Algorithms for Model Checking (2IW55)

Lecture 4

Symbolic Model Checking: Fairness and Counterexamples

Chapter 6.3, 6.4.

Tim Willemse
(timw@win.tue.nl)
http://www.win.tue.nl/∼timw
HG 6.81

Outline

1. Symbolic Model Checking
2. Fair Symbolic Model Checking
3. Counterexamples and Witnesses
   - Witnesses for E [U]
   - Witnesses for fair E G
4. Exercise

Symbolic Model Checking

In summary, symbolic model checking:

- Recursively processes subformulae
- Represent the set of states satisfying a subformula by OBDDs
- Treats temporal operators by fixed point computations
- Relies on efficient implementation of equivalence test, and $\wedge, \vee, \neg$ and $\exists$ connectives on OBDDs.

Symbolic Model Checking

Fix a Kripke Structure $\mathcal{M} = \langle \mathcal{S}, \mathcal{R}, \mathcal{L} \rangle$.

The temporal operators of CTL are characterised by fixed points:

- $\mathsf{E\ F}\ g = \mu Z. g \vee \mathsf{E\ X}\ Z$
- $\mathsf{E\ G}\ f = \nu Z. f \wedge \mathsf{E\ X}\ Z$
- $\mathsf{E}\ [f\ \mathsf{U}\ g] = \mu Z. g \vee (f \wedge \mathsf{E\ X}\ Z)$

- Least Fixed Points: start iteration at false ($\emptyset$)
- Greatest Fixed Points: start iteration at true ($\mathcal{S}$)

Intuition:

- Eventually .... least fixed points
- Globally .... greatest fixed points

Symbolic Model Checking

CTL model checking with Fixed Points

Function check(f) takes a formula f and returns the set of states where f holds: $\{s \mid s \models f\}$ (given a fixed Kripke Structure $\mathcal{M} = \langle \mathcal{S}, \mathcal{R}, \mathcal{L} \rangle$).

check(p)	$\{s \mid p \in \mathcal{L}(s)\}$
check($\neg f$)	$\mathcal{S} \setminus$ check(f)
check($f \vee g$)	check(f)$\cup$ check(g)
check(E X f)	$\mathcal{P}re_{\mathcal{R}}$(check($f$)
check(E [f U g])	lfp$\big(\mathcal{Z} \mapsto$ check(g) $\cup$ (check(f) $\cap$ $\mathcal{P}re_{\mathcal{R}}(\mathcal{Z})$))$\big)$
check(E G f)	gfp$\big(\mathcal{Z} \mapsto$ check(f) $\cap$ $\mathcal{P}re_{\mathcal{R}}(\mathcal{Z})\big)$

Recall: $\mathcal{P}re_{\mathcal{R}}(\mathcal{Z}) = \{s \in \mathcal{S} \mid \exists t \in \mathcal{Z}. s \ \mathcal{R} \ t\}$

Outline

Fair Symbolic Model Checking

Fix a fair Kripke Structure $\mathcal{M} = \langle \mathcal{S}, \mathcal{R}, \mathcal{L}, \{\mathcal{F}_1, \ldots, \mathcal{F}_n\} \rangle$

Recall that a fair path infinitely often hits some state from each fairness constraint $\mathcal{F}_i$

- First, note that in fair CTL (with $\models_{\mathcal{F}}$),

$$\mathsf{E\,G}\,f \equiv f \wedge \bigwedge_{k=1}^{n} \mathsf{E\,X\,E}\,[f \;\mathsf{U}\; (\mathcal{F}_k \wedge \mathsf{E\,G}\,f)] \qquad \text{(prove } \subseteq \text{ and } \supseteq\text{)}$$

- Next, if

$$\mathcal{Z} \equiv f \wedge \bigwedge_{k=1}^{n} \mathsf{E\,X\,E}\,[f \;\mathsf{U}\; (\mathcal{F}_k \wedge \mathcal{Z})]$$

