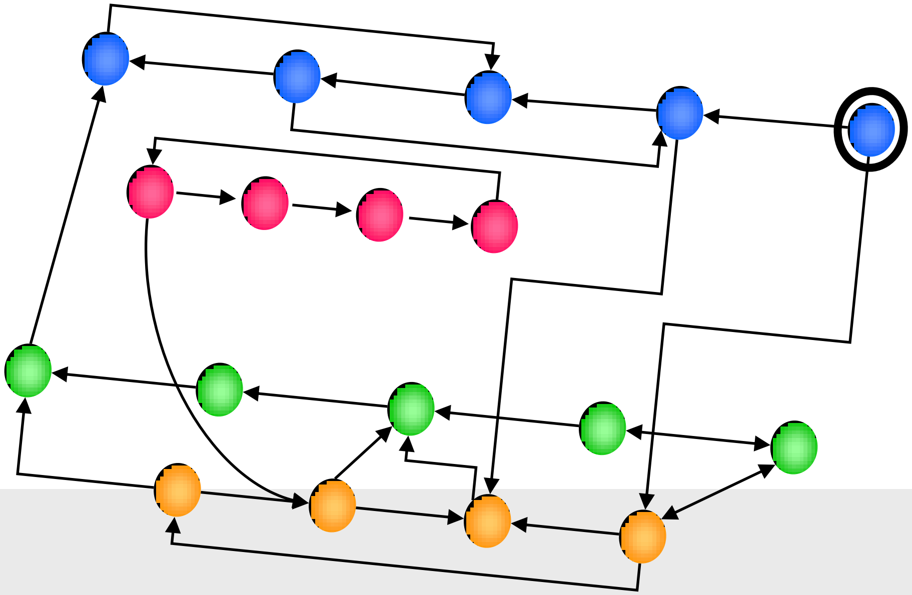
Branching Time Model Checking and Abstraction

Helmut Veith

Branching Time Logic

Kripke Structures



Kripke structures

K = (States, Transition Relation, Initial States, Labelling) = (S,R,I,L)

