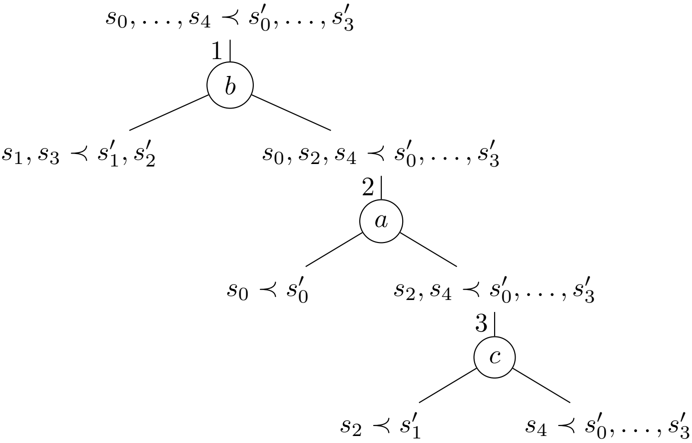
$$
\begin{array}{rcl}
R_0 & = & S \times S' \\
& = & \{\langle s_0, s'_0\rangle, \langle s_0, s'_1\rangle, \langle s_0, s'_2\rangle, \langle s_0, s'_3\rangle \\
& & , \langle s_1, s'_0\rangle, \langle s_1, s'_1\rangle, \langle s_1, s'_2\rangle, \langle s_1, s'_3\rangle \\
& & , \langle s_2, s'_0\rangle, \langle s_2, s'_1\rangle, \langle s_2, s'_2\rangle, \langle s_2, s'_3\rangle \\
& & , \langle s_3, s'_0\rangle, \langle s_3, s'_1\rangle, \langle s_3, s'_2\rangle, \langle s_3, s'_3\rangle \\
& & , \langle s_4, s'_0\rangle, \langle s_4, s'_1\rangle, \langle s_4, s'_2\rangle, \langle s_4, s'_3\rangle\} \\
R_1 & = & \{\langle s_0, s'_0\rangle, \quad \langle s_1, s'_1\rangle, \langle s_1, s'_2\rangle \\
& & , \langle s_2, s'_1\rangle, \quad \langle s_3, s'_1\rangle, \langle s_3, s'_2\rangle \\
& & , \langle s_4, s'_0\rangle, \langle s_4, s'_1\rangle, \langle s_4, s'_2\rangle, \langle s_4, s'_3\rangle\} \\
R_2 & = & R_1
\end{array}
$$

Donc, R_2 est la plus grande simulation sur $S \times S'$. On a :

$$
\begin{array}{rl}
s_0 & \prec s'_0 \\
s_1, s_3 & \prec s'_1, s'_2 \\
s_2 & \prec s'_1 \\
s_4 & \prec s'_0, s'_1, s'_2, s'_3
\end{array}
$$

Exemple 1

- On préfèrera une forme arborescente (lettre par lettre).
- $\simeq$ minimisation des automates.
- On construit des partitions de S.



Exemple 1 (encore des petits carrés)

Passage de la partition : $\Pi_1 \triangleq \begin{cases} s_1, s_3 \prec s'_1, s'_2 \\ s_0, s_2, s_4 \prec s'_0, \ldots, s'_3 \end{cases}$

à la partition suivante : $\Pi_2 \triangleq \begin{cases} s_1, s_3 \prec s'_1, s'_2 \\ s_0 \prec s'_0 \\ s_2, s_4 \prec s'_0, \ldots, s'_3 \end{cases}$

en utilisant la lettre : a sur la branche $s_0, s_2, s_4 \prec s'_0 \ldots, s'_3$

$$
\begin{tikzcd}
s_0 \arrow[r, dash, "\Pi_2"] \arrow[d, "a"] & s'_0 \arrow[d, "a"] \\
s_1 \arrow[r, dash, "\Pi_1"] & \exists s'_1
\end{tikzcd}
\qquad
\begin{tikzcd}
s_0 \arrow[r, dash, "\Pi_2"] \arrow[d, "a"] & s'_0 \arrow[d, "a"] \\
s_2 \arrow[r, dash, "\Pi_1"] & \exists s'_1
\end{tikzcd}
$$

$$
\begin{tikzcd}
s_0 \arrow[r, dash, "\cancel{\Pi_2}"] \arrow[d, "a"] & s'_1, s'_2, s'_3 \arrow[d, "a"] \\
s_1 \arrow[r, dash, "\Pi_1"] & \times
\end{tikzcd}
\quad \text{OU} \quad
\begin{tikzcd}
s_0 \arrow[r, dash, "\cancel{\Pi_2}"] \arrow[d, "a"] & s'_1, s'_2, s'_3 \arrow[d, "a"] \\
s_2 \arrow[r, dash, "\Pi_1"] & \times
\end{tikzcd}
$$