  Then $\mathcal{Z} \subseteq \mathsf{E\,G}\,f$ (construct a path cycling through $\mathcal{F}_1, \ldots, \mathcal{F}_n$)

- Hence, we found:

$$\mathsf{E\,G}\,f \equiv \nu \mathcal{Z}. f \wedge \bigwedge_{k=1}^{n} \mathsf{E\,X\,E}\,[f \;\mathsf{U}\; (\mathcal{F}_k \wedge \mathcal{Z})]$$

Fair Symbolic Model Checking

The equivalence

$$\mathsf{E\,G}\ f \equiv \nu Z. f \wedge \bigwedge_{k=1}^{n} \mathsf{E\,X\,E}\,[f\ \mathsf{U}\ (\mathcal{F}_k \wedge Z)]$$

leads to the following algorithm:

$\text{check}_{\mathcal{F}}(\mathsf{E\,G}\ f)$	$\text{gfp}\big(Z \mapsto \text{check}(f \wedge \bigwedge_{k=1}^{n} \mathsf{E\,X}\ (\mathsf{E}\,[f\ \mathsf{U}\ (\mathcal{F}_k \wedge Z)]))\big)$

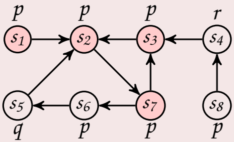
So, in the greatest fixed point computation for E G , we perform nested least fixed point computations to compute E [U].

Next, we can compute an OBDD $\mathit{fair} := \text{check}_{\mathcal{F}}(\mathsf{E\,G}\ \text{true})$. The remaining temporal operators can then be encoded as follows:

$\text{check}_{\mathcal{F}}(\mathsf{E\,X}\ f)$	$\text{check}(\mathsf{E\,X}\ (f \wedge \mathit{fair}))$
$\text{check}_{\mathcal{F}}(\mathsf{E}\,[f\ \mathsf{U}\ g])$	$\text{check}(\mathsf{E}\,[f\ \mathsf{U}\ (g \wedge \mathit{fair})])$

Fair Symbolic Model Checking

Example



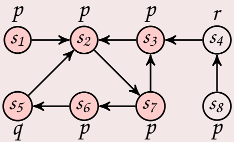
- To check: E G p
- Fairness constraint: $\neg r$
- Compute: $\nu \mathcal{Z}.\mathsf{check}(p \wedge \mathsf{E\ X}\ (\mathsf{E}\ [p\ \mathsf{U}\ (\neg r \wedge \mathcal{Z})]))$
- Set
  $\phi(\mathcal{Z}) = \mathsf{lfp}(\mathcal{Y} \mapsto (\mathsf{check}(\neg r) \cap \mathcal{Z}) \cup (\mathsf{check}(p) \cap \mathsf{pre}_{\mathcal{R}}(\mathcal{Y})))$

$$
\begin{aligned}
\mathcal{Z}_0 &= \mathcal{S} \\
\mathcal{Z}_1 &= \mathsf{check}(p) \cap \mathsf{pre}_{\mathcal{R}}(\phi(\mathcal{S})) = \{s_1, s_2, s_3, s_6, s_7\} \\
\mathcal{Z}_2 &= \mathsf{check}(p) \cap \mathsf{pre}_{\mathcal{R}}(\{s_1, s_2, s_3, s_6, s_7\}) \\
&= \{s_1, s_2, s_3, s_7\} \\
\mathcal{Z}_3 &= \mathsf{check}(p) \cap \mathsf{pre}_{\mathcal{R}}(\{s_1, s_2, s_3, s_7\}) \\
&= \{s_1, s_2, s_3, s_7\}
\end{aligned}
$$

$\mathcal{Z}_2 = \mathcal{Z}_3$, so this is the greatest fixed point.

