

Algorithms for Model Checking (2IW55)

Lecture 5

Equivalences and Pre-orders:
State Space Reduction and Preservation of Properties
Chapter 11, 11.1

Tim Willemse

(timw@win.tue.nl)

<http://www.win.tue.nl/~timw>

HG 6.81

Equivalences

Pre-orders

Bisimulation Reduction

Summarising

Complexity of model checking arises from:

- ▶ **State space explosion**: the state space is usually much larger than the specification
- ▶ **Expressive logics** have complex model checking algorithms

Ways to deal with the state space explosion:

- ▶ **equivalence reduction**: remove states with identical potentials from a state space
- ▶ **on-the-fly**: integrate the generation and verification phases, to prune the state space
- ▶ **symbolic model checking**: represent sets of states by clever data structures
- ▶ **partial-order reduction**: ignore some executions, because they are covered by others
- ▶ **abstraction**: remove details by working on conservative over-approximation

- ▶ A **state space reduction** reduces model checking complexity.
- ▶ Of course, the reduced state space must **preserve** (an interesting class of) temporal properties.
- ▶ This is often characterised by an **equivalence relation** on Kripke Structures:
 - reduction must yield an ‘equivalent’ model.
 - “equivalent” models must satisfy the same properties.
- ▶ Different instances of this scheme:
 - trace equivalence preserves LTL formulae.
 - **strong bisimulation** preserves **CTL*** (and **μ -calculus**) formulae.
 - **simulation** preserves **ACTL*** (and **universal μ -calculus**) formulae.
 - branching bisimulation preserves CTL*-X formulae.

Let two Kripke Structures over AP be given:

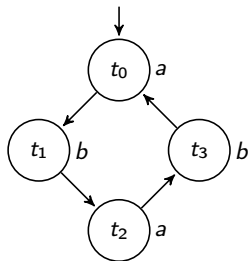
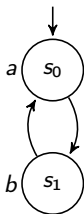
- ▶ $M = \langle S, R, S_0, L \rangle$ and
- ▶ $M' = \langle S', R', S'_0, L' \rangle$

Definition (Strong Bisimulation)

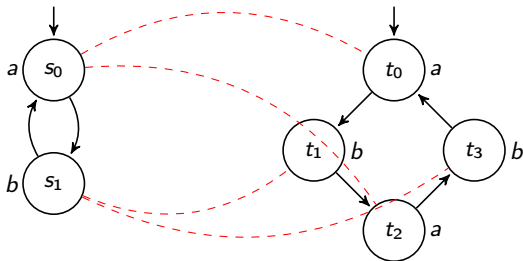
A relation $B \subseteq S \times S'$ is a **strong bisimulation relation** (also *zig-zag* relation) iff for every $s \in S$ and $s' \in S'$ with sBs' :

- ▶ $L(s) = L'(s')$
- ▶ for all $s_1 \in S$, if sRs_1 , then there exists $s'_1 \in S'$ such that $s'R's'_1$ and $s_1Bs'_1$
- ▶ for all $s'_1 \in S'$, if $s'R's'_1$, then there exists $s_1 \in S$ such that sRs_1 and $s_1Bs'_1$

Example

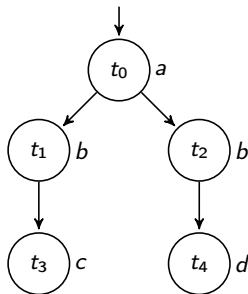
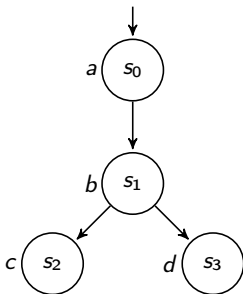


Example

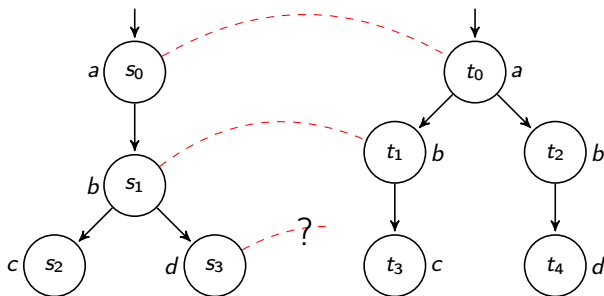


- ▶ unwinding and duplication preserves bisimulation
- ▶ Sensitive to the moment of choice

