Hoare logic and Model checking

Part II: Model checking

Lecture 11: Relating temporal models

Christopher Pulte cp526
University of Cambridge

CST Part II – 2023/24

In the last lecture we saw a (very naïve) implementation of CTL model checking.

In applying model checking to realistic artefacts, such as complex software or hardware, we may face the problem that the temporal model's state space is too large to explore.

In this lecture we will discuss the topic of abstracting a temporal model to reduce its state space.

Assume we wish to verify some artefact, and we have developed a **concrete** temporal model – a model that captures state space and transitions of the real artefact.

If the state space of the model is too large to model check directly, we can develop a more **abstract model**.

But how do we develop a good abstract model?

Abstracting temporal models

The premise of model checking is that checking the model translates to confidence in the modelled artefact.

If we have a concrete temporal model that is closer to the real artefact it is easier to gain confidence in this model.

We will see two ways of relating an abstract model to a concrete model/two criteria for showing that an abstract model is a good abstraction:

- simulation
- bisimulation

Easy abstraction

Assume we have a concrete temporal model M.

Easy optimisation: compute the **reachable** states S_r in M, and define a slightly simpler model M':

- $M'.S = S_r \subseteq M.S$
- $M'.S_0 = M.S_0$
- $s_1\ M'.T\ s_2$ whenever $s_1\ M.T\ s_2$ for two states $s_1, s_2 \in M'.S$
- $M'.\ell\ s = M.\ell\ s$ for every state $s \in M'.S$

Easy abstraction

Then the sets of paths from initial states in M and M' are the same, so: for all CTL* formulas ψ,

$$M' \models \psi \Rightarrow M \models \psi$$

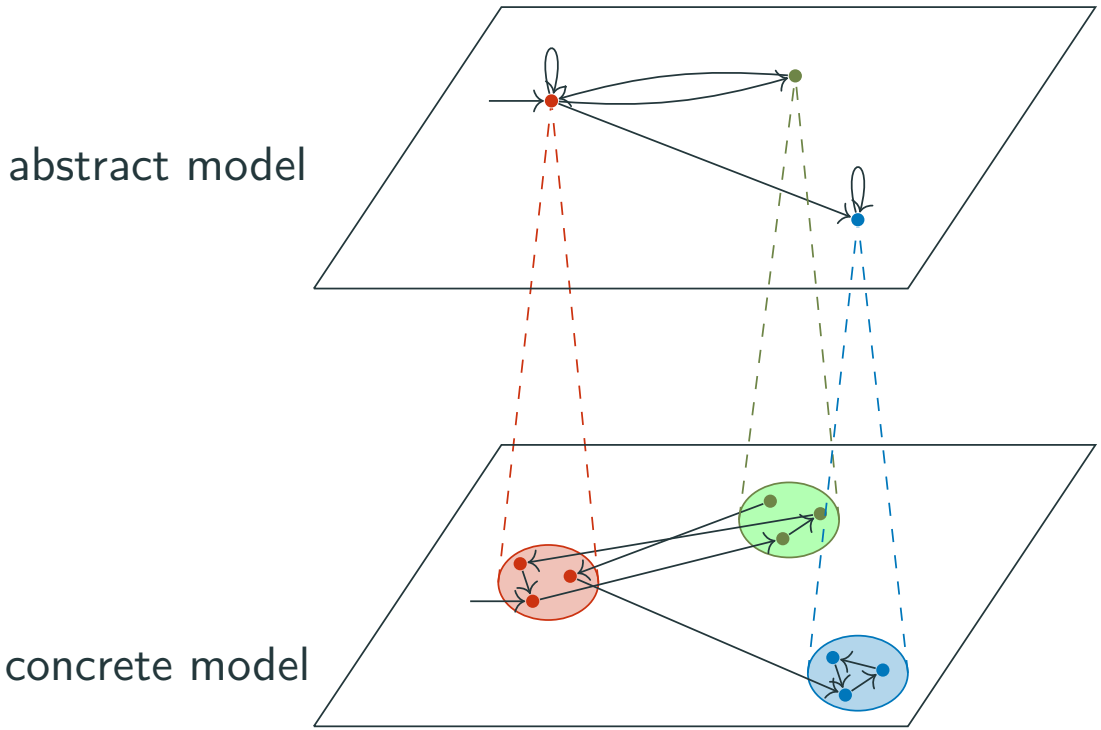
Since M' has a smaller state space, checking M' against ψ is simpler than checking M.

M' is a sound abstraction: properties verified about M' translate to properties verified about M, and (if M is a good concrete model) to properties verified about the artefact.

Can we generalise this observation?

Temporal model simulation, informally

The abstract model can match the steps of the concrete model and similarly labels states with atomic propositions.