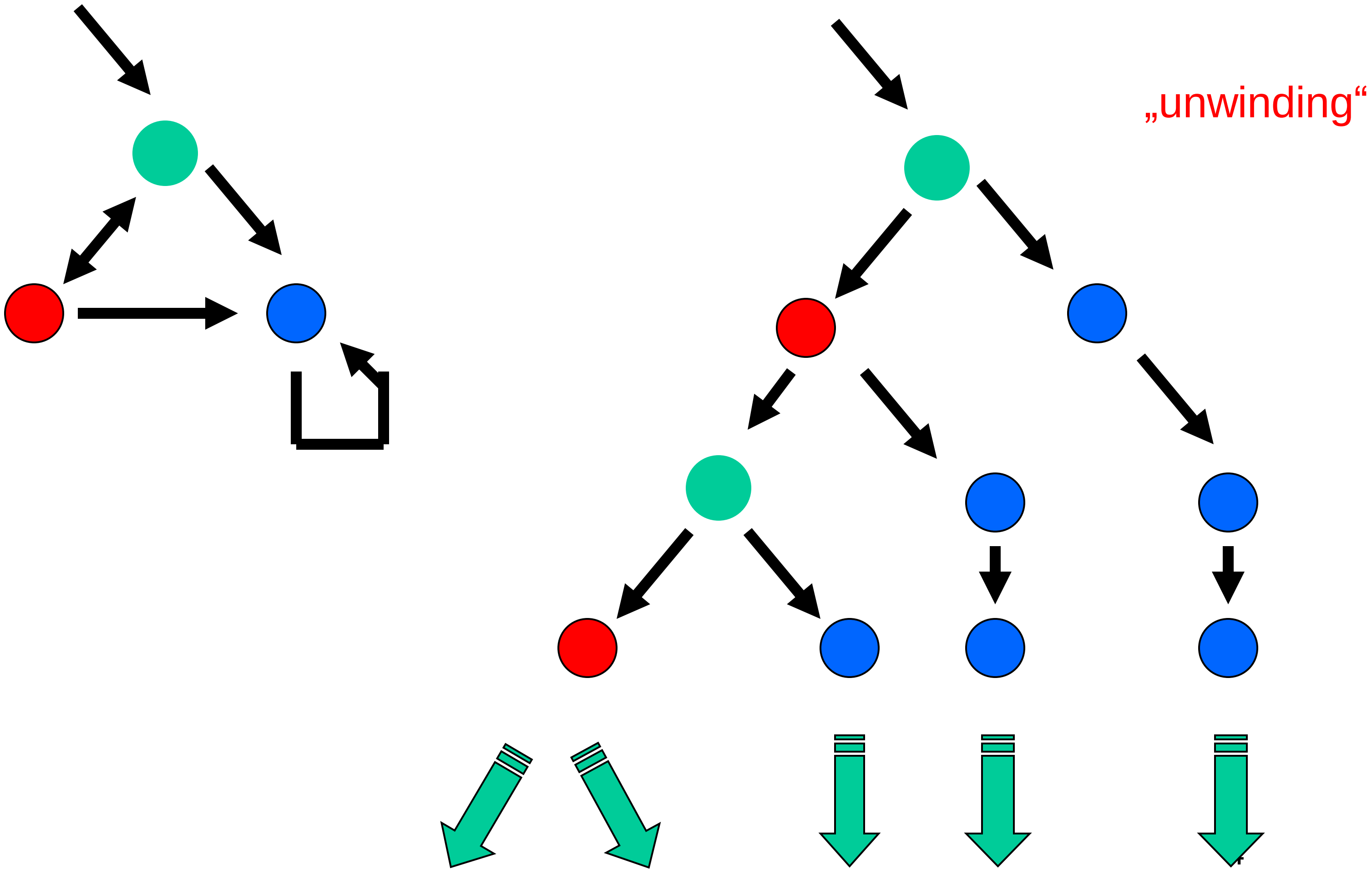
Specifications

Temporal logic, e.g. CTL (branching time) and LTL (linear time)

