Bisimulation

Implementation relations

- A binary relation on transition systems
  - when does a transition systems correctly implement another?
- Important for system synthesis
  - stepwise refinement of a system specification TS into an "implementation" TS'
- Important for system analysis
  - use the implementation relation as a means for abstraction
  - replace $TS \models \varphi$ by $TS' \models \varphi$ where $|\,TS'\,| \ll |\,TS\,|$ such that:

$$TS \models \varphi \text{ iff } TS' \models \varphi \quad \text{or} \quad TS' \models \varphi \;\Rightarrow\; TS \models \varphi$$

$\Rightarrow$ Focus on state-based bisimulation and simulation
- logical characterization: which logical formulas are preserved by bisimulation?

Bisimulation equivalence

Let $TS_i = (S_i, Act_i, \rightarrow_i, I_i, AP, L_i)$, i=1,2, be transition systems.
A **bisimulation** for (TS_1, TS_2) is a **binary relation** $\mathcal{R} \subseteq S_1 \times S_2$ such that:

1. $\forall s_1 \in I_1\, \exists s_2 \in I_2.\ (s_1, s_2) \in \mathcal{R}$ **and** $\forall s_2 \in I_2\, \exists s_1 \in I_1.\ (s_1, s_2) \in \mathcal{R}$
2. for all states $s_1 \in S_1$, $s_2 \in S_2$ with $(s_1, s_2) \in \mathcal{R}$ it holds:

   2.1 $L_1(s_1) = L_2(s_2)$

   2.2 if $s_1' \in Post(s_1)$ then there exists $s_2' \in Post(s_2)$ with $(s_1', s_2') \in \mathcal{R}$

   2.3 if $s_2' \in Post(s_2)$ then there exists $s_1' \in Post(s_1)$ with $(s_1', s_2') \in \mathcal{R}$

TS_1 and TS_2 are bisimilar, denoted $TS_1 \sim TS_2$, if there exists a bisimulation for (TS_1, TS_2)

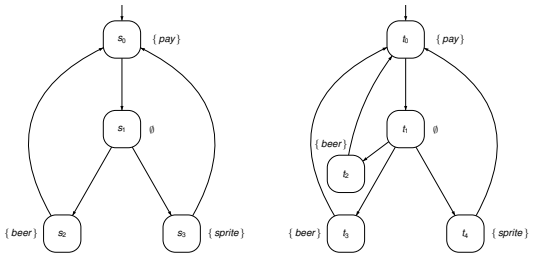
Bisimulation equivalence

$$\begin{array}{ccc} s_1 & \rightarrow & s_1' \\ \mathcal{R} & & \\ s_2 & & \end{array} \quad \text{can be completed to} \quad \begin{array}{ccc} s_1 & \rightarrow & s_1' \\ \mathcal{R} & & \mathcal{R} \\ s_2 & \rightarrow & s_2' \end{array}$$

and

$$\begin{array}{ccc} s_1 & & \\ \mathcal{R} & & \\ s_2 & \rightarrow & s_2' \end{array} \quad \text{can be completed to} \quad \begin{array}{ccc} s_1 & \rightarrow & s_1' \\ \mathcal{R} & & \mathcal{R} \\ s_2 & \rightarrow & s_2' \end{array}$$

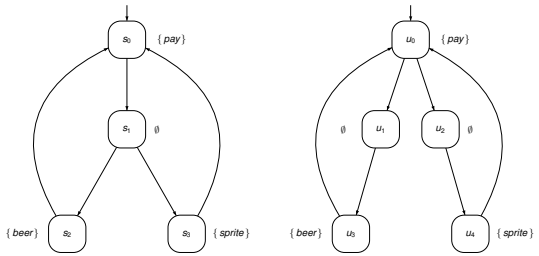
Example (1)



$\mathcal{R} = \{(s_0, t_0), (s_1, t_1), (s_2, t_2), (s_2, t_3), (s_3, t_4)\}$

is a bisimulation for (TS_1, TS_2) where $AP = \{ pay, beer, sprite \}$

Example (2)



$TS_1 \not\sim TS_3$ for $AP = \{ pay, beer, sprite \}$

But: $\{ (s_0, u_0), (s_1, u_1), (s_1, u_2), (s_2, u_3), (s_2, u_4), (s_3, u_3), (s_3, u_4) \}$

is a bisimulation for (TS_1, TS_3) for $AP = \{ pay, drink \}$

$\sim$ is an equivalence

For any transition systems TS, TS_1, TS_2 and TS_3 over AP:

- $TS \sim TS$ (reflexivity)
- $TS_1 \sim TS_2$ implies $TS_2 \sim TS_1$ (symmetry)
- $TS_1 \sim TS_2$ and $TS_2 \sim TS_3$ implies $TS_1 \sim TS_3$ (transitivity)

Bisimulation on paths

Whenever we have:

$$\begin{array}{ccccccccccc} s_0 & \to & s_1 & \to & s_2 & \to & s_3 & \to & s_4 \ldots\ldots \\ \mathcal{R} & & & & & & & & \\ t_0 & & & & & & & & \end{array}$$

this can be completed to

$$\begin{array}{ccccccccccc} s_0 & \to & s_1 & \to & s_2 & \to & s_3 & \to & s_4 \ldots\ldots \\ \mathcal{R} & & \mathcal{R} & & \mathcal{R} & & \mathcal{R} & & \mathcal{R} \\ t_0 & \to & t_1 & \to & t_2 & \to & t_3 & \to & t_4 \ldots\ldots \end{array}$$

proof: by induction on index i of state s_i

Bisimulation vs. trace equivalence

$TS_1 \sim TS_2$ implies $Traces(TS_1) = Traces(TS_2)$

bisimilar transition systems thus satisfy the same LT properties!

Bisimulation on states

$\mathcal{R} \subseteq S \times S$ is a **bisimulation** on TS if for any $(s_1, s_2) \in \mathcal{R}$:

- $L(s_1) = L(s_2)$
- if $s_1' \in Post(s_1)$ then there exists an $s_2' \in Post(s_2)$ with $(s_1', s_2') \in \mathcal{R}$
- if $s_2' \in Post(s_2)$ then there exists an $s_1' \in Post(s_1)$ with $(s_1', s_2') \in \mathcal{R}$

s_1 and s_2 are **bisimilar**, $s_1 \sim_{TS} s_2$,
if $(s_1, s_2) \in \mathcal{R}$ for some bisimulation $\mathcal{R}$ for TS

$s_1 \sim_{TS} s_2$ if and only if $TS_{s_1} \sim TS_{s_2}$

TS_s is the transition system obtained from TS by declaring s as the initial state.

Coarsest bisimulation

$\sim_{TS}$ is an equivalence and the coarsest bisimulation for TS

Quotient transition system

For $TS = (S, Act, \rightarrow, I, AP, L)$ and bisimulation $\sim_{TS} \subseteq S \times S$ on TS let

$TS/\sim_{TS} = (S', \{\tau\}, \rightarrow', I', AP, L')$ be the **quotient** of TS under $\sim_{TS}$

where

- $S' = S/\sim_{TS} = \{[s]_\sim \mid s \in S\}$ with $[s]_\sim = \{s' \in S \mid s \sim_{TS} s'\}$
- $\rightarrow'$ is defined by: $\dfrac{s \xrightarrow{\alpha} s'}{[s]_\sim \xrightarrow{\tau} [s']_\sim}$
- $I' = \{[s]_\sim \mid s \in I\}$
- $L'([s]_\sim) = L(s)$

The Bakery algorithm

$$P_1 :: \left[\begin{array}{l} \textbf{loop forever do} \\ \left[\begin{array}{ll} & \textbf{noncritical} \\ n_1 : & y_1 := y_2 + 1 \\ w_1 : & \textbf{await } (y_2 = 0 \vee y_1 < y_2) \\ c_1 : & \textbf{critical} \\ & y_1 := 0 \end{array}\right] \end{array}\right] \quad \| \quad P_2 :: \left[\begin{array}{l} \textbf{loop forever do} \\ \left[\begin{array}{ll} & \textbf{noncritical} \\ n_1 : & y_2 := y_1 + 1 \\ w_1 : & \textbf{await } (y_1 = 0 \vee y_2 < y_1) \\ c_1 : & \textbf{critical} \\ & y_2 := 0 \end{array}\right] \end{array}\right]$$

Example path fragment

process P_1	process P_2	y_1	y_2	effect
n_1	n_2	0	0	P_1 requests access to critical section
w_1	n_2	1	0	P_2 requests access to critical section
w_1	w_2	1	2	P_1 enters the critical section
c_1	w_2	1	2	P_1 leaves the critical section
n_1	w_2	0	2	P_1 requests access to critical section
w_1	w_2	3	2	P_2 enters the critical section
w_1	c_2	3	2	P_2 leaves the critical section
w_1	n_2	3	0	P_2 requests access to critical section
w_1	w_2	3	4	P_1 enters the critical section
...	...	..	..	...