- abstract temporal model can merge states
- a path in the abstract model may represent several paths in concrete model

Temporal model simulation, formally

Let $M = \langle S, S_0, T, \ell \rangle$ be a temporal model over AP and $M' = \langle S', S'_0, T', \ell' \rangle$ a temporal model over $AP' \subseteq AP$. A relation $R : S \times S' \to \mathbb{B}$ is a **simulation**[1] $M \preccurlyeq^R M'$ if:

1. R is consistent with labels:
   $\forall s \in S, s' \in S'.\ s\ R\ s' \Rightarrow \ell'\ s' = \ell\ s \cap AP'$
2. R relates initial states of M to initial states in M':
   $\forall s \in S.\ S_0\ s \Rightarrow \exists s' \in S'.\ S'_0\ s' \wedge s\ R\ s'$

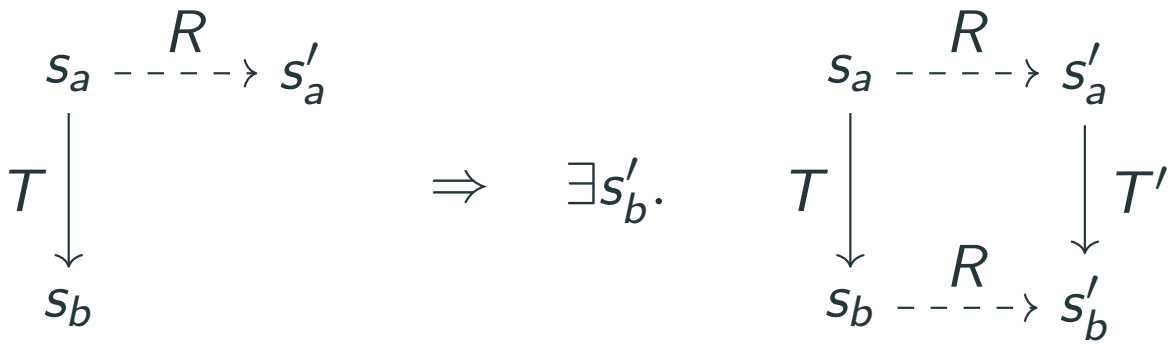
(continued on the next slide)

[1] Adopting Jan Willemse's definition

Temporal model simulation formally (continued)

3. Any step in M can be matched by a step in M' from any R-related start state to some R-related end state:

$$\forall s_a, s_b \in S, s'_a \in S'.\ (s_a\ R\ s'_a) \wedge (s_a\ T\ s_b) \Rightarrow$$
$$\exists s'_b \in S'.\ s'_a\ T'\ s'_b \wedge s_b\ R\ s'_b$$



Temporal model simulations

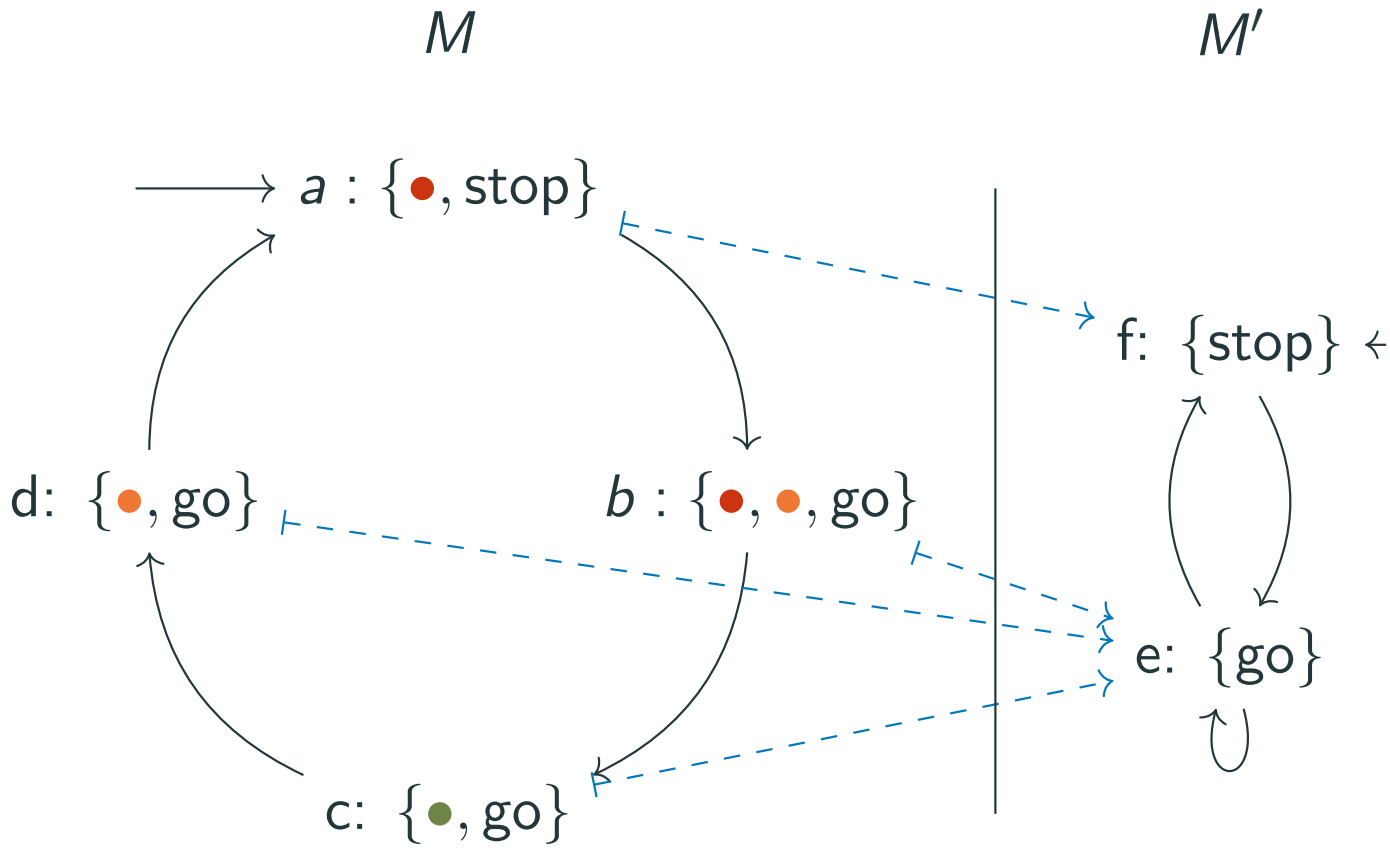
Often, only the existence of a simulation is important, not the simulation itself.