Plan

Notion de Bisimulation

- Relation fondamentale entre systèmes.
- Exprime l'équivalence, la non-discernabilité de systèmes.
- Etend la notion de simulation.
- Plus forte que l'égalité des langages.
- Plus forte que la double simulation :
  $\mathcal{S}$ simule $\mathcal{S}'$, $\mathcal{S}'$ simule $\mathcal{S} \not\Rightarrow \mathcal{S}$ bisimilaire à $\mathcal{S}'$
- Différentes versions, selon le traitement de τ.

Bisimulation forte

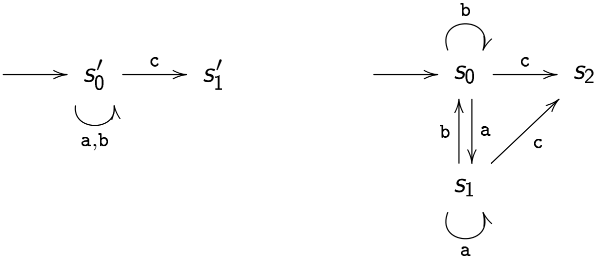
Définition 4 (Bisimulation forte)

On appelle bisimulation forte sur $(S \times S') \cup (S' \times S)$ toute relation R telle que :

- R est une relation de simulation forte sur $(S \times S') \cup (S' \times S)$;
- R est une relation symétrique.

Note : 2 systèmes bisimilaires font vraiment la même chose au même moment.

Exemple 2



Une relation de bisimulation possible est :

$$R \triangleq \{\langle s_0, s'_0\rangle, \langle s_1, s'_0\rangle, \langle s_2, s'_1\rangle, \langle s'_0, s_0\rangle, \langle s'_0, s_1\rangle, \langle s'_1, s_2\rangle\}$$

Contre-exemple 3

$$\begin{tikzcd} {} \arrow[r] & s'_0 \arrow[r, "\mathtt{a}"] & s'_1 \arrow[r, "\mathtt{b}"] & s'_2 \end{tikzcd}$$

$$\begin{tikzcd} {} \arrow[r] & s_0 \arrow[r, "\mathtt{a}"] \arrow[dr, "\mathtt{a}"] & s_1 & \\ & & s_2 \arrow[r, "\mathtt{b}"] & s_3 \end{tikzcd}$$

- $\mathcal{S}'$ simule $\mathcal{S}$ grâce à la relation suivante qui contient $\langle s_0, s'_0 \rangle$:

$$R_1 \triangleq \{\langle s_0, s'_0 \rangle, \langle s_1, s'_1 \rangle, \langle s_2, s'_1 \rangle, \langle s_3, s'_2 \rangle\}$$

- $\mathcal{S}$ simule $\mathcal{S}'$ grâce à la relation suivante qui contient $\langle s'_0, s_0 \rangle$:

$$R_2 \triangleq \{\langle s'_0, s_0 \rangle, \langle s'_1, s_2 \rangle, \langle s'_2, s_3 \rangle\}$$

- Mais aucune bisimulation entre $\mathcal{S}$ et $\mathcal{S}'$ ne contient la paire $\langle s_0, s'_0 \rangle$ (ni $\langle s'_0, s_0 \rangle$).

Structure algébrique

- L'ensemble des bisimulations possède les mêmes opérateurs que l'ensemble des simulations.
- De plus, toute bisimulation étant symétrique, c'est une relation d'équivalence.

Plus grande bisimulation forte

On peut définir la plus grande bisimulation entre S et S', définis sur le même alphabet L. Cette relation existe car :

- L'union de 2 bisimulations est une bisimulation, plus grande ;
- Toute bisimulation est bornée par $(S \times S') \cup (S' \times S)$.