Fair Symbolic Model Checking

Example



- To check: $\mathsf{E}\ [p\ \mathsf{U}\ q]$
- Fairness constraint: $\neg r$
- Compute $\mathit{fair} := \mathrm{check}_{\mathcal{F}}(\mathsf{E\ G\ true})$ $(= \mathcal{S})$
- Compute: $\mu Z.(q \wedge \mathit{fair}) \vee (p \wedge \mathsf{E\ X}\ Z)$ (with lfp)

$$
\begin{aligned}
Z_0 &= \text{false} = \emptyset \\
Z_1 &= q \vee (p \wedge \mathsf{E\ X}\ Z_0) = \{s_5\} \\
Z_2 &= q \vee (p \wedge \mathsf{E\ X}\ Z_1) = \{s_5, s_6\} \\
Z_3 &= q \vee (p \wedge \mathsf{E\ X}\ Z_2) = \{s_5, s_6, s_7\} \\
Z_4 &= q \vee (p \wedge \mathsf{E\ X}\ Z_3) = \{s_2, s_5, s_6, s_7\} \\
Z_5 &= q \vee (p \wedge \mathsf{E\ X}\ Z_4) = \{s_1, s_2, s_3, s_5, s_6, s_7\} \\
Z_6 &= q \vee (p \wedge \mathsf{E\ X}\ Z_5) = \{s_1, s_2, s_3, s_5, s_6, s_7\}
\end{aligned}
$$

$Z_5 = Z_6$, so this is the least fixed point.

Outline

1. Symbolic Model Checking
2. Fair Symbolic Model Checking
3. Counterexamples and Witnesses
   - Witnesses for E [U]
   - Witnesses for fair E G
4. Exercise

Counterexamples and Witnesses

- Motivation:
  - In practice, a model checker is often used as an extended debugger
  - If a bug is found, the model checker should provide a particular trace, which shows it
- A formula with a universal path quantifier has a counterexample consisting of one trace
- A formula with an existential path quantifier has a witness consisting of one trace
- Due to the dualities in CTL, we only have to consider:
  - a finite trace witnessing $\mathsf{E}\ [f\ \mathsf{U}\ g]$
  - an infinite trace witnessing $\mathsf{E}\ \mathsf{G}\ f$; for finite systems, the latter is a so-called lasso, consisting of a prefix and a loop
- For fair counter examples we require that the loop contains a state from each fairness constraint

Counterexamples and Witnesses – Witnesses for E [U]

- $\mathsf{E}\,[f \;\mathsf{U}\; g] = \mu Z.\; g \vee (f \wedge \mathsf{E\,X}\; Z)$
- Unfolding the recursion, we get:

$$\begin{array}{lcl} Z_0 & = & \text{false} \\ Z_1 & = & g \\ Z_2 & = & g \vee (f \wedge \mathsf{E\,X}\; g) \\ Z_3 & = & g \vee (f \wedge \mathsf{E\,X}\; (g \vee (f \wedge \mathsf{E\,X}\; g))) \end{array}$$

- So, the fixed point computation corresponds to a backward reachability analysis
- Z_i contains those states that can reach g in at most $i-1$ steps (and f holds in between).
- Assume $s_0 \models \mathsf{E}\,[f \;\mathsf{U}\; g]$. To find a minimal witness from state s_0, we start in the smallest $\mathcal{N}$ such that $s_0 \in Z_{\mathcal{N}}$.
- For $i \in 1, \ldots, \mathcal{N}-1$, we define s_i to be a state in $Z_{\mathcal{N}-i}$ satisfying $s_{i-1}\; \mathcal{R}\; s_i$.