Example

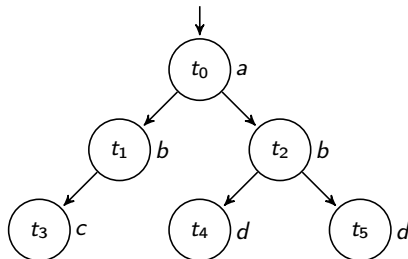
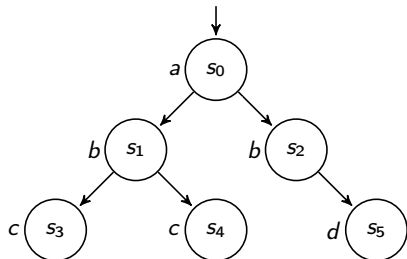


Example

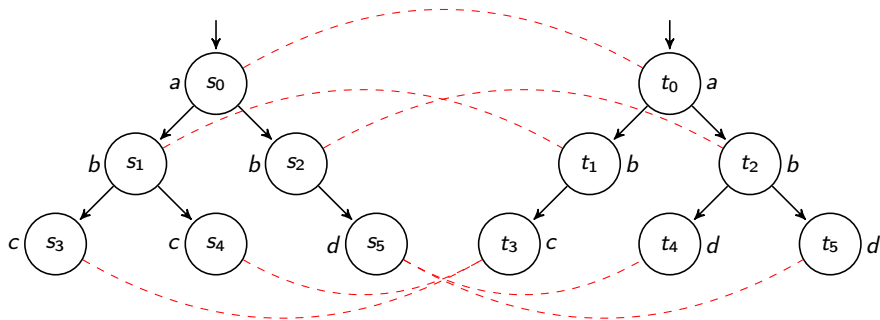


- ▶ unwinding and duplication preserves bisimulation
- ▶ Sensitive to the moment of choice

Example



Example



- ▶ unwinding and duplication preserves bisimulation
- ▶ Sensitive to the moment of choice

Let two Kripke Structures over AP be given:

- ▶ $M = \langle S, R, S_0, L \rangle$ and
- ▶ $M' = \langle S', R', S'_0, L' \rangle$

Definition (bisimilarity)

Two states $s \in S$ and $s' \in S'$ are **bisimilar**, if for some bisimulation relation B , sBs' . The Kripke Structures M and M' are bisimilar (notation: $M \equiv M'$) iff there exists a bisimulation relation B , “containing initial states”, i.e.:

- ▶ $\forall s_0 \in S_0 \exists s'_0 \in S'_0 : s_0 B s'_0$
- ▶ $\forall s'_0 \in S'_0 \exists s_0 \in S_0 : s_0 B s'_0$

Note:

- ▶ bisimilarity is an equivalence relation
- ▶ the union of bisimulation relations is again a bisimulation relation
- ▶ “bisimilarity” itself is the greatest bisimulation relation

Strong bisimulation preserves CTL*:

- ▶ Recall the CTL* semantics:
 - $M, s \models f$: state formula f holds in state s ,
 - $M, \pi \models f$: path formula f holds along path π .
- ▶ Recall that $M \models f$ iff for all $s_0 \in S_0$, $M, s_0 \models f$.

Theorem (14)

If $M \equiv M'$ (i.e. M and M' are bisimilar), then for every CTL* state formula f :

$$M \models f \quad \text{iff} \quad M' \models f$$

Practical consequence: In order to check $M \models f$, it is safe and sufficient to:

1. Reduce M to M' modulo bisimilarity,
2. Check whether $M' \models f$.

Proof sketch:

Given a relation B , we define that path π **corresponds** to path π' iff: $\forall i. \pi(i) B \pi'(i)$

Lemma (31)

If B is a bisimulation relation and $s B s'$ (*correction to Lemma 31*), then for every $\pi \in \text{path}(s)$ there exists a corresponding path $\pi' \in \text{path}(s')$ (*and vice versa*).

Next, with structural induction on CTL* formula f one can show: if s and s' are bisimilar and π and π' correspond, then:

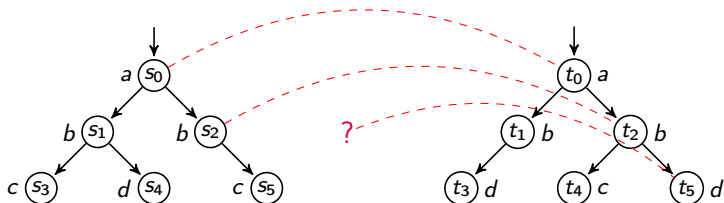
1. $s \models f$ if and only if $s' \models f$
2. $\pi \models f$ if and only if $\pi' \models f$

From this, the theorem follows:

for all M, M' and CTL* formulae f : **if $M \equiv M'$ then $M \models f$ iff $M' \models f$.**

Theorem (reverse)

If $M \not\equiv M'$ then there exists a formula f in **CTL**, such that $M \models f$ and $M' \not\models f$.