De plus, si S et S' sont finis :

- La plus grande bisimulation est calculable ;
- Toute question "s' est-t'il bisimilaire à s ?" peut se résoudre en considérant la plus grande bisimulation.

Note : On représente cette relation par $\mathcal{S}' \sim \mathcal{S}$

Autres propriétés

Soient 2 S.T.E. $\mathcal{S}$ et $\mathcal{S}'$, définis sur le même alphabet L, tels que $\mathcal{S}$ et $\mathcal{S}'$ sont fortement bisimilaires.

Compositionalité

Tout système $\mathcal{G}$ contenant le sous-système $\mathcal{S}$, noté $\mathcal{G}[\mathcal{S}]$ est bisimilaire à $\mathcal{G}[\mathcal{S}']$.

Autres propriétés

Soient 2 S.T.E. $\mathcal{S}$ et $\mathcal{S}'$, définis sur le même alphabet L, tels que $\mathcal{S}$ et $\mathcal{S}'$ sont fortement bisimilaires.

Conservation des traces

- $Exec(\mathcal{S}) = Exec(\mathcal{S}')$.
- Pour tout s bisimilaire à s', $Traces(s) = Traces(s')$.

Autres propriétés

Soient 2 S.T.E. $\mathcal{S}$ et $\mathcal{S}'$, définis sur le même alphabet L, tels que $\mathcal{S}$ et $\mathcal{S}'$ sont fortement bisimilaires.

Conservation des propriétés temporelles

$\mathcal{S}'$ et $\mathcal{S}$ vérifient les mêmes propriétés temporelles (LTL ou CTL).

Autres propriétés

Soient $\mathcal{A} = \langle \mathcal{S}_{\mathcal{A}}, \mathcal{L}, \mathcal{I}_{\mathcal{A}}, \mathcal{R}_{\mathcal{A}}, \mathcal{F}_{\mathcal{A}} \rangle$ un automate déterministe et $\mathcal{M}$ le résultat de la minimisation de $\mathcal{A}$.

Minimisation des automates

- $\mathcal{M}$ est bisimilaire à $\mathcal{A}$.
- $s \sim s' \Leftrightarrow L(s) = L(s') \Leftrightarrow s \equiv s'$.

Calcul de la plus grande bisimulation forte

Soient 2 S.T.E. $\mathcal{S}$ et $\mathcal{S}'$, définis sur le même alphabet L.

Définition 5 (Plus grande bisimulation entre $\mathcal{S}$ et $\mathcal{S}'$)

La plus grande bisimulation est la limite de la suite (décroissante) $(R_i)_{i\in\mathbb{N}}$:

$$\begin{cases} R_0 & \triangleq & (S \times S') \cup (S' \times S) \\ R_{i+1} & \triangleq & R_i \cap \mathcal{F}(R_i) \cap \mathcal{F}(R_i)^{-1} \end{cases}$$

Avec $\mathcal{F}(R)$ la fonctionnelle suivante :

$$\begin{aligned} \mathcal{F}(R) \triangleq \quad & \{\langle s_1, s_1' \rangle \mid \forall l \in L . \forall s_2 \in S \cup S' . \\ & (s_1 \xrightarrow{l} s_2 \Rightarrow \exists s_2' \in S \cup S' . s_1' \xrightarrow{l} s_2' \wedge \langle s_2, s_2' \rangle \in R)\} \end{aligned}$$

Note : On a toujours $R_i = R_i^{-1}$.

Calcul pour l'exemple 3

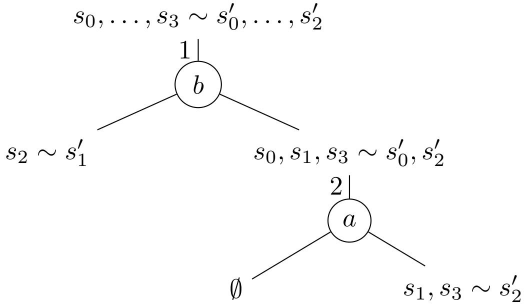
$$
\begin{array}{lcl}
R_0 & = & (S \times S') \cup (S' \times S) \\
R_1 & = & \{\langle s_0, s'_0\rangle, \langle s'_0, s_0\rangle, \langle s_1, s'_2\rangle, \langle s'_2, s_1\rangle, \langle s_2, s'_1\rangle, \langle s'_1, s_2\rangle \\
 & & , \langle s_3, s'_2\rangle, \langle s'_2, s_3\rangle\} \\
R_2 & = & \{\langle s_1, s'_2\rangle, \langle s'_2, s_1\rangle, \langle s_2, s'_1\rangle, \langle s'_1, s_2\rangle, \langle s_3, s'_2\rangle, \langle s'_2, s_3\rangle\} \\
R_3 & = & R_2
\end{array}
$$