$$M \preccurlyeq M' \stackrel{def}{=} \exists R.\, M \preccurlyeq^{R} M'$$

It means that M' is "more abstract" than M.

Examples: a simulation (adapted from Grumberg)

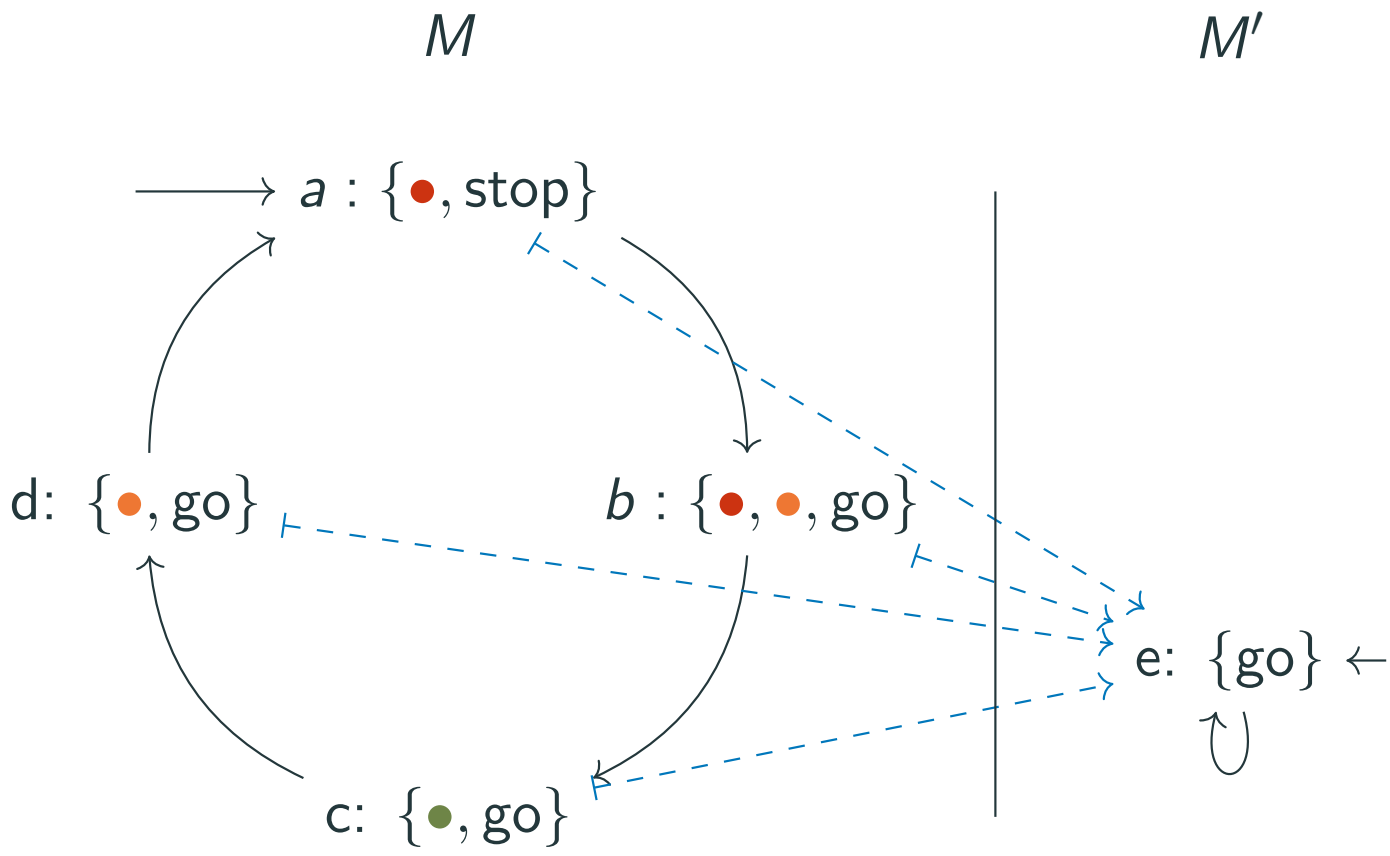
$AP ::= \bullet \mid \bullet \mid \bullet \mid \text{go} \mid \text{stop}$ $\quad AP' ::= \text{go} \mid \text{stop}$