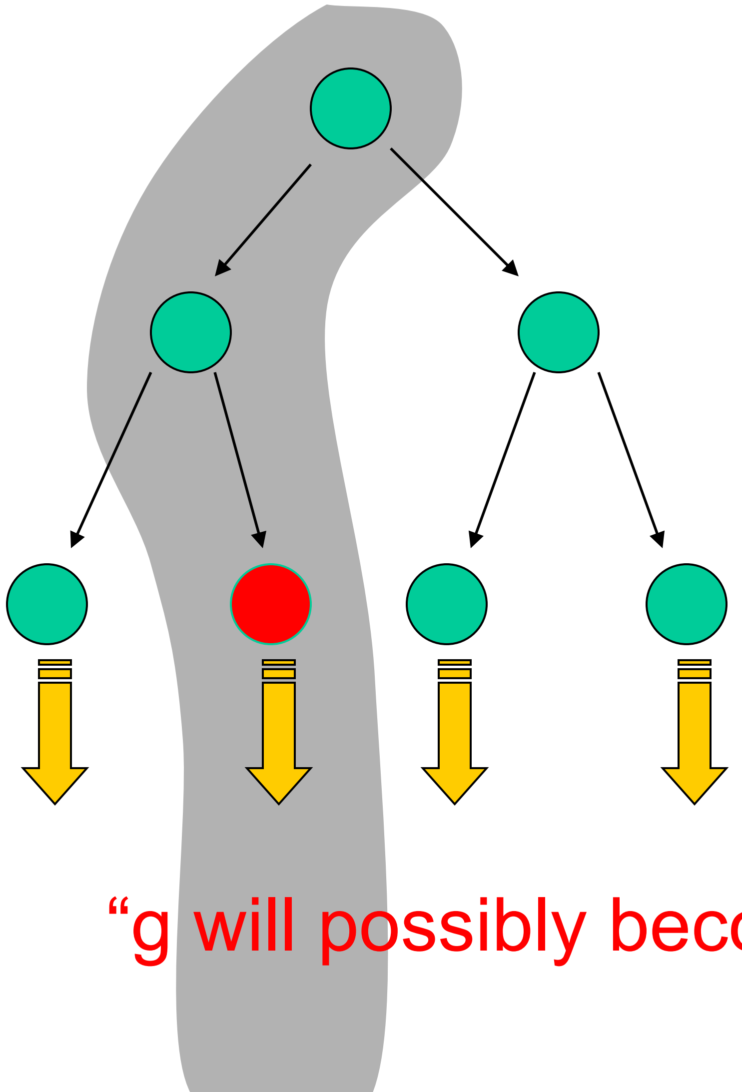
Model Relation

K |= f Specification f holds true in model K

Branching Time Logic

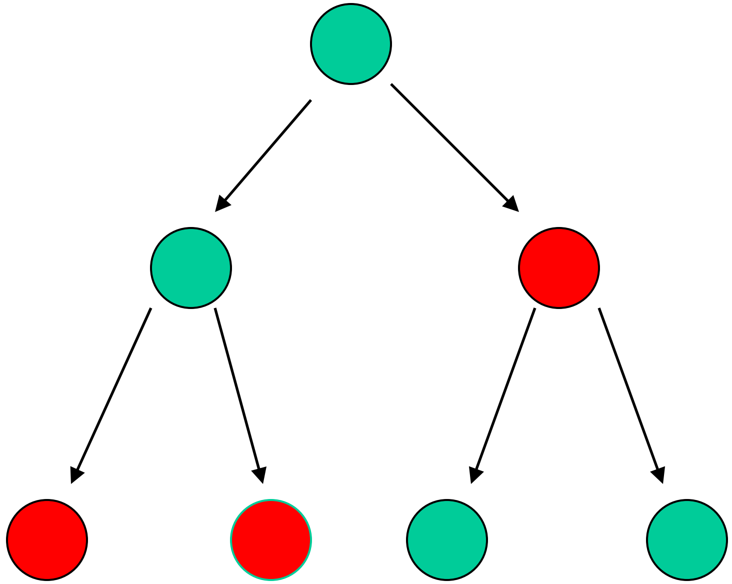


CTL - Computation Tree Logic



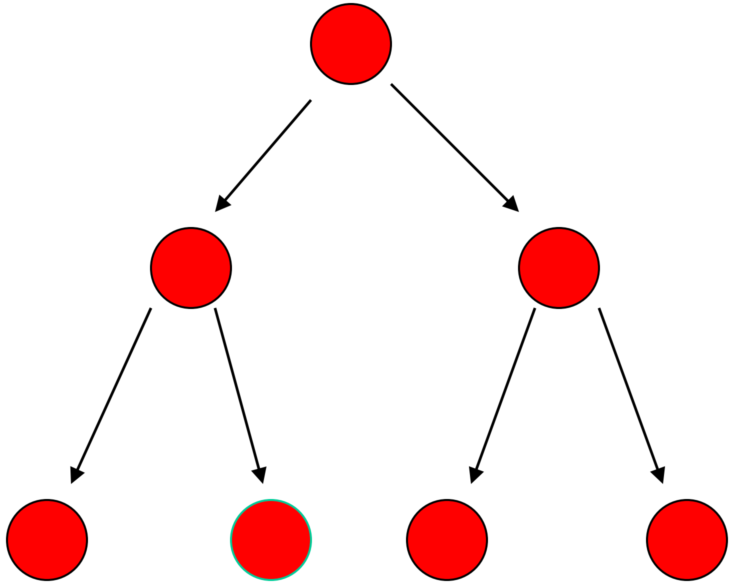
EF g "g will possibly become true"

CTL - Computation Tree Logic



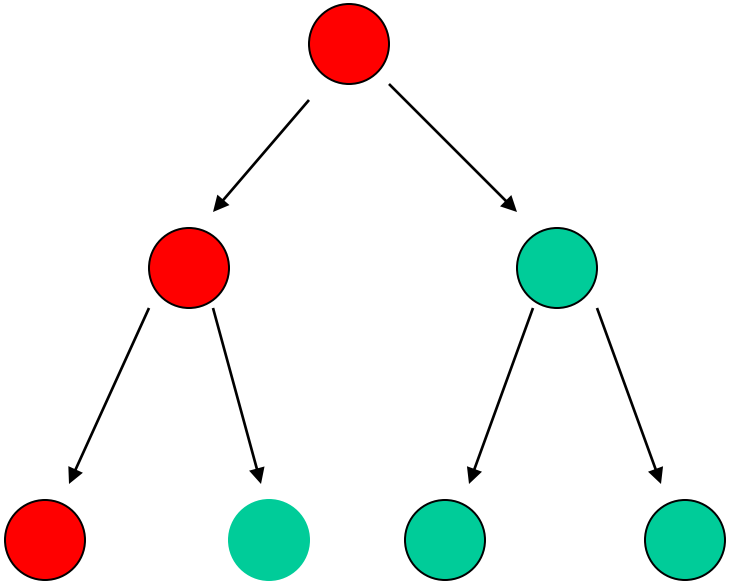
AF g "g will necessarily become true"

