Verification – Lecture 20
Bisimulation

Bernd Finkbeiner – Sven Schewe
Rayna Dimitrova – Lars Kuhtz – Anne Proetzsch

Wintersemester 2007/2008

REVIEW

Bisimulation equivalence

Let $S_i = (Q_i, Q_{0,i}, E_i, L_i)$, $i{=}1,2$, be two state graphs over *AP*.

A *bisimulation* for (S_1, S_2) is a binary relation $\mathcal{R} \subseteq Q_1 \times Q_2$ such that:

1. $\forall q_1 \in Q_{0,1}\, \exists q_2 \in Q_{0,2}.\ (q_1, q_2) \in \mathcal{R}$ and
   $\forall q_2 \in Q_{0,2}\, \exists q_1 \in Q_{0,1}.\ (q_1, q_2) \in \mathcal{R}$

2. for all states $q_1 \in Q_1$, $q_2 \in Q_2$ with $(q_1, q_2) \in \mathcal{R}$ it holds:

   (a) $L_1(q_1) = L_2(q_2)$

   (b) if $q_1' \in$ *Successors*(q_1) then there exists $q_2' \in$ *Successors*(q_2) with $(q_1', q_2') \in \mathcal{R}$

   (c) if $q_2' \in$ *Successors*(q_2) then there exists $q_1' \in$ *Successors*(q_1) with $(q_1', q_2') \in \mathcal{R}$

S_1 and S_2 are bisimilar, denoted $S_1 \sim S_2$, if there exists a bisimulation for (S_1, S_2)

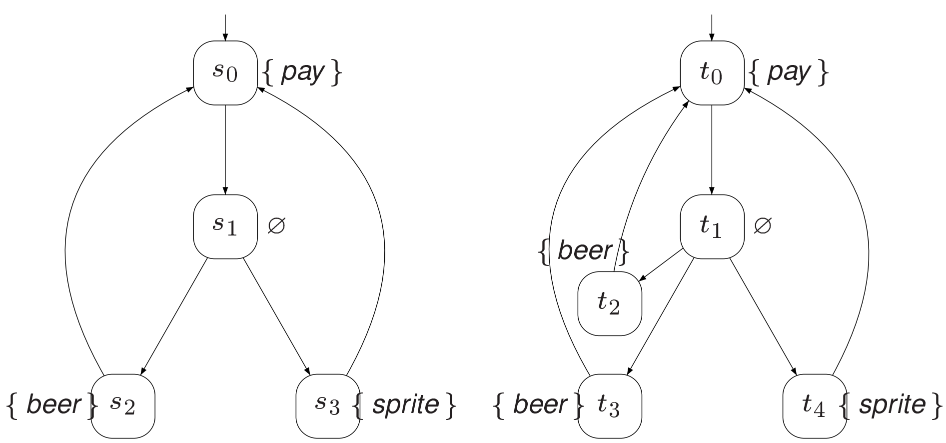
Bisimulation equivalence

$$\begin{array}{ccc} q_1 & \to & q_1' \\ \mathcal{R} & & \\ q_2 & & \end{array} \quad \text{can be completed to} \quad \begin{array}{ccc} q_1 & \to & q_1' \\ \mathcal{R} & & \mathcal{R} \\ q_2 & \to & q_2' \end{array}$$

and

$$\begin{array}{ccc} q_1 & & \\ \mathcal{R} & & \\ q_2 & \to & q_2' \end{array} \quad \text{can be completed to} \quad \begin{array}{ccc} q_1 & \to & q_1' \\ \mathcal{R} & & \mathcal{R} \\ q_2 & \to & q_2' \end{array}$$