Data abstraction

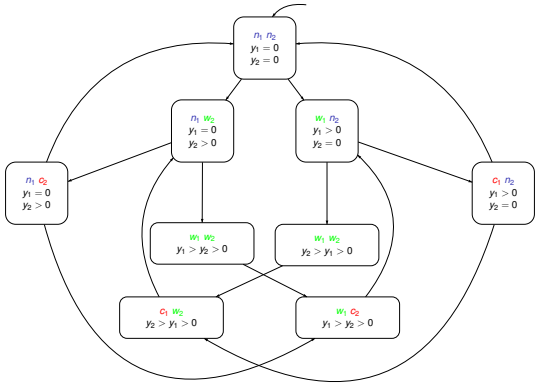
Function f maps a reachable state of TS_{Bak} onto an abstract one in TS^{abs}_{Bak}
Let $s = \langle \ell_1, \ell_2, y_1 = b_1, y_2 = b_2 \rangle$ be a state of TS_{Bak} with
$\ell_i \in \{ n_i, w_i, c_i \}$ and $b_i \in \mathbb{N}$
Then:

$$f(s) \;=\; \begin{cases} \langle \ell_1, \ell_2, y_1 = 0, y_2 = 0 \rangle & \text{if } b_1 = b_2 = 0 \\ \langle \ell_1, \ell_2, y_1 = 0, y_2 > 0 \rangle & \text{if } b_1 = 0 \text{ and } b_2 > 0 \\ \langle \ell_1, \ell_2, y_1 > 0, y_2 = 0 \rangle & \text{if } b_1 > 0 \text{ and } b_2 = 0 \\ \langle \ell_1, \ell_2, y_1 > y_2 > 0 \rangle & \text{if } b_1 > b_2 > 0 \\ \langle \ell_1, \ell_2, y_2 > y_1 > 0 \rangle & \text{if } b_2 > b_1 > 0 \end{cases}$$

$\mathcal{R} = \{ (s, f(s)) \mid s \in S \}$ is a bisimulation for $(TS_{Bak}, TS^{abs}_{Bak})$

for any subset of $AP = \{ crit_i, wait_i, crit_i \mid i = 1, 2 \}$

Bisimulation quotient



$TS^{abs}_{Bak} \;=\; TS_{Bak} / \sim \quad \text{for} \quad AP = \{ crit_1, crit_2 \}$

Remarks

- In this example, data abstraction yields a bisimulation relation
  - (typically, only a simulation relation is obtained, more later)
- $TS^{abs}_{Bak} \models \varphi$ with, e.g.,:
  - $\Box(\neg crit_1 \vee \neg crit_2)$ and $(\Box\Diamond wait_1 \Rightarrow \Box\Diamond crit_1) \wedge (\Box\Diamond wait_2 \Rightarrow \Box\Diamond crit_2)$
- Since $TS^{abs}_{Bak} \sim TS_{Bak}$, it follows $TS_{Bak} \models \varphi$
- Note: $Traces(TS^{abs}_{Bak}) = Traces(TS_{Bak})$

Bisimulation vs. CTL* and CTL equivalence

Let TS be a finite transition system without terminal states, and let s_1, s_2 states in TS

The following statements are equivalent:

(1) $s_1 \sim_{TS} s_2$

(2) s_1 and s_2 are CTL-equivalent, i.e., $s_1 \equiv_{CTL} s_2$

(3) s_1 and s_2 are CTL*-equivalent, i.e., $s_1 \equiv_{CTL^*} s_2$

this is proven in three steps: $\equiv_{CTL} \subseteq \sim_{TS} \subseteq \equiv_{CTL^*} \subseteq \equiv_{CTL}$

equivalence is also obtained for any sub-logic containing $\neg$, $\wedge$ and EX

CTL* equivalence

States s_1 and s_2 in TS (over AP) are CTL*-equivalent:

$s_1 \equiv_{CTL^*} s_2$ if and only if $(s_1 \models \Phi$ iff $s_2 \models \Phi)$

for all CTL* state formulas over AP

$TS_1 \equiv_{CTL^*} TS_2$ if and only if $(TS_1 \models \Phi$ iff $TS_2 \models \Phi)$

for any sublogic of CTL*, logical equivalence is defined analogously