Donc, R_3 est la plus grande bisimulation sur $(S \times S') \cup (S' \times S)$.

$$
\begin{array}{c}
s_1 \sim s'_2 \\
s_2 \sim s'_1 \\
s_3 \sim s'_2
\end{array}
$$

Exemple 3

- Sous forme arborescente (lettre par lettre).
- On construit des partitions de $S \times S'$.



Exemple 3 (encore des petits carrés)

Passage de la partition : $\Pi_1 \triangleq \begin{cases} s_2 \sim s'_1 \\ s_0, s_1, s_3 \sim s'_0, s'_2 \end{cases}$

à la partition suivante : $\Pi_2 \triangleq \begin{cases} s_2 \sim s'_1 \\ s_1, s_3 \sim s'_2 \end{cases}$

en utilisant la lettre : a sur la branche $s_0, s_1, s_3 \prec s'_0, s'_2$

$$\begin{tikzcd} s_1 \arrow[r, dash, "\Pi_2"] \arrow[d, "\mathtt{a}"] & s'_2 & s_3 \arrow[r, dash, "\Pi_2"] \arrow[d, "\mathtt{a}"] & s'_2 \\ \times & & \times & \end{tikzcd}$$

$$\begin{tikzcd} s'_2 \arrow[r, dash, "\Pi_2"] \arrow[d, "\mathtt{a}"] & s_1 & s'_2 \arrow[r, dash, "\Pi_2"] \arrow[d, "\mathtt{a}"] & s_3 \\ \times & & \times & \end{tikzcd}$$

Exemple 3 (encore des petits carrés)

Passage de la partition : $\Pi_1 \triangleq \begin{cases} s_2 \sim s'_1 \\ s_0, s_1, s_3 \sim s'_0, s'_2 \end{cases}$

à la partition suivante : $\Pi_2 \triangleq \begin{cases} s_2 \sim s'_1 \\ s_1, s_3 \sim s'_2 \end{cases}$

en utilisant la lettre : a sur la branche $s_0, s_1, s_3 \prec s'_0, s'_2$

$$
\begin{tikzcd}
s_0 \arrow[r, dash, "\cancel{\Pi_2}"] \arrow[d, "\mathtt{a}"] & s'_2 \arrow[d, "\mathtt{a}"] & s'_0 \arrow[r, dash, "\cancel{\Pi_2}"] \arrow[d, "\mathtt{a}"] & s_1 \arrow[d, "\mathtt{a}"] & s'_0 \arrow[r, dash, "\cancel{\Pi_2}"] \arrow[d, "\mathtt{a}"] & s_3 \arrow[d, "\mathtt{a}"] \\
s_1 & \times & s'_1 & \times & s'_1 & \times
\end{tikzcd}
$$

$$
\begin{tikzcd}
s_0 \arrow[r, dash, "\Pi_2"] \arrow[d, "\mathtt{a}"] & s'_0 \arrow[d, "\mathtt{a}"] & s_0 \arrow[r, dash, "\cancel{\Pi_2}"] \arrow[d, "\mathtt{a}"] & s'_0 \arrow[d, "\mathtt{a}"] & s'_0 \arrow[r, dash, "\Pi_2"] \arrow[d, "\mathtt{a}"] & s_0 \arrow[d, "\mathtt{a}"] \\
s_2 \arrow[r, dash, "\Pi_1"] & \exists s'_1 & s_1 \arrow[r, dash, "\cancel{\Pi_1}"] & \cancel{s'_1} & s'_1 \arrow[r, dash, "\Pi_1"] & \exists s_2
\end{tikzcd}
$$

Plan

(Bi)simulation faible

- La (bi)simulation forte traite l'étiquette τ comme un événement ordinaire.
- Une implantation comporte souvent beaucoup d'événements inobservables par rapport à sa spécification (affectations, utilisation de la pile, communications entre composants, etc).
- Ce qui apparaît comme un événement atomique à l'utilisateur se décompose en plusieurs événements (instructions assembleur).