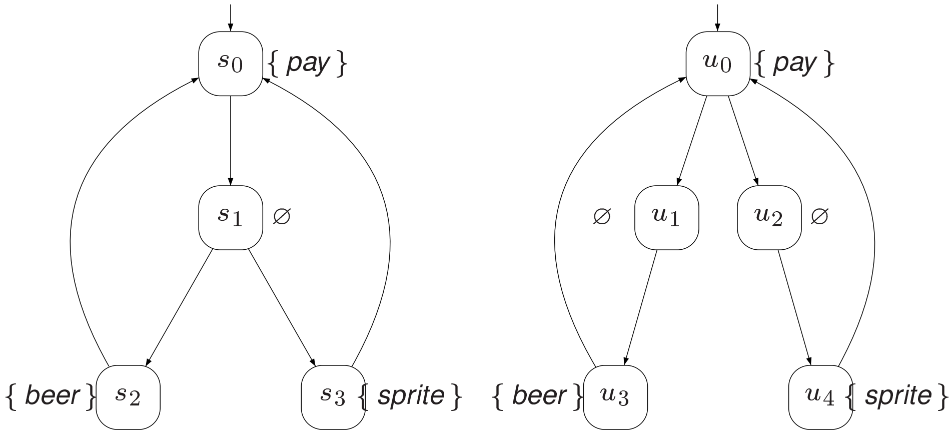
Example (1)



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

is a bisimulation for (S_1, S_2) where $AP = \{ \textit{pay}, \textit{beer}, \textit{sprite} \}$

Example (2)



$S_1 \not\sim S_3$ for $AP = \{ \textit{pay}, \textit{beer}, \textit{sprite} \}$

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 (S_1, S_3) for $AP = \{ \textit{pay}, \textit{drink} \}$

$\sim$ is an equivalence

For any transition systems S, S_1, S_2 and S_3 over AP:

$S \sim S$ (reflexivity)

$S_1 \sim S_2$ implies $S_2 \sim S_1$ (symmetry)

$S_1 \sim S_2$ and $S_2 \sim S_3$ implies $S_1 \sim S_3$ (transitivity)

Bisimulation on paths

Whenever we have:

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

this can be completed to

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

proof: by induction on index i of state s_i

REVIEW

Bisimulation vs. trace equivalence

$$\boxed{S_1 \sim S_2 \quad \text{implies} \quad \mathit{Traces}(S_1) = \mathit{Traces}(S_2)}$$

bisimilar transition systems thus satisfy the same LT properties!

Bisimulation on states

$\mathcal{R} \subseteq S \times S$ is a *bisimulation* on *S* if for any $(q_1, q_2) \in \mathcal{R}$:

- $L(q_1) = L(q_2)$
- if $q_1' \in$ *Successors*(q_1) then there exists an $q_2' \in$ *Successors*(q_2) with $(q_1', q_2') \in \mathcal{R}$
- if $q_2' \in$ *Successors*(q_2) then there exists an $q_1' \in$ *Successors*(q_1) with $(q_1', q_2') \in \mathcal{R}$

q_1 and q_2 are *bisimilar*, $q_1 \sim_S q_2$, if $(q_1, q_2) \in \mathcal{R}$ for some bisimulation $\mathcal{R}$ for *S*

$q_1 \sim_S q_2$ if and only if $S_{q_1} \sim S_{q_2}$

Coarsest bisimulation

$\sim_S$ is an equivalence and the coarsest bisimulation for *S*

Quotient state graph

For $S = (Q, Q_0, E, L)$ and bisimulation $\sim_S \subseteq S \times S$ on S let

$$S/\sim_S = \ (Q', Q'_0, E', L') \quad \text{be the } \textit{quotient} \text{ of } S \text{ under } \sim_S$$

where

- $Q' = S/\sim_S = \ \{ [q]_\sim \mid q \in Q \}$ with $[q]_\sim \ = \ \{ q' \in Q \mid q \sim_S q' \}$
- $Q'_0 = \{ [q]_\sim \mid q \in Q_0 \}$
- $E' = \{([q]_\sim, [q']_\sim) \mid (q, q') \in E\}$
- $L'([q]_\sim) = L(q)$

note that $S \sim S/\sim_S$ Why?

The Bakery algorithm

P_1 ::

```
loop forever do
        noncritical
  n1 :  y1 := y2 + 1
  w1 :  await (y2 = 0 ∨ y1 < y2)
  c1 :  critical
        y1 := 0
```

|| P_2 ::

```
loop forever do
        noncritical
  n1 :  y2 := y1 + 1
  w1 :  await (y1 = 0 ∨ y2 < y1)
  c1 :  critical
        y2 := 0
```

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_2 enters the critical section
...	...	..	..	...

