

Algorithms for Model Checking 2IW50

Lecture 6: Bisimulation:
State Space Reduction and
Preservation of Properties

Jaco van de Pol

`vdpol@cwi.nl`

`http://www.cwi.nl/~vdpol/amc.html`

Technische Universiteit Eindhoven
Centrum voor Wiskunde en Informatica

Motivation

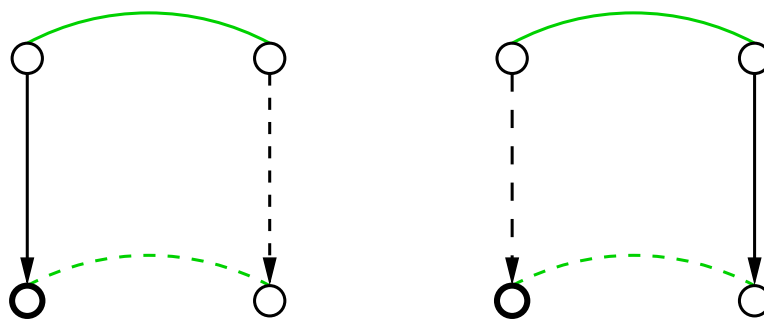
- So far, we have looked at Model Checking algorithms for a **fixed** Kripke Structure
- We will now consider techniques for **state space reduction**, to be applied before model checking
- Of course, the reduced state space must **preserve** (an interesting class of) temporal properties.
- This is often characterized by an **equivalence relation** on Kripke Structures:
 1. reduction must yield an “equivalent” model
 2. “equivalent” models must satisfy the same properties
- Different instances of this scheme:
 - (Trace equivalence preserves LTL formulas)
 - **Bisimulation** preserves **CTL*** formulas
 - **Simulation** preserves **ACTL*** formulas
 - (Branching bisimulation preserves CTL*-X)

O V E R V I E W

1. Definition of Bisimulation
2. Preservation of CTL* formulas
3. Definition of Simulation
4. Preservation of ACTL* formulas
5. Bisimulation reduction algorithm
6. Notes on Paige-Tarjan's efficient version