$\rightarrow$ La (bi)simulation forte est donc beaucoup trop restrictive.

$\rightarrow$ On ne peut pas prouver qu'un compilateur est correct, i.e. que le programme source et sa traduction sont en bisimulation forte, alors qu'ils sont équivalents en un certain sens.

$\rightarrow$ **(bi)simulation faible ou observationnelle**.

Simulation faible

Soient $\mathcal{S} = \langle S, L, I, R \rangle$ et $\mathcal{S}' = \langle S', L, I', R' \rangle$ deux S.T.E. définis sur un même alphabet d'événements $\mathcal{L}$.

Définition 6 (Relation de simulation faible)

On dit qu'une relation $Simu \subseteq S \times S'$ est une relation de simulation faible de S par S' ssi :

$$
\begin{array}{l}
\forall s_1, s_2 \in S, l \in L, s_1' \in S'. \\
\quad (\langle s_1, s_1' \rangle \in Simu \;\wedge\; s_1 \xrightarrow{l} s_2 (\in R) \\
\quad \Rightarrow \\
\quad \exists s_2'. \langle s_2, s_2' \rangle \in Simu \;\wedge\; s_1' \overset{l}{\Rightarrow} s_2' (\in R')) \\
\wedge \\
\quad (\langle s_1, s_1' \rangle \in Simu \;\wedge\; s_1 \xrightarrow{\tau} s_2 (\in R) \\
\quad \Rightarrow \\
\quad \exists s_2'. \langle s_2, s_2' \rangle \in Simu \;\wedge\; s_1' \Rightarrow s_2' (\in R'))
\end{array}
$$

Simulation faible

Graphiquement parlant :

$$
\begin{tikzcd}
s_1 \arrow[r, dash, "Simu"] \arrow[d, "l"] & s_1' \arrow[d, Rightarrow, dashed, "l"] \\
s_2 \arrow[r, dash, dashed, "Simu"] & \exists s_2'
\end{tikzcd}
\qquad
\begin{tikzcd}
s_1 \arrow[r, dash, "Simu"] \arrow[d, "\tau"] & s_1' \arrow[d, Rightarrow, dashed] \\
s_2 \arrow[r, dash, dashed, "Simu"] & \exists s_2'
\end{tikzcd}
$$

C'est-à-dire :

- Toute l-transition de $\mathcal{S}$ est simulée par une l-transition de $\mathcal{S}'$, modulo quelques τ-transitions de $\mathcal{S}'$.
- Toute τ-transition de $\mathcal{S}$ est simulée par une séquence de τ-transitions de $\mathcal{S}'$.

Simulation faible entre S.T.E.

Simulation entre états

On dit que $s' \in S'$ simule faiblement $s \in S$ ssi il existe une relation de simulation faible $Simu \subseteq S \times S'$ vérifiant la définition 6 et telle que :

$$\langle s, s' \rangle \in Simu$$

Simulation entre S.T.E.

On dit que $\mathcal{S}'$ simule faiblement $\mathcal{S}$ ssi il existe une relation de simulation faible $Simu \subseteq S \times S'$ vérifiant la définition 6 et telle que :

$$\forall i \in I . \exists i' \in I' . \langle i, i' \rangle \in Simu$$

Note : toute (bi)simulation forte est une (bi)simulation faible.

Exemple 4

Soient 2 systèmes $\mathcal{S}$ et $\mathcal{S}'$.

$$\longrightarrow s'_0 \xrightarrow{\tau} s'_1 \xrightarrow{\mathtt{a}} s'_2 \xrightarrow{\tau} s'_3 \xrightarrow{\mathtt{b}} s'_4 \qquad \longrightarrow s_0 \xrightarrow{\mathtt{a}} s_1 \xrightarrow{\mathtt{b}} s_2$$

$\mathcal{S}'$ simule faiblement $\mathcal{S}$:

$$R \triangleq \{\langle s_0, s'_0\rangle, \langle s_1, s'_2\rangle, \langle s_2, s'_4\rangle\}$$

On peut également ajouter la paire $\langle s_1, s'_3\rangle$.
Réciproquement, $\mathcal{S}$ simule faiblement $\mathcal{S}'$:

$$R \triangleq \{\langle s'_0, s_0\rangle, \langle s'_1, s_0\rangle, \langle s'_2, s_1\rangle, \langle s'_3, s_1\rangle, \langle s'_4, s_2\rangle\}$$

En fait, $\mathcal{S}$ et $\mathcal{S}'$ sont ici faiblement bisimilaires.