Counterexamples and Witnesses – Witnesses for fair E G

- We want an initial path to a cycle on which each fairness constraint $\{\mathcal{F}_1, \ldots, \mathcal{F}_n\}$ occurs (i.e. the cycle must contain at least one state from all $\mathcal{F}_i$).
- $\mathsf{E\ G}\ f = \nu Z. f \wedge \bigwedge_{k=1}^{n} \mathsf{E\ X\ \ E}\,[f\ \mathsf{U}\ (\mathcal{F}_k \wedge Z)]$
- Unfolding the recursion, we get:

$$\begin{array}{lcl} Z_0 & = & \text{true} \\ & \ldots & \\ Z_L & = & f \wedge \bigwedge_{k=1}^{n} \mathsf{E\ X\ \ E}\,[f\ \mathsf{U}\ (\mathcal{F}_k \wedge Z_{L-1})] \end{array}$$

- Let $Z := Z_L = Z_{L-1} = \mathsf{E\ G}\ f$ be the fixed point
- To compute Z, we compute for each k $(1 \leq k \leq n)$, $\mathsf{E}\ [f\ \mathsf{U}\ (\mathcal{F}_k \wedge Z)]$ using backward reachability. So, we have for each k the approximations: $Q_0^k \subseteq Q_1^k \subseteq Q_2^k \subseteq \ldots \subseteq Q_{j_k}^k$
- From the E [U] case, recall that Q_i^k contains those states that can reach $\mathcal{F}_k \wedge Z$ in at most i steps

Counterexamples and Witnesses – Witnesses for fair E G

- Assume $s_0 \models_{\mathcal{F}} \mathsf{E\,G}\, f$, hence, $s_0 \in \mathcal{Z}$
- We will now inductively construct a path $s_0 \to^* s_1 \to^* \ldots \to^* s_n$, such that:
  - f holds along the whole path
  - $s_k \in \mathcal{Z} \wedge \mathcal{F}_k$ (for $1 \leq k \leq n$)
- Observe: by induction $s_{k-1} \models \mathcal{Z}$, so, by definition of $\mathcal{Z}$: $s_{k-1} \models \mathsf{E\,X\ \ E}\,[f\ \mathsf{U}\ (\mathcal{Z} \wedge \mathcal{F}_k)]$
- For $1 \leq k \leq n$ do:
  1. Determine the minimal $\mathcal{M}$ such that s_{k-1} has a successor $t_0^k \in Q_{\mathcal{M}}^k$.
  2. Construct (as the witness for E [U]):

     $$s_{k-1} \to t_0^k \to \cdots \to t_{\mathcal{M}}^k \in \mathcal{Z} \wedge \mathcal{F}_k$$

  3. Define $s_k := t_{\mathcal{M}}^k$.
- heuristic improvement: Visit the $\mathcal{F}_k$ in a different order: continue with the closest $\mathcal{F}_k$ that has not yet been visited.

Counterexamples and Witnesses – Witnesses for fair E G

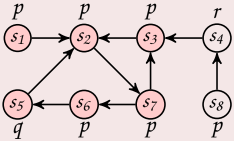
- Finally, we must close the loop, but this is not always possible: Check if $s_n \models \mathsf{E}\,\mathsf{X}\;\; \mathsf{E}\,[f \;\mathsf{U}\; \{s_1\}]$.
- If so: the E [U]-witness closes the loop
- If not: the cycle cannot be closed. Hence:
  - The sequence so far $s_0 \to \cdots \to s_n$ is in the prefix of the lasso, not yet on the loop.
  - Restart the whole procedure of the previous slide, now starting in $s_n \in \mathcal{Z}$.
- Eventually, this process must terminate:
  - We only restart if s_n cannot reach s_1
  - so we moved to the next Strongly Connected Component
  - The SCC graph cannot contain cycles
- Optimisation: By precomputing $\mathsf{E}\,[f \;\mathsf{U}\; \{s_1\}]$, one can detect earlier that closing the cycle will not be possible.

Outline

1. Symbolic Model Checking
2. Fair Symbolic Model Checking
3. Counterexamples and Witnesses
   - Witnesses for E [U]
   - Witnesses for fair E G
4. Exercise

Exercise

Example



- Check that $s_1 \models_{\mathcal{F}} \mathsf{E}\,\mathsf{G}\,(p \vee q)$
- Fairness constraint: $\neg r$ and q
- Construct a witness for $s_1 \models_{\mathcal{F}} \mathsf{E}\,\mathsf{G}\,(p \vee q)$