Data abstraction

Function f maps a reachable state of S_{Bak} onto an abstract one in S_{Bak}^{abs}

Let $s = \langle \ell_1, \ell_2, y_1 = b_1, y_2 = b_2 \rangle$ be a state of S_{Bak} with $\ell_i \in \{ n_i, w_i, c_i \}$ and $b_i \in \mathbb{N}$

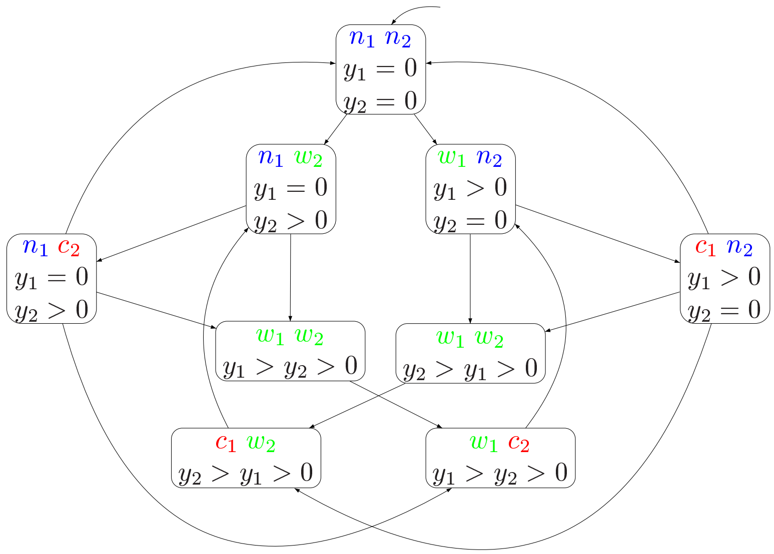
Then:

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

It follows: $\mathcal{R} = \{ (s, f(s)) \mid s \in S \}$ is a bisimulation for (S_{Bak}, S_{Bak}^{abs})

for any subset of $AP = \{ \mathit{noncrit}_i, \mathit{wait}_i, \mathit{crit}_i \mid i = 1, 2 \}$

Bisimulation quotient



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

Remarks

- Data abstraction yields a bisimulation relation
  - in this example; typically a simulation relation is obtained
- $S^{abs}_{Bak} \models \varphi$ 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 $S^{abs}_{Bak} \sim S_{Bak}$, it follows $S_{Bak} \models \varphi$
- Note: $Traces(S^{abs}_{Bak}) = Traces(S_{Bak})$
  - but checking trace equivalence is PSPACE-complete
  - while checking bisimulation equivalence is in poly-time

Syntax of CTL*

CTL* *state-formulas* are formed according to:

$$\Phi ::= \text{true} \;\big|\; a \;\big|\; \Phi_1 \wedge \Phi_2 \;\big|\; \neg\Phi \;\big|\; \exists\varphi$$

where $a \in AP$ and φ is a path-formula

CTL* *path-formulas* are formed according to the grammar:

$$\varphi ::= \Phi \;\big|\; \varphi_1 \wedge \varphi_2 \;\big|\; \neg\varphi \;\big|\; \bigcirc\varphi \;\big|\; \varphi_1 \,\mathsf{U}\, \varphi_2$$

where Φ is a state-formula, and φ, φ_1 and φ_2 are path-formulas

CTL* equivalence

States q_1 and q_2 in *S* (over *AP*) are CTL*-equivalent:

$$q_1 \equiv_{CTL^*} q_2 \quad \text{if and only if} \quad (q_1 \models \Phi \;\text{ iff }\; q_2 \models \Phi)$$

for all CTL* state formulas over *AP*

$$S_1 \equiv_{CTL^*} S_2 \quad \text{if and only if} \quad (S_1 \models \Phi \;\text{ iff }\; S_2 \models \Phi)$$

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

Bisimulation vs. CTL* and CTL equivalence

Let S be a *finite* state graph and s, s' states in S

The following statements are equivalent:

(1) $s \sim_S s'$

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

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

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

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

The importance of this result

- CTL and CTL* equivalence coincide
  - despite the fact that CTL* is more expressive than CTL
- Bisimilar transition systems preserve the same CTL* formulas
  - and thus the same LTL formulas (and LT properties)
- Non-bisimilarity can be shown by a single CTL (or CTL*) formula
  - $S_1 \models \Phi$ and $S_2 \not\models \Phi$ implies $S_1 \not\sim S_2$
- You even do not need to use an until-operator!
- To check $S \models \Phi$, it suffices to check $S/\sim \models \Phi$