Exemple 5

Soient 3 systèmes $\mathcal{S}$, $\mathcal{S}'$ et $\mathcal{S}''$.

$$
\begin{tikzcd}
\arrow[r] & s_0'' \arrow[r, "\mathtt{a}"] \arrow[dr, "\mathtt{b}"] & s_1'' \\
& & s_2''
\end{tikzcd}
\qquad
\begin{tikzcd}
\arrow[r] & s_0' \arrow[r, "\tau"] \arrow[dr, "\mathtt{b}"] & s_1' \arrow[r, "\mathtt{a}"] & s_3' \\
& & s_2' &
\end{tikzcd}
\qquad
\begin{tikzcd}
\arrow[r] & s_0 \arrow[r, "\tau"] \arrow[dr, "\tau"] & s_1 \arrow[r, "\mathtt{a}"] & s_3 \\
& & s_2 \arrow[r, "\mathtt{b}"] & s_4
\end{tikzcd}
$$

$\mathcal{S}''$ simule faiblement $\mathcal{S}'$:

$$R \triangleq \{\langle s_0', s_0''\rangle, \langle s_1', s_0''\rangle, \langle s_2', s_2''\rangle, \langle s_3', s_1''\rangle\}$$

$\mathcal{S}'$ simule faiblement $\mathcal{S}$:

$$R \triangleq \{\langle s_0, s_0'\rangle, \langle s_1, s_1'\rangle, \langle s_2, s_0'\rangle, \langle s_3, s_3'\rangle, \langle s_4, s_2'\rangle\}$$

Propriétés

- Même structure algébrique que la simulation forte.
- Existence d'une plus grande simulation faible.
- La compositionalité de la simulation faible est beaucoup plus faible. Dans le cas général, on ne peut pas remplacer une partie d'un système par une autre partie faiblement similaire et s'attendre à ce que le résultat soit faiblement similaire.
- Les propriétés temporelles de sûreté sur les événements observables sont conservées.
- Si on interprète τ comme un élément neutre (i.e. $\tau = \epsilon$), alors les traces sont conservées.
- La plus grande simulation faible peut être calculée en suivant le même principe que pour la simulation forte, avec la définition 6 au lieu de la définition 2.

Plan

Bisimulation faible

Définition 7 (Bisimulation faible)

On appelle bisimulation faible sur $(S \times S') \cup (S' \times S)$ toute relation R telle que :

- R est une relation de simulation faible sur $(S \times S') \cup (S' \times S)$.
- R est une relation symétrique.

Propriétés

- Même structure algébrique que la bisimulation forte.
- Existence d'une plus grande bisimulation faible, notée $\approx$.
- La compositionalité de la bisimulation faible est beaucoup plus faible. Dans le cas général, on ne peut pas remplacer une partie d'un système par une autre partie faiblement bisimilaire et s'attendre à ce que le résultat soit faiblement bisimilaire (voir contre-exemple 8).
- Les propriétés temporelles sur les événements observables sont identiques.
- Si on interprète τ comme un élément neutre (i.e. $\tau = \epsilon$), alors les traces sont identiques.