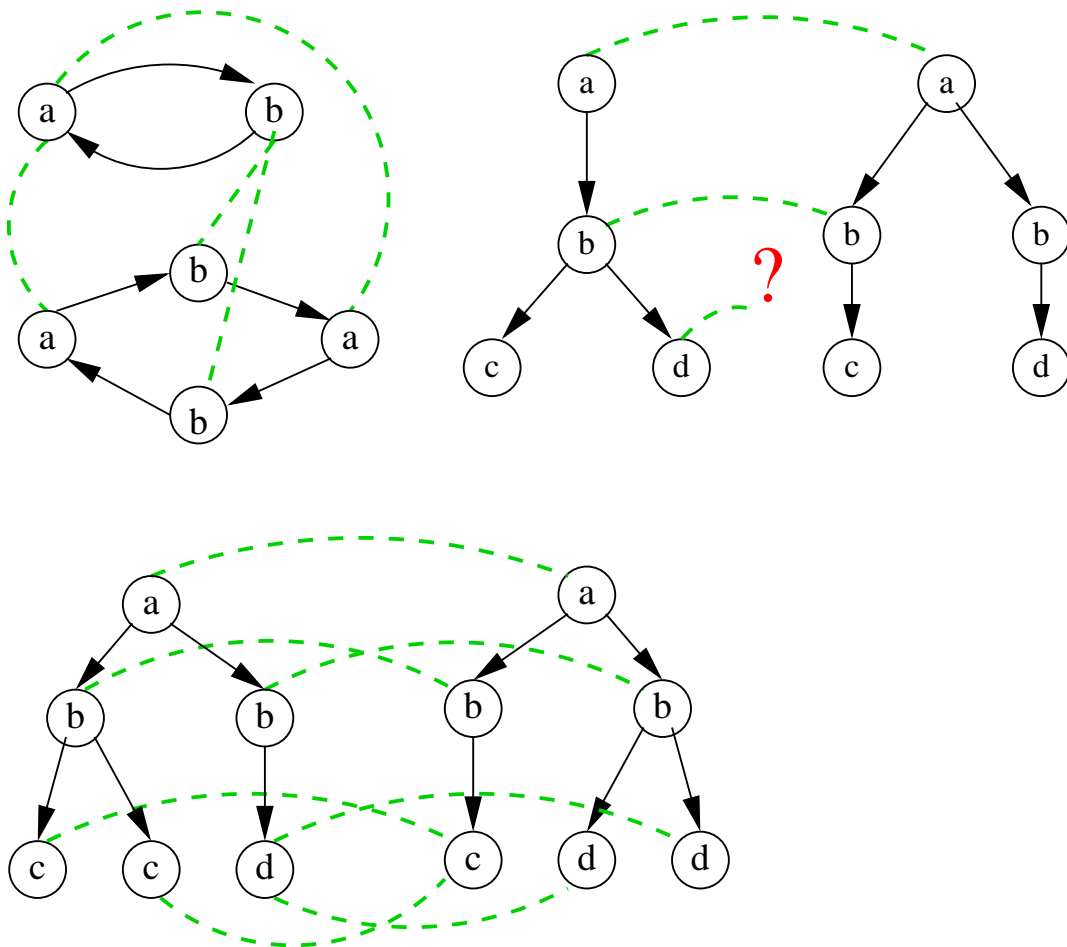
Bisimulation Equivalence: definition

- Let two Kripke Structures be given:
 $M = (AP, S, R, S_0, L)$ and
 $M' = (AP, S', R', S'_0, L')$.
(AP is in the structure, also initial states S_0)
- A relation $B \subseteq S \times S'$ is a **bisimulation relation** iff for every $s \in S$ and $s' \in S'$ with $B(s, s')$:
 1. $L(s) = L'(s')$
 2. for all s_1 , if $R(s, s_1)$, then there exists s'_1 such that $R'(s', s'_1)$ and $B(s_1, s'_1)$.
 3. for all s'_1 , if $R'(s', s'_1)$, then there exists s_1 such that $R(s, s_1)$ and $B(s_1, s'_1)$.
- (2) and (3) in a picture:



Bisimulation Equivalence: examples

- The concrete bisimulation relations for Figure 11.1 (**corrected!**) and Fig 11.2
- Unwinding and duplication preserves bisimulation.
- Bisimulation is sensitive to the moment of choice. (Fig 11.3)



Bisimulation Equivalence: defs (2)

- Let two Kripke Structures be given:
 $M = (AP, S, R, S_0, L)$ and
 $M' = (AP, S', R', S'_0, L')$.
- Two states $s \in S$ and $s' \in S'$ are **bisimilar**, if for some bisimulation relation B , $B(s, s')$.
- The Kripke Structures M and M' are bisimilar ($M \equiv M'$) iff there exists a bisimulation relation B , “containing initial states”, that is, such that both:
 - $\forall s_0 \in S_0, \exists s'_0 \in S'_0$ such that $B(s_0, s'_0)$
 - $\forall s'_0 \in S'_0, \exists s_0 \in S_0$ such that $B(s_0, s'_0)$
- Additional notes:
 - bisimilarity is an equivalence relation
 - the union of bisimulation relations is again a bisimulation relation
 - “bisimilarity” itself is the greatest bisimulation relation

Bisimulation and CTL* formulas

- 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
- We now define:

$$M \models f: \text{forall } s_0 \in S_0, M, s_0 \models f.$$

- **Theorem:** If $M \equiv M'$ (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