- ▶ Note that both systems have the same paths.
- ▶ There is no bisimulation relation between these two systems containing the initial states.
- ▶ Indeed, the following **CTL** formula holds in (the initial state of) the right system, but not on the left: **$A X (b \wedge E X d)$**
- ▶ We will see later that using **E** is essential.

Equivalences

Pre-orders

Bisimulation Reduction

Summarising

- ▶ bisimilar models have **the same behaviour**, so they make true exactly the same properties.
- ▶ **Idea**: If we allow to really **forget** information, we may:
 - reduce the state space further, but:
 - preserve only a smaller class of formulae.
- ▶ We say that system M' **simulates** system M if M' has **at least** the behaviour of M .

Let two Kripke Structures be given:

- ▶ $M = \langle AP, S, R, S_0, L \rangle$ and
- ▶ $M' = \langle AP', S', R', S'_0, L' \rangle$, with $AP' \subseteq AP$.

Definition (Simulation Relation)

A relation $H \subseteq S \times S'$ is a **simulation relation** iff for every $s \in S$ and $s' \in S'$ with $s H s'$:

- ▶ $L(s) \cap AP' = L'(s')$
- ▶ for all s_1 , if $s R s_1$, then there exists s'_1 such that $s' R' s'_1$ and $s_1 H s'_1$.

Definition (Simulation)

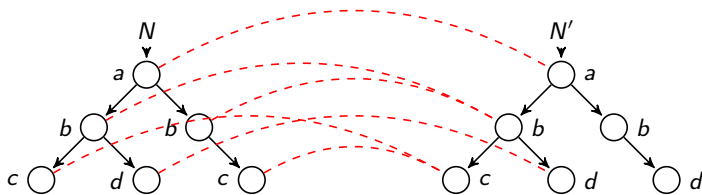
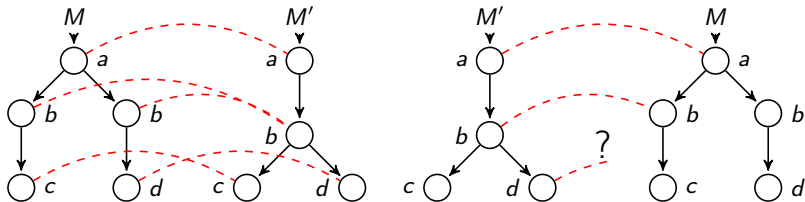
M' **simulates** M (written: $M \sqsubseteq M'$) iff there exists a simulation relation H , such that

$$\forall s_0 \in S_0. \exists s'_0 \in S'_0. s_0 H s'_0$$

This defines an equivalence relation as follows: $M \sim M'$ iff $M \sqsubseteq M'$ and $M' \sqsubseteq M$.

Note:

- ▶ \sqsubseteq is a **pre-order** on Kripke Structures (i.e. it is reflexive and transitive, but not necessarily symmetric).
- ▶ **Warning:**
 - it is possible that $M \sim M'$ but still $M \not\cong M'$
 - In words: if two systems simulate each other, they need not be bisimilar.
 - Intuitively: the two simulations may use a different H , while a bisimulation requires **one** B .



- ▶ $M \sqsubseteq M'$ but not $M' \sqsubseteq M$;
- ▶ $N \sim N'$ but $N \not\cong N'$.

Definition (ACTL*)

ACTL* (see p.31) is the fragment of CTL* with only universal path quantifiers, no existential path quantifiers.