This is a simulation between M and M'

Examples: not a simulation

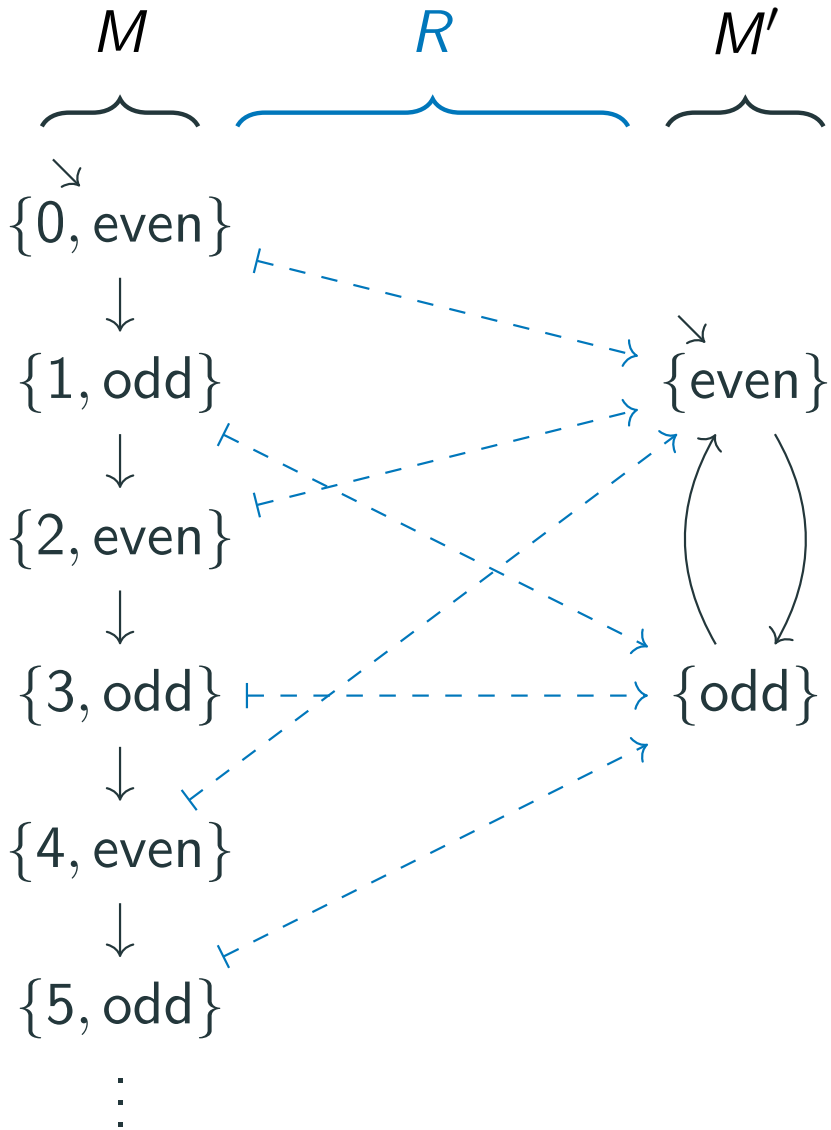
$AP ::= \bullet \mid \bullet \mid \bullet \mid \text{go} \mid \text{stop}$ $\quad AP' ::= \text{go} \mid \text{stop}$



R is not consistent with labels: $a\ R\ e$, but $\text{go} \in \ell'\ e$ and $\text{go} \notin \ell\ a$.

Examples: another simulation

$AP ::= \mathbb{N} \cup \{\text{even}, \text{odd}\}$ $\quad$ $AP' ::= \{\text{even}, \text{odd}\}$



ACTL* is compatible with simulation

ACTL* is compatible with the simulation preorder. (Recall: ACTL* is the universal fragment of CTL*: assuming negation normal form, the fragment of CTL* using only universal path quantification.)

