Verification

Lecture 14

Bernd Finkbeiner
Peter Faymonville
Michael Gerke

UNIVERSITÄT
DES
SAARLANDES

Coming up in two weeks...

Midterm Exam will take place
on Dec 20th, 4pm-7pm
Günter-Hotz-Hörsaal (building E2 2, formerly called Audimo)
Open Book

REVIEW: Bounded model checking

Search for counterexamples of bounded length

There exists a counterexample of length k to the invariant AG p
iff the following formula is satisfiable:

$$f_I(\vec{v}_0) \wedge f_{\rightarrow}(\vec{v}_0, \vec{v}_1) \wedge f_{\rightarrow}(\vec{v}_1, \vec{v}_2) \wedge \ldots f_{\rightarrow}(\vec{v}_{k-2}, \vec{v}_{k-1}) \wedge (\neg p_0 \vee \neg p_1 \vee \ldots \vee \neg p_{k-1})$$

REVIEW: Automata-based approach

Automata-based approach:

- Translate LTL formula $\neg\varphi$ to Büchi automaton
- Build product with transition system
- Encode all paths that start in initial state and are k steps long
- Require that path contains loop with accepting state

$$f_I(\vec{v}_0) \wedge \bigwedge_{i=0}^{k-2} f_{\rightarrow}(\vec{v}_i, \vec{v}_{i+1}) \wedge \bigvee_{i=0}^{k-1} \left((\vec{v}_i = \vec{v}_k) \wedge \bigvee_{j=i}^{k-1} f_F(\vec{v}_j) \right)$$

Formula size: $O(k \cdot |TS| \cdot 2^{|\varphi|})$

REVIEW: Fixpoint-based translation

$$\psi_{TS} \wedge \psi_{loop} \wedge [\psi]_0$$

- $\psi_{TS} = f_I(\vec{v}_0) \ \wedge \ \bigwedge_{i=0}^{k-2} f_{\rightarrow}(\vec{v}_i, \vec{v}_{i+1})$
- ψ_{loop}: loop constraint, ensures the existence of exactly one loop
- $[\varphi]_0$: fixpoint formula, ensures that LTL formula holds

Formula size: $O(k \cdot (|TS| + |\varphi|))$

REVIEW: The Completeness Threshold

The bound k is increased incrementally until

- a counterexample is found, or
- the problem becomes intractable due to the complexity of the SAT problem
- k reaches a precomputed threshold that guarantees that there is no counterexample

→ this threshold is called the completeness threshold CL.

The completeness threshold

- Computing *CL* is as hard as model checking
- Idea: Compute an overapproximation of *CL* based on the graph structure

Basic notions:

- Diameter D: Longest shortest path between any two reachable states
- Recurrence diameter RD: Longest loop-free path between any two reachable states
- Initialized diameter D^I: Longest shortest path between some initial state and some reachable state
- Initialized recurrence diameter RD^I: Longest loop-free path between some initial state and some reachable state

Completeness thresholds

- For $\Box p$ properties, $CT \leq D^l$.
- For $\Diamond p$ properties, $CT \leq RD^l + 1$.
- For general LTL properties, $CT \leq \min(RD^l + 1, D^l + D)$
  (where D, D^l, RD, RD^l refer to the product graph)

Complexity

- k chosen as $\min(RD^l + 1, D^l + D)$ is exponential in number of state variables
- Size of SAT instance is $O(k \cdot (|TS| + |\varphi|))$
- SAT is solved in exponential time

$\Rightarrow$ double exponential in number of state variables
(Compare: BDD-based model checking is single-exponential)

- In practice, bounded model checking is very successful
- Finds shallow errors fast
- In practice, RD, D are often not exponential

Implementation Relations

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:

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

⇒ 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)