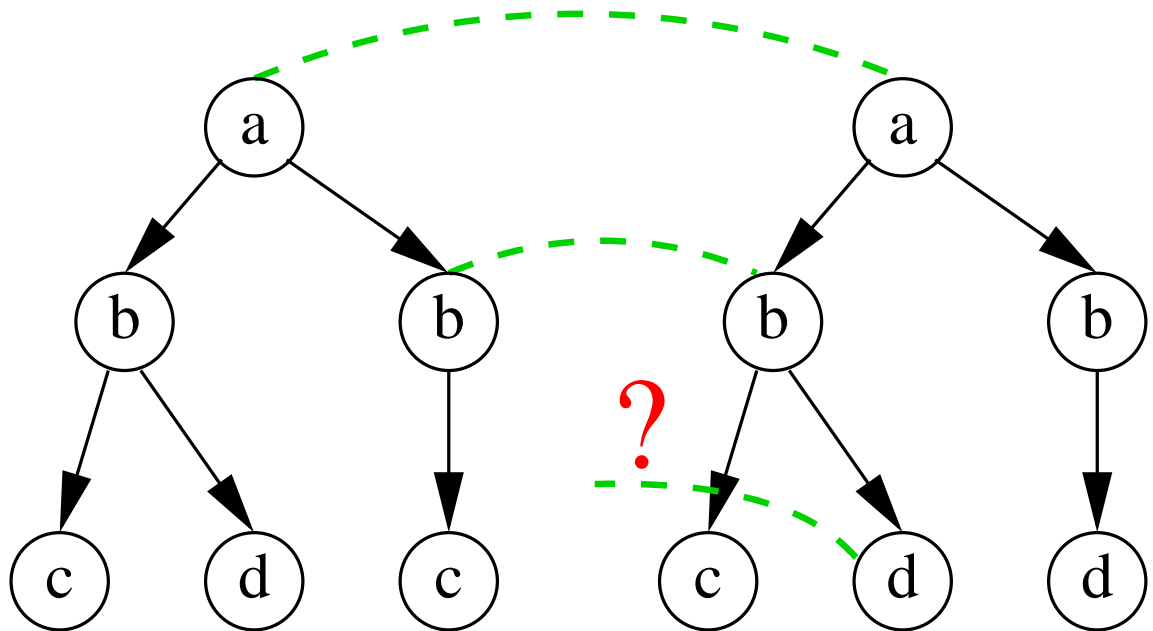
1. Reduce M to M' modulo bisimilarity
2. Check whether $M' \models f$.

- Actually, the following reverse also holds:

Theorem: If $M \not\equiv M'$, then there exists a formula f in CTL (!), such that

$$M \models f \text{ and } M' \not\models f.$$

Example of non-bisimulation



- Note that both systems have the same paths
- There is no bisimulation relation between these two systems that contain their roots.
- Indeed, the following CTL formula holds in (the root of) the right system, but not on the left.

$$\mathbf{AX} (b \wedge \mathbf{EX} d)$$

- We will see later that using \mathbf{E} is essential.

Proof Sketch

- Given a bisimulation relation B , we define that path π **corresponds** to path π' iff

$$\forall i. B(\pi[i], \pi'[i])$$

- **Lemma:** If $B(s, s')$ and B is a bisimulation relation (**correction to Lemma 31**), then for every $\pi \in Path(s)$ there exists a corresponding path $\pi' \in Path(s')$ (**and vice versa**).
- Next, with 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 indeed follows:
if $M \equiv M'$ then $M \models f$ if and only if $M' \models f$,
(for all M, M' and CTL* formulas f)