Let M be a temporal model over AP, M' a temporal model over $AP' \subseteq AP$ and ψ an ACTL* formula over AP'. Then:

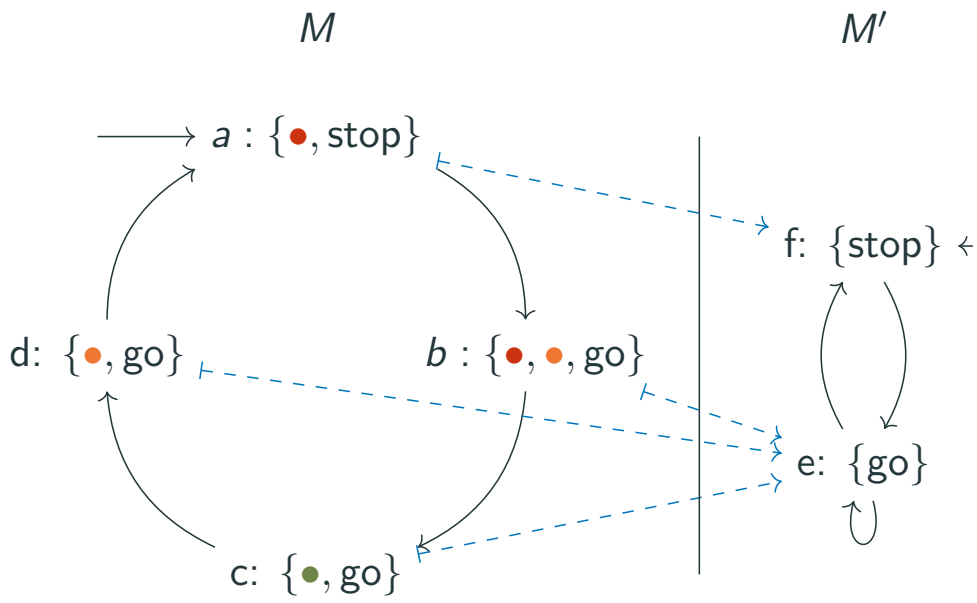
$$M \preccurlyeq M' \wedge M' \models \psi \Rightarrow M \models \psi$$

This means, we can model check M' against ψ, and if M' satisfies ψ, so does M.

Note: the implication only holds in one direction.
If $M' \models \psi$ fails this does not imply that $M \models \psi$ fails — this is a potential source of spurious counter examples.

Examples: traffic light simulation (adapted from Grumberg)

$AP ::= \bullet \mid \bullet \mid \bullet \mid \text{go} \mid \text{stop}$ $\quad AP' ::= \text{go} \mid \text{stop}$



- ACTL* formula A G A F (go) holds in M'; by $M \preccurlyeq M'$ this implies it also holds in M.
- ACTL* formula A G A F (stop) fails in M'; this does not imply it also fails in M.