Note:

- ▶ This only makes sense for formulae in **positive normal form**, i.e. negations only occur directly in front of atomic propositions.
- ▶ Examples: $A F G p$, $A G (p \rightarrow A X q)$ are in ACTL*, but $A G (p \rightarrow E X q)$ is not.
Careful: $(A G p) \rightarrow (A G q)$ is not in ACTL*, because actually:

$$\begin{aligned} (A G p) \rightarrow (A G q) &\equiv \neg(A G p) \vee (A G q) \\ &\equiv (E F \neg p) \vee (A G q) \end{aligned}$$

Simulation preserves ACTL*:

Theorem

If $M \sqsubseteq M'$ (i.e. M' simulates M), then for every ACTL* state formula f over AP' :

$$\text{if } M' \models f \quad \text{then} \quad M \models f$$

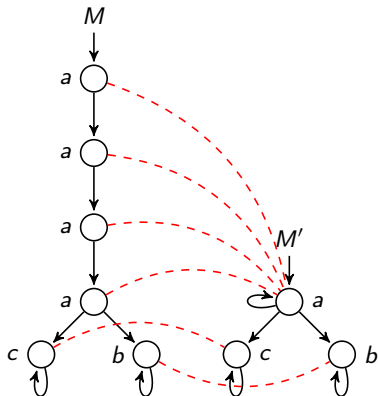
Practical consequence: In order to check $M \models f$, it is safe to find an approximation M' with $M \sqsubseteq M'$ and check that $M' \models f$.

However: if $M' \not\models f$, we obtain **no information** about $M \models f$ — it may or may not hold.

In the previous example, we had: $N \sim N'$ but $N \not\equiv N'$. Hence:

- ▶ N and N' satisfy the same ACTL* formulae
- ▶ N and N' do not satisfy the same CTL formulae
- ▶ They can only be distinguished using operator **E**.

Example



- ▶ Observe that $M \sqsubseteq M'$ with H indicated left.
- ▶ Note that $M' \models \text{A G } (a \vee b \vee c)$ and hence $M \models \text{A G } (a \vee b \vee c)$.
- ▶ Note that $M' \not\models \text{A F } (b \vee c)$, but actually $M \models \text{A F } (b \vee c)$. This shows that **some information is really lost**.
- ▶ Note: $M \models \text{A X } a$ but $M' \not\models \text{A X } a$.
Conclusion: $M' \not\sqsubseteq M$.
- ▶ Note: $M' \models \text{E X } b$, but $M \not\models \text{E X } b$ (**not in ACTL***, is in ECTL*).

Equivalences

Pre-orders

Bisimulation Reduction

Summarising

Computing Bisimulation Equivalence:

Let two Kripke Structures be given:

- ▶ $M = \langle AP, S, R, S_0, L \rangle$ and
- ▶ $M' = \langle AP, S', R, S'_0, L' \rangle$.

Define a **sequence of relations** $s B_i^* s'$ iff s and s' cannot be distinguished within i steps:

- ▶ $s B_0^* s'$ if and only if $L(s) = L'(s)$.
- ▶ $s B_{n+1}^* s'$ if and only if:
 1. $s B_n^* s'$, and
 2. $\forall s_1$ with $R(s, s_1), \exists s'_1$ with $s' R' s'_1$ and $s_1 B_n^* s'_1$.
 3. $\forall s'_1$ with $R'(s', s'_1), \exists s_1$ with $s R s_1$ and $s_1 B_n^* s'_1$.
- ▶ Let $B^* := \bigcap_i B_i^*$

Clearly, $B_i^* \supseteq B_{i+1}^*$, so B^* can be computed by fixed point iteration.

Actually, this can be implemented symbolically by OBDDs

- ▶ **Observe:** B^* is the largest bisimulation between M and M' .
- ▶ So: if s and s' are bisimilar, then $s B^* s'$.
- ▶ To test if $M \equiv M'$: check if for each $s_0 \in S_0$ there exists an $s'_0 \in S'_0$ such that $s_0 B^* s'_0$.
- ▶ By carefully splitting equivalence classes, the procedure can run in $\mathcal{O}(|R| \times \log(|S|))$ time (Paige-Tarjan).
- ▶ Similar ideas apply to checking $M \sqsubseteq M'$.

The algorithm can be modified for state space reduction as follows:

- ▶ The equivalence classes of B^* form the states of the reduced state space (minimal modulo bisimulation).
- ▶ The transitions between two classes are derived from the transitions between elements of these classes.

Equivalences

Pre-orders

Bisimulation Reduction

Summarising

- ▶ Bisimulation is an equivalence relation.
- ▶ Bisimulation preserves CTL* formulae.
- ▶ Simulation is a pre-order.
- ▶ Simulation preserves ACTL* formulae only, and only in one direction.
- ▶ Simulation allows for more reduction but sometimes crucial information is lost.
- ▶ Bisimulation and Simulation reduction can be computed in polynomial time.

Possible improvement: Instead of:

1. generating state space
2. reducing state space
3. model checking reduced state space,

it would be better to generate a smaller state space immediately.