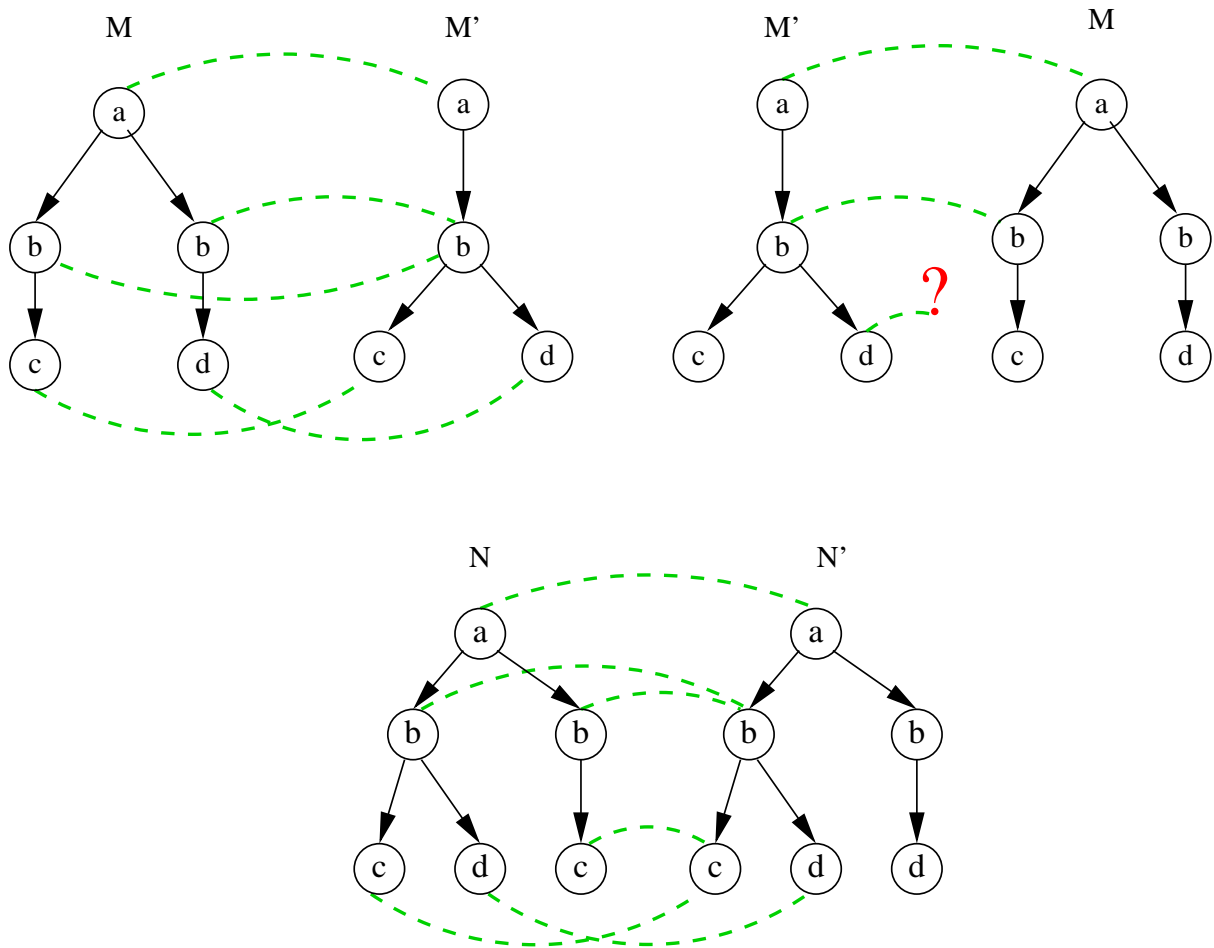
Simulation preorder: Definition

- 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 formulas.
- We say that system M' **simulates** system M , if M' has **at least** the behaviour of M .
- Let two Kripke Structures be given:
 $M = (AP, S, R, S_0, L)$ and
 $M' = (AP', S', R', S'_0, L')$, with $AP' \subseteq AP$.
- A relation $H \subseteq S \times S'$ is a **simulation relation** iff for every $s \in S$ and $s' \in S'$ with $H(s, s')$:
 1. $L(s) \cap AP' = L'(s')$
 2. for all s_1 , if $R(s, s_1)$, then there exists s'_1 such that $R'(s', s'_1)$ and $H(s_1, s'_1)$.

Simulation: defs (2)

- M' **simulates** M (written: $M \preceq M'$) iff there exists a simulation relation H , such that
$$\forall s_0 \in S_0. \exists s'_0 \in S'_0. H(s_0, s'_0)$$
- Note: \preceq is a **preorder** on Kripke structures (that is: transitive and reflexive).
- This defines an equivalence relation as follows:
$$M \preceq M' \text{ and } M' \preceq M.$$
- **Warning:**
 - it is possible that $M \preceq M'$ and $M' \preceq M$ but still $M \not\equiv M'$.
 - 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 a single B .

Simulation preorder: Examples



- Examples from Fig. 11.3 and 11.4 reconsidered:
- Note: $M \preceq M'$ but not $M' \preceq M$.
- Note: $N \preceq N'$ and by symmetry $N' \preceq N$, but as we saw earlier: $N \not\equiv N'$