Exemple 6

Soient $\mathcal{S}$ et $\mathcal{S}'$ 2 S.T.E. définis sur le même alphabet.

$$\longrightarrow s'_0 \xrightarrow{\tau} s'_1 \xrightarrow{\mathtt{a}} s'_2 \qquad\qquad \longrightarrow s_0 \xrightarrow{\mathtt{a}} s_1$$

$\mathcal{S}$ et $\mathcal{S}'$ sont faiblement bisimilaires, comme le prouve la relation suivante $R \triangleq \{\langle s_0, s'_0\rangle, \langle s'_0, s_0\rangle, \langle s_0, s'_1\rangle, \langle s'_1, s_0\rangle, \langle s_1, s'_2\rangle, \langle s'_2, s_1\rangle\}$

$$\begin{tikzcd}
s_0 \arrow[r, dash, "R"] \arrow[d, "\mathtt{a}"] & s'_0 \arrow[d, Rightarrow, "\mathtt{a}"] & s_0 \arrow[r, dash, "R"] \arrow[d, "\mathtt{a}"] & s'_1 \arrow[d, Rightarrow, "\mathtt{a}"] & s_1 \arrow[r, dash, "R"] \arrow[d, "\mathtt{a}"] & s'_2 \\
s_1 \arrow[r, dash, "R"] & \exists s'_2 & s_1 \arrow[r, dash, "R"] & \exists s'_2 & \times &
\end{tikzcd}$$

$$\begin{tikzcd}
s'_0 \arrow[r, dash, "R"] \arrow[d, "\mathtt{a}"] & s_0 & s'_1 \arrow[r, dash, "R"] \arrow[d, "\mathtt{a}"] & s_0 \arrow[d, Rightarrow, "\mathtt{a}"] & s'_2 \arrow[r, dash, "R"] \arrow[d, "\mathtt{a}"] & s_1 \\
\times & & s'_2 \arrow[r, dash, "R"] & \exists s_1 & \times &
\end{tikzcd}$$

Exemple 6

Soient $\mathcal{S}$ et $\mathcal{S}'$ 2 S.T.E. définis sur le même alphabet.

$$\longrightarrow s'_0 \xrightarrow{\tau} s'_1 \xrightarrow{\text{a}} s'_2 \qquad\qquad \longrightarrow s_0 \xrightarrow{\text{a}} s_1$$

$\mathcal{S}$ et $\mathcal{S}'$ sont faiblement bisimilaires, comme le prouve la relation suivante $R \triangleq \{\langle s_0, s'_0\rangle, \langle s'_0, s_0\rangle, \langle s_0, s'_1\rangle, \langle s'_1, s_0\rangle, \langle s_1, s'_2\rangle, \langle s'_2, s_1\rangle\}$

$$\begin{tikzcd}
s_0 \arrow[r, dash, "R"] \arrow[d, "\tau"] & s'_0 & s_0 \arrow[r, dash, "R"] \arrow[d, "\tau"] & s'_1 & s_1 \arrow[r, dash, "R"] \arrow[d, "\tau"] & s'_2 \\
\times & & \times & & \times &
\end{tikzcd}$$

$$\begin{tikzcd}
s'_0 \arrow[r, dash, "R"] \arrow[d, "\tau"] & s_0 \arrow[d, Rightarrow] & s'_1 \arrow[r, dash, "R"] \arrow[d, "\tau"] & s_0 & s'_2 \arrow[r, dash, "R"] \arrow[d, "\tau"] & s_1 \\
s'_1 \arrow[r, dash, "R"] & \exists s_0 & \times & & \times &
\end{tikzcd}$$

Exemple 7

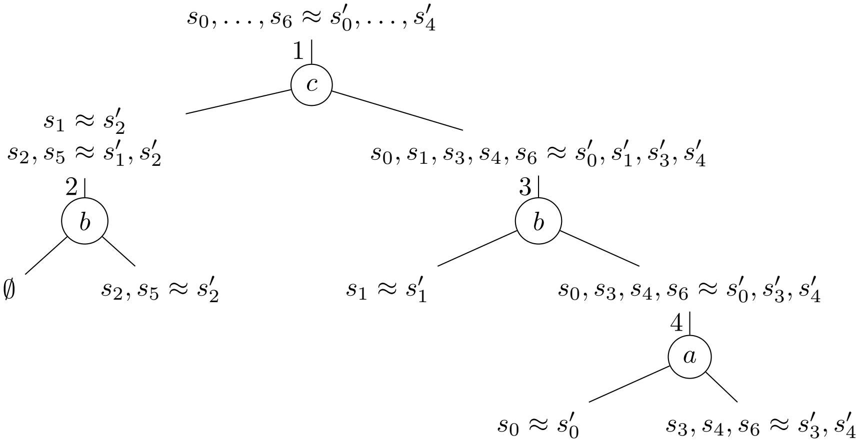
Soient $\mathcal{S}$ et $\mathcal{S}'$ 2 S.T.E. définis sur le même alphabet.