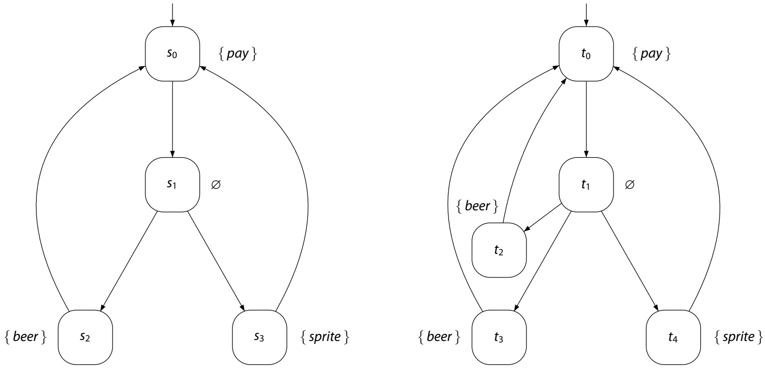
Bisimulation equivalence

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

and

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

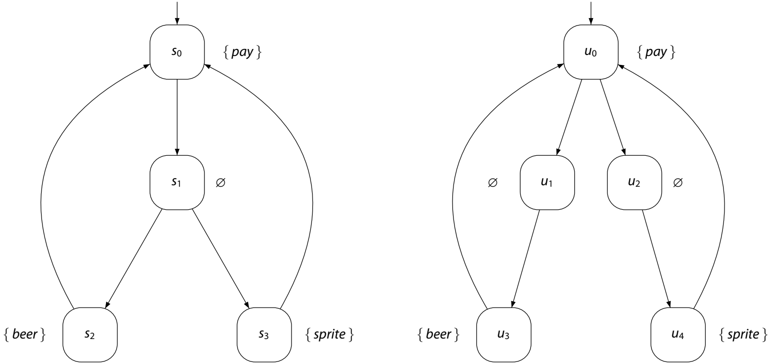
Example (1)



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

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:

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

this can be completed to

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

proof: by induction on index i of state s_i

Bisimulation vs. trace equivalence

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

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 $(q_1, q_2) \in \mathcal{R}$:

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

q_1 and q_2 are bisimilar, $q_1 \sim_{TS} q_2$, if $(q_1, q_2) \in \mathcal{R}$ for some bisimulation $\mathcal{R}$ for TS

$q_1 \sim_{TS} q_2$ if and only if $TS_{q_1} \sim TS_{q_2}$

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

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

The Bakery algorithm

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

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

Data abstraction

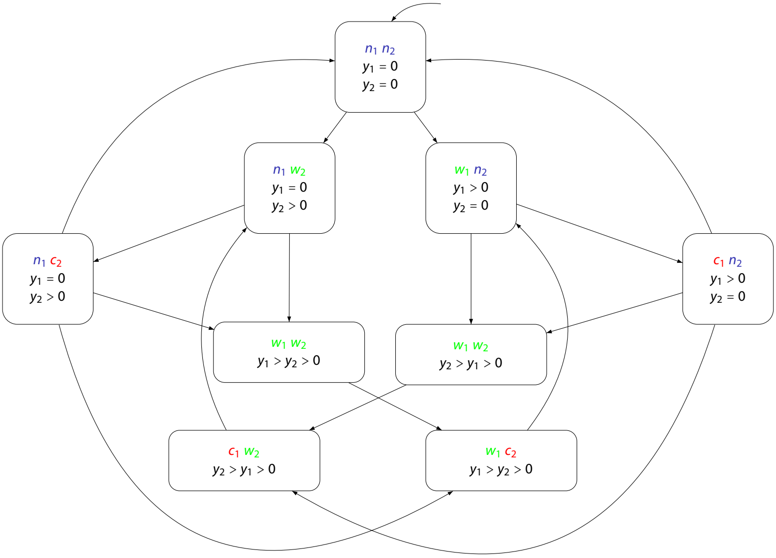
Function f maps a reachable state of TS_{Bak} onto an abstract one in TS_{Bak}^{abs}
Let $s = \langle \ell_1, \ell_2, y_1 = b_1, y_2 = b_2 \rangle$ be a state of TS_{Bak} with $\ell_i \in \{ n_i, w_i, c_i \}$ and $b_i \in \mathbb{N}$
Then:

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

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

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

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)
- $TS_{Bak}^{abs} \models \varphi$ with, e.g.,:
  - $\Box(\neg crit_1 \lor \neg crit_2)$ and
    $(\mathsf{G}\,\mathsf{F}\, wait_1 \Rightarrow \mathsf{G}\,\mathsf{F}\, crit_1) \land (\mathsf{G}\,\mathsf{F}\, wait_2 \Rightarrow \mathsf{G}\,\mathsf{F}\, crit_2)$
- Since $TS_{Bak}^{abs} \sim TS_{Bak}$, it follows $TS_{Bak} \models \varphi$
- Note: $Traces(TS_{Bak}^{abs}) = Traces(TS_{Bak})$

REVIEW: Syntax of CTL*

CTL* state-formulas are formed according to:

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

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

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

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

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 TS (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

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

<u>for any sublogic of CTL*, logical equivalence is defined analogously</u>

Bisimulation vs. CTL* and CTL equivalence

Let TS be a finite state graph 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 X

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 $TS/\sim \models \Phi$