Bisimulation and CTL*

Lecture #23 of Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

E-mail: katoen@cs.rwth-aachen.de

June 27, 2007

Overview Lecture #23

$\Rightarrow$ Repetition: Bisimulation equivalence

- CTL* equivalence
-

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:

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

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

   (c) 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

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

and

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

Bisimulation on paths

Whenever we have:

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

this can be completed to

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

proof: by induction on index i of state s_i

Bisimulation vs. trace equivalence

$$TS_1 \sim TS_2 \quad \text{implies} \quad \mathit{Traces}(TS_1) = \mathit{Traces}(TS_2)$$

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

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

$$TS/\!\sim_{TS} = (S', \{\tau\}, \rightarrow', I', AP, L'), \quad \text{the } \textit{quotient} \text{ of } TS \text{ under } \sim_{TS}$$

where

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

The Bakery algorithm

Process 1:

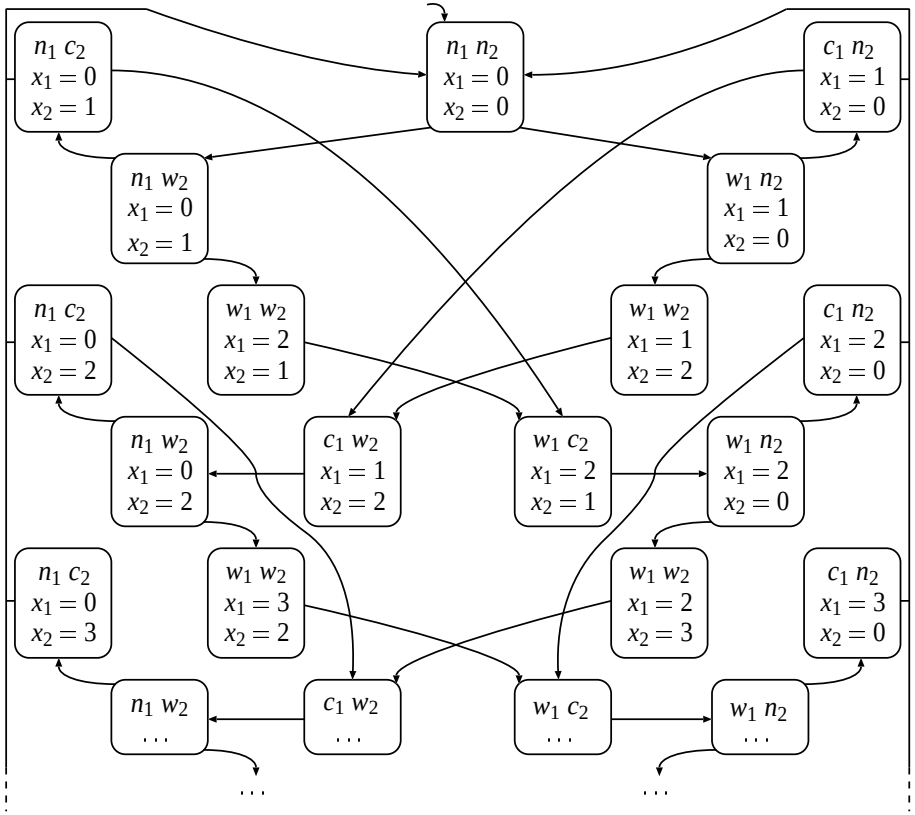
```
        ......
        while true {
            ......
n1 :        x1 := x2 + 1;
w1 :        wait until(x2 = 0 || x1 < x2 ) {
c1 :            ... critical section ...}
            x1 := 0;
            ......
        }
```

Process 2:

```
        ......
        while true {
            ......
n2 :        x2 := x1 + 1;
w2 :        wait until(x1 = 0 || x2 < x1) {
c2 :            ... critical section ...}
            x2 := 0;
            ......
        }
```

this algorithm can be applied to arbitrary many processes

Bakery algorithm transition system



infinite state space due to possible unbounded increase of counters

Data abstraction

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

Let $s = \langle \ell_1, \ell_2, x_1 = b_1, x_2 = b_2 \rangle$ be a state of TS_{Bak} with $\ell_i \in \{ n_i, w_i, c_i \}$ and $b_i \in \mathbb{N}$