ACTL* formulas

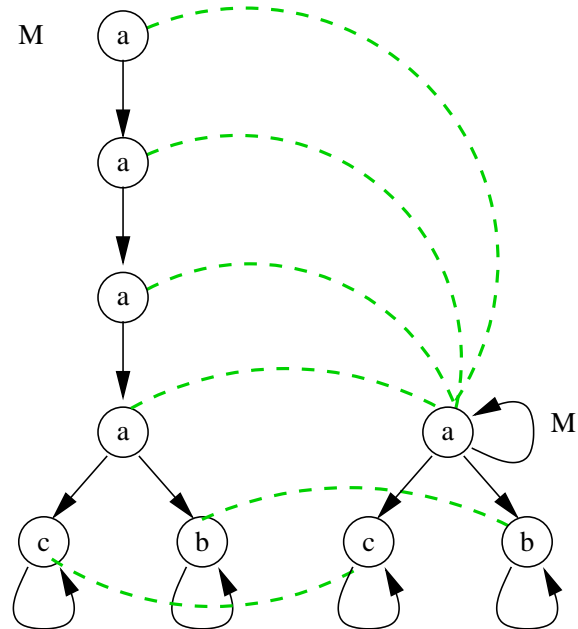
- ACTL* (see p. 31) is the fragment of CTL* with only universal path quantifiers, no existential path quantifiers.
- This only makes sense for formulas in **positive normal form**, i.e. negations only occur directly before atomic propositions.
- Examples: $\mathbf{A}(\mathbf{F} \mathbf{G} p)$, $\mathbf{A} \mathbf{G} (p \rightarrow \mathbf{A} \mathbf{X} q)$ are in ACTL*, but $\mathbf{A} \mathbf{G} (p \rightarrow \mathbf{E} \mathbf{X} q)$ is not.
Careful: $(\mathbf{A} \mathbf{G} p) \rightarrow (\mathbf{A} \mathbf{G} q)$ is not in ACTL*, because actually:

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

Simulation and Preservation of ACTL*

- We now have: If $H(s, s')$ and H is a simulation relation, then each path $\pi \in Path(s)$ corresponds to some $\pi' \in Path(s')$ (but not vice versa).
- **Theorem:** Let f be a formula in ACTL* over AP' . If $M \preceq M'$, then $M' \models f \Rightarrow M \models f$.
- **Practical consequence:** In order to check $M \models f$ it is safe to find an approximation M' with $M \preceq M'$, 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:
 $N \preceq N'$, $N' \preceq N$, but $N \not\equiv N'$. Hence,
 - N and N' satisfy the same ACTL* formulas,
 - but not the same CTL formulas.
 - They can only be distinguished by using **E**.

More ACTL* examples



- Observe that $M \preceq M'$ with H given above.
- Note that $M' \models \mathbf{AG} \neg d$ hence $M \models \mathbf{AG} \neg d$.
- Note that $M' \not\models \mathbf{AF} (b \vee c)$, but actually $M \models \mathbf{AF} (b \vee c)$. This shows that **some information is really lost**.
- Note: $M \models \mathbf{AX} a$ but $M' \not\models \mathbf{AX} a$
(wrong direction) conclusion: $M' \not\preceq M$
- Note: $M' \models \mathbf{EX} b$, but $M \not\models \mathbf{EX} b$.
(not in ACTL*)

Bisimulation Equivalence: Algorithm

- Let two Kripke Structures be given:
 $M = (AP, S, R, S_0, L)$ and
 $M' = (AP, S', R', S'_0, L')$.
- Define a sequence of relations $B_i^*(s, s')$ iff s and s' cannot be distinguished within i steps:
 - $B_0^*(s, s')$ if and only if $L(s) = L'(s')$
 - $B_{n+1}^*(s, s')$ if and only if:
 1. $B_n^*(s, s')$, and
 2. $\forall s_1$ with $R(s, s_1)$, $\exists s'_1$ with $R'(s', s'_1)$ and $B_n^*(s_1, s'_1)$.
 3. $\forall s'_1$ with $R'(s', s'_1)$, $\exists s_1$ with $R(s, s_1)$ and $B_n^*(s_1, s'_1)$.
 - Let $B^* := \bigcap_i B_i^*$
- Clearly, $B_i^* \supseteq B_{i+1}^*$
- So B^* can be computed by fixpoint iteration.
- Actually, this can be implemented symbolically by OBDD's

Bisimulation Reduction

- **Actually:** B^* is the largest bisimulation between M and M' .
- So: If s and s' are bisimilar, then $B^*(s, s')$.
- 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 $B^*(s_0, s'_0)$.
- 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).
- By carefully splitting equivalence classes, the procedure can run in $O(|R| \cdot \log(|S|))$ time (Paige-Tarjan).
- Similar ideas apply to checking $M \preceq M'$.

Conclusion

- Bisimulation is an equivalence relation.
- Bisimulation preserves CTL* formulas
- Simulation is a preorder
- Simulation preserves ACTL* formulas 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. generate state space
2. reduce it
3. model check,

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