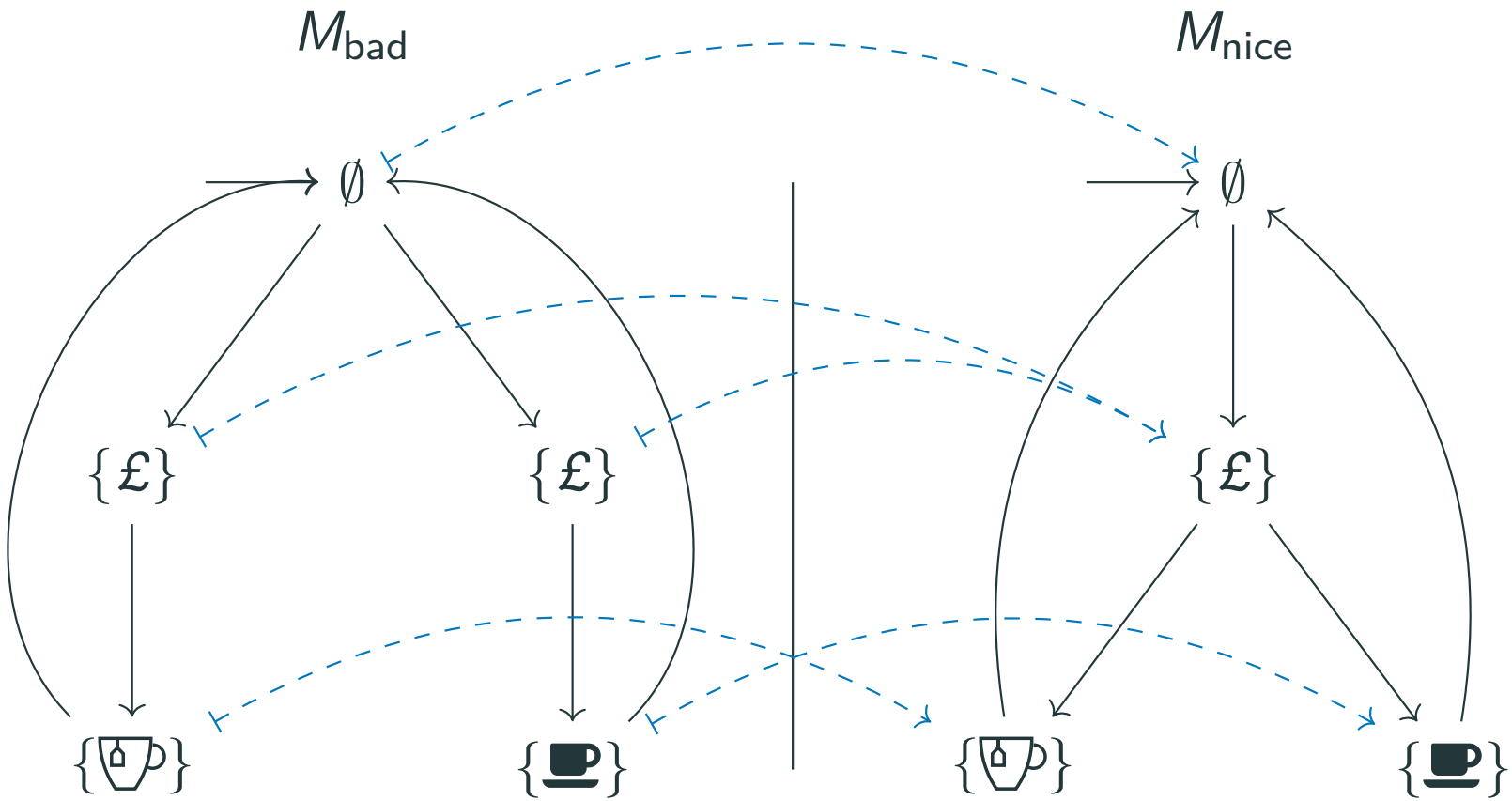
Simulation as criterion for "good" abstraction

+ simulation does not impose very strong requirements on the similarity of the models: more scope for "optimising" the abstract model[2]
– simulation is not compatible with arbitrary properties: only ACTL* formulas

[2] cf. Willemse slides

Tea & coffee machines

M_{nice} simulates M_{bad}. But: A G ($£ \rightarrow$ E X ☕) holds in M_{nice}, but not in M_{bad}.



Temporal model bisimulation

Let $M = \langle S, S_0, T, \ell \rangle$ and $M' = \langle S', S'_0, T', \ell' \rangle$ be temporal models over AP. Relation $R : S \times S' \to \mathbb{B}$ is a **bisimulation**[3] $M \approx^R M'$ if:

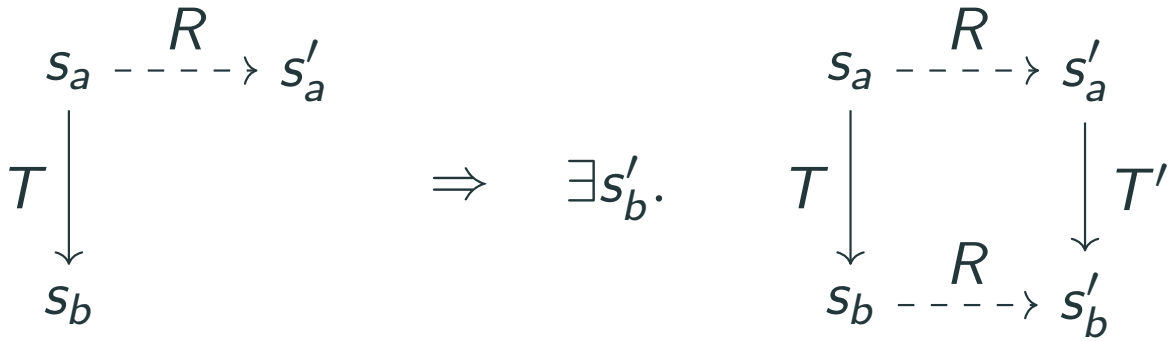
1. R is consistent with labels:
   $\forall s \in S, s' \in S'.\ s\ R\ s' \Rightarrow \ell'\ s' = \ell\ s$
2. R relates initial states:
   $\forall s \in S.\ S_0\ s \Rightarrow \exists s' \in S'.\ S'_0\ s' \wedge s\ R\ s'$
   $\forall s' \in S'.\ S'_0\ s' \Rightarrow \exists s \in S.\ S_0\ s \wedge s\ R\ s'$

(continued on the next slide)