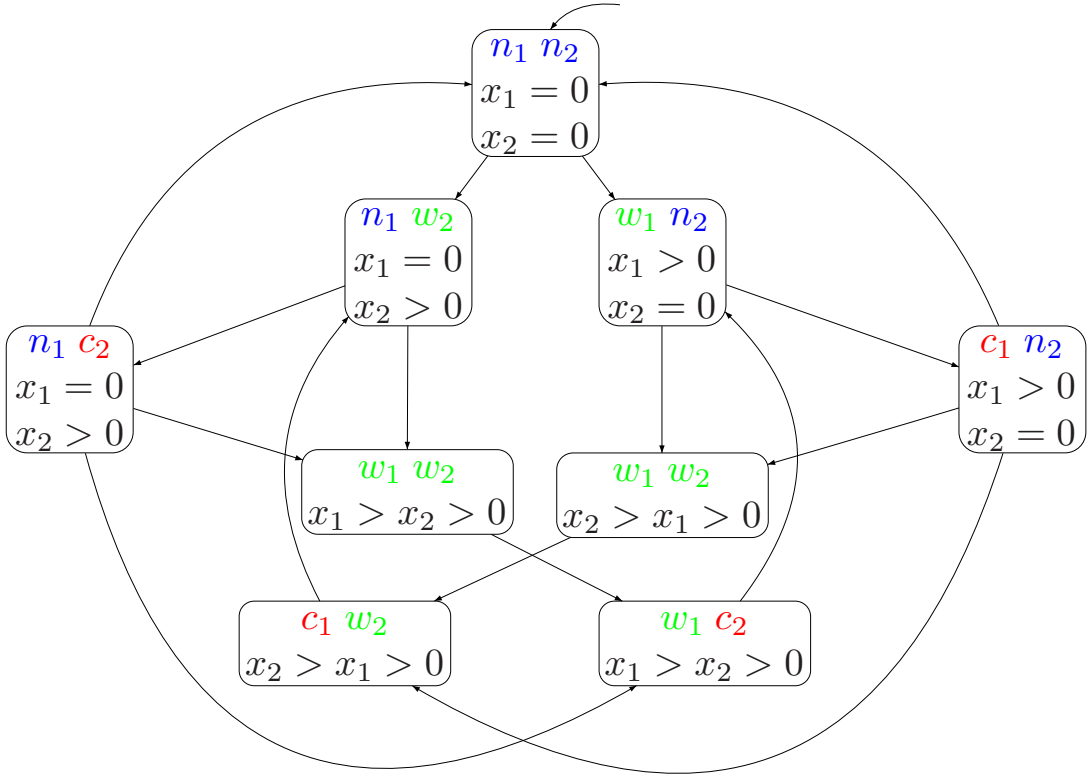
Then:

$$f(s) = \begin{cases} \langle \ell_1, \ell_2, x_1 = 0, x_2 = 0 \rangle & \text{if } b_1 = b_2 = 0 \\ \langle \ell_1, \ell_2, x_1 = 0, x_2 > 0 \rangle & \text{if } b_1 = 0 \text{ and } b_2 > 0 \\ \langle \ell_1, \ell_2, x_1 > 0, x_2 = 0 \rangle & \text{if } b_1 > 0 \text{ and } b_2 = 0 \\ \langle \ell_1, \ell_2, x_1 > x_2 > 0 \rangle & \text{if } b_1 > b_2 > 0 \\ \langle \ell_1, \ell_2, x_2 > x_1 > 0 \rangle & \text{if } b_2 > b_1 > 0 \end{cases}$$

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

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

Bisimulation quotient



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

Remarks

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

Overview Lecture #23

- Repetition: Bisimulation equivalence

$\Rightarrow$ CTL* equivalence

Syntax of CTL*

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

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

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

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

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

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

in CTL*: $\forall\varphi \;=\; \neg\exists\neg\varphi$. This does not hold in CTL!

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

Trace equivalence and LTL equivalence

Let *TS* be a *finite* transition system and s, s' states in *TS*

The following statements are equivalent:

(1) $\mathit{Traces}(s) \sim_{TS} \mathit{Traces}(s')$

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

Bisimulation vs. CTL* and CTL equivalence

Let *TS* be a *finite* transition system and s, s' states in *TS*

The following statements are equivalent:

(1) $s \sim_{TS} s'$

(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 $\bigcirc$

Proof: $\sim \subseteq \equiv_{CTL^*}$

Example master formula

Proof: $\equiv_{CTL} \subseteq \sim$

Bisimulation vs. CTL*-equivalence

For any transition systems TS and TS' (over AP):

$TS \sim TS'$ iff $TS \equiv_{CTL} TS'$ iff $TS \equiv_{CTL^*} TS'$

$\Rightarrow$ prior to model-check Φ, it is safe to first minimize TS wrt. $\sim$

this can be done with time complexity $\mathcal{O}(K \cdot \log N)$

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
  - $TS_1 \models \Phi$ and $TS_2 \not\models \Phi$ implies $TS_1 \not\sim TS_2$
- You even do not need to use an until-operator!
- To check $TS \models \Phi$, it suffices to check $TS/\!\sim \models \Phi$

Example