$$\begin{tikzcd}
& & & s'_4 & \\
{} \arrow[r] & s'_0 \arrow[r, "a"] & s'_1 \arrow[r, "\tau"] \arrow[ur, "b"] & s'_2 \arrow[r, "c"] & s'_3
\end{tikzcd}
\qquad
\begin{tikzcd}
& & & s_4 & \\
{} \arrow[r] & s_0 \arrow[r, "a"] \arrow[dr, "a"] & s_1 \arrow[r, "\tau"] \arrow[ur, "b"] & s_2 \arrow[r, "c"] & s_3 \\
& & s_5 \arrow[r, "c"] & s_6 &
\end{tikzcd}$$

- $\mathcal{S}$ et $\mathcal{S}'$ sont faiblement bisimilaires.
- On le prouve, par exemple, en démontrant : $s_0 \approx s'_0$.

Exemple 7

- Sous forme arborescente (lettre par lettre).
- On construit des partitions de $S \cup S'$.



Exemple 7 (encore des petits carrés)

- Cas $s_0 \approx s'_0$:

$$
\begin{tikzcd}
s_0 \arrow[r, dash, "\approx"] \arrow[d, "\mathtt{a}"] & s'_0 \arrow[d, Rightarrow, "\mathtt{a}"] & s_0 \arrow[r, dash, "\approx"] \arrow[d, "\mathtt{a}"] & s'_0 \arrow[d, Rightarrow, "\mathtt{a}"] & s'_0 \arrow[r, dash, "\approx"] \arrow[d, "\mathtt{a}"] & s_0 \arrow[d, Rightarrow, "\mathtt{a}"] \\
s_1 \arrow[r, dash, "\approx"] & \exists s'_1 & s_5 \arrow[r, dash, "\approx"] & \exists s'_2 & s'_1 \arrow[r, dash, "\approx"] & \exists s_1
\end{tikzcd}
$$

- Cas $s_1 \approx s'_1$:

$$
\begin{tikzcd}
s_1 \arrow[r, dash, "\approx"] \arrow[d, "\mathtt{b}"] & s'_1 \arrow[d, Rightarrow, "\mathtt{b}"] & s'_1 \arrow[r, dash, "\approx"] \arrow[d, "\mathtt{b}"] & s_1 \arrow[d, Rightarrow, "\mathtt{b}"] \\
s_4 \arrow[r, dash, "\approx"] & \exists s'_4 & s'_4 \arrow[r, dash, "\approx"] & \exists s_4
\end{tikzcd}
$$

Contre-exemple 8

Soient $\mathcal{S}$, $\mathcal{S}'$ et $\mathcal{S}''$ 3 S.T.E. définis sur le même alphabet.

$$
\begin{tikzcd}
{} \arrow[r] & s_0'' \arrow[r, "\mathtt{a}"] \arrow[dr, "\mathtt{b}"] & s_1'' \\
 & & s_2''
\end{tikzcd}
\qquad
\begin{tikzcd}
{} \arrow[r] & s_0' \arrow[r, "\tau"] \arrow[dr, "\mathtt{b}"] & s_1' \arrow[r, "\mathtt{a}"] & s_3' \\
 & & s_2' &
\end{tikzcd}
\qquad
\begin{tikzcd}
{} \arrow[r] & s_0 \arrow[r, "\tau"] \arrow[dr, "\tau"] & s_1 \arrow[r, "\mathtt{a}"] & s_3 \\
 & & s_2 \arrow[r, "\mathtt{b}"] & s_4
\end{tikzcd}
$$

- On a : $\mathcal{S} \not\approx \mathcal{S}'$, $\mathcal{S} \not\approx \mathcal{S}''$, $\mathcal{S}' \not\approx \mathcal{S}''$.
- On prouve $\mathcal{S}' \not\approx \mathcal{S}''$, en démontrant $s_0' \not\approx s_0''$.

Contre-exemple 8

- On construit la plus grande bisimulation faible entre $\mathcal{S}'$ et $\mathcal{S}''$.
- On vérifie qu'elle ne contient pas la paire $\langle s'_0, s''_0 \rangle$ (ni $\langle s'_0, s''_0 \rangle$).