[3] Adopting Jan Willemse's definition

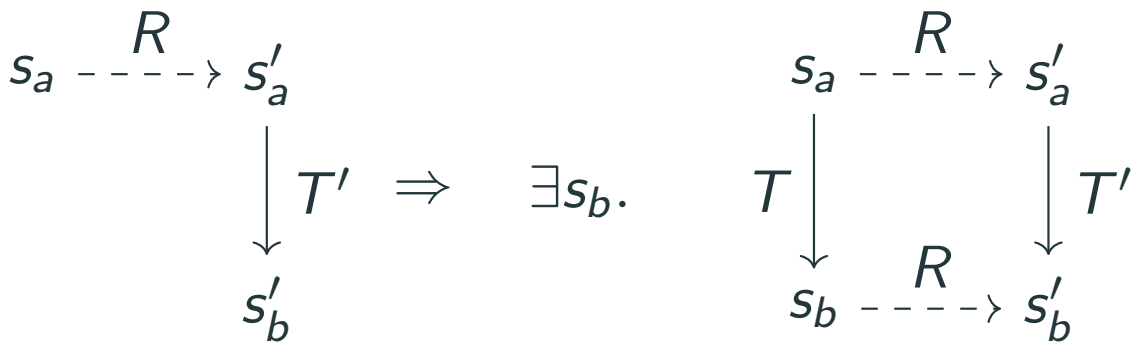
3.a) M' can match the steps of M:

$$\forall s_a, s_b \in S, s'_a \in S'.\ (s_a\ R\ s'_a) \wedge (s_a\ T\ s_b) \Rightarrow$$
$$\exists s'_b \in S'.\ s'_a\ T'\ s'_b \wedge s_b\ R\ s'_b$$



3.b) M can match the steps of M':

$$\forall s'_a, s'_b \in S', s_a \in S.\ (s_a\ R\ s'_a) \wedge (s'_a\ T'\ s'_b) \Rightarrow$$
$$\exists s_b \in S.\ s_a\ T\ s_b \wedge s_b\ R\ s'_b$$



Bisimilarity

As in the case of simulations, sometimes only the existence of a bisimulation is important, not the bisimulation itself

$$M \approx M' \stackrel{def}{=} \exists R.\, M \approx^R M'$$

We then call M and M' bisimilar.

Bisimulation preserves CTL*

All of CTL* is compatible with bisimulation equivalence.

Let M and M' be temporal models over AP and ψ a CTL* formula over AP. Then:

$$M \approx M' \Rightarrow (M \vDash \psi \Leftrightarrow M' \vDash \psi)$$

This means, if we model check M' against ψ:

- if M' satisfies ψ, then we know M also satisfies ψ,
- if M' fails ψ, then so does M (no spurious counter examples)

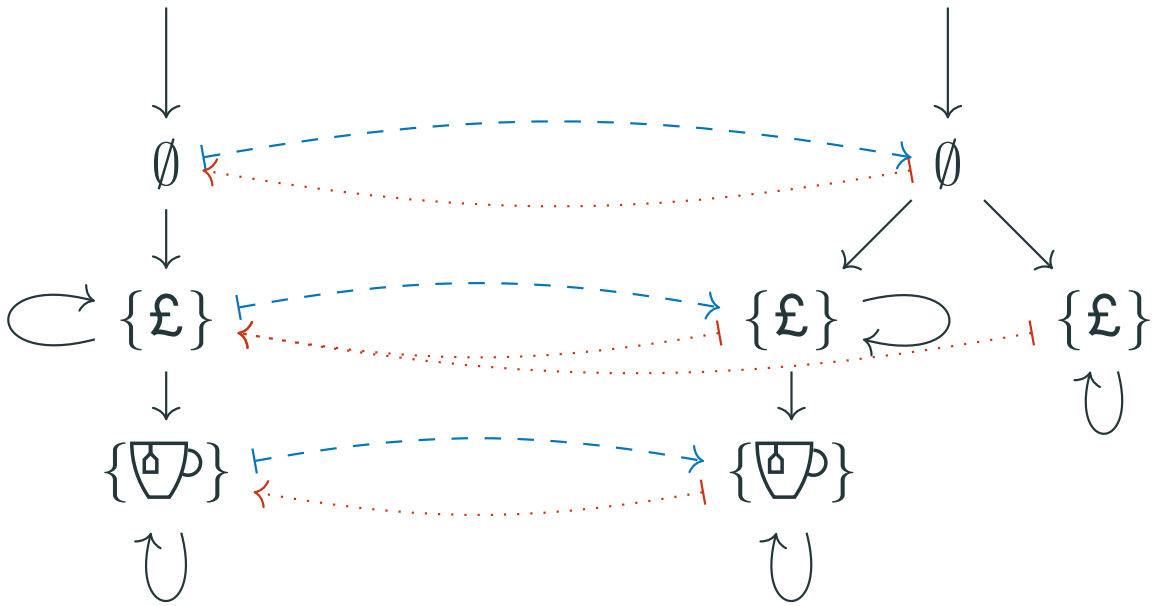
Caution: bisimulation and simulations

Bisimulation implies simulations in both directions

$$M \approx M' \Rightarrow (M \preccurlyeq M' \wedge M' \preccurlyeq M)$$

⚠ but in general not the other way around!

For example, on a variation of the tea & coffee machines example:



Here the blue relation is not the inverse of the red relation.

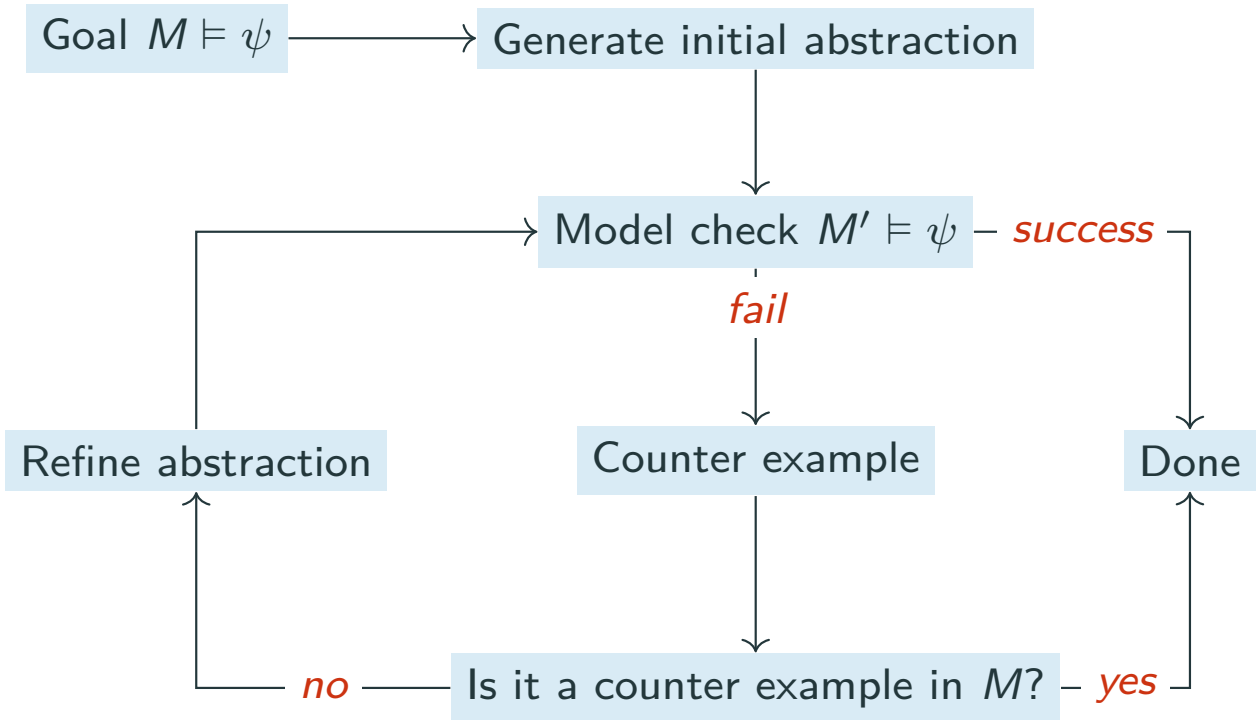
Revisiting stuttering

What if we want to abstract multiple steps of the concrete model with one step of the abstract model?

$\rightsquigarrow$ We can change our notion of path to allow staying any finite number of times in any state (in addition to allowing forever on states with self-loops).

We can then adapt most of the notions we have seen so far. However, in this setting, we do not want to use the X temporal operator.

CEGAR – Counter Example Guided Abstraction Refinement



Lots of detail to fill out:

- how to generate the abstraction
- how to check counter examples in M
- how to refine abstractions

Summary

Abstracting a concrete temporal model into an abstract one is a method for reducing the large state space in concrete models of complex artefacts.

Simulation guarantees that model checking the abstract model is sound for ACTL* properties, bisimulation for CTL* properties.

CEGAR is a method based on iterative refinement of abstract models.

Overall summary

We have seen model checking as a method for checking the correctness of various kinds of artefacts, including software and hardware systems.

Artefacts are mathematically captured in temporal models that model checkers check against temporal logic specifications.

Designing suitable temporal models can require effort and expert knowledge, but model checking an existing model against a specification is typically push-button/automatic.

Some links

- **SPIN model checker** (supports LTL specs).
  Download, Tutorial, Successes
- **NuSMV model checker** (supports LTL and CTL specs).
  Download, Tutorial
- **TLA+ model checker**.
  Download, Tutorial, Successes
- **CBMC model checker for C**.
  Download, Manual, Projects