CTL - Computation Tree Logic



AG g "g is an invariant"

CTL - Computation Tree Logic



EG g “g is a potential invariant”

Computation Tree Logic

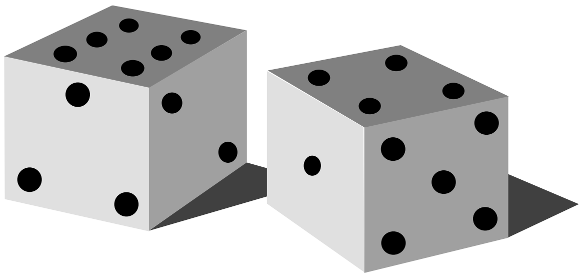
Computation Tree Logic

ACTL	AX, AG, AF, AU
ECTL	EX, EG, EF, EU
CTL	ACTL & ECTL
CTL*	AXX, AGX, EXF, ...

Family of Temporal Logics

Simulation and Bisimulation

Simulation Game



Combinatorial two player game between Spoiler and Duplicator.

Spoiler wins if Duplicator gets stuck.

Duplicator wins if game continues forever.

Example of a Combinatorial Game.

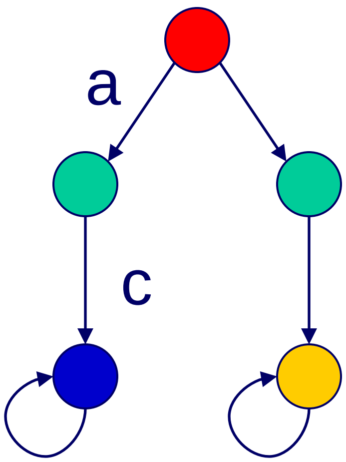
→ *Ehrenfeucht-Fraissee Games, Pebble Games, Parity Games etc.*

Simulation

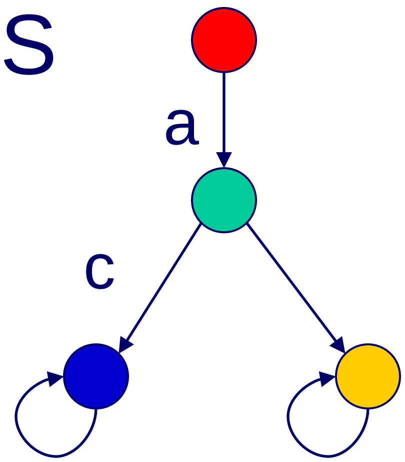
I can be simulated by S step by step.

“S simulates I”: $I \leq S$

I



S



The simulation preorder [Milner]

Given two models $M_1 = (S_1,I_1,R_1,L_1)$, $M_2 = (S_2,I_2,R_2,L_2)$

$\mathbf{H} \subseteq \mathbf{S_1} \times \mathbf{S_2}$ is a **simulation** iff

for every $(\mathbf{s_1}, \mathbf{s_2}) \in H$:

- s_1 and s_2 satisfy the same propositions
- For every successor $\mathbf{t_1}$ of $\mathbf{s_1}$ there is a successor $\mathbf{t_2}$ of $\mathbf{s_2}$ such that $(\mathbf{t_1},\mathbf{t_2}) \in \mathbf{H}$

Notation: $\mathbf{s_1} \leq \mathbf{s_2}$

The simulation preorder [Milner]

Given two models $M_1 = (S_1,I_1,R_1,L_1)$, $M_2 = (S_2,I_2,R_2,L_2)$

$\mathbf{H \subseteq S_1 \times S_2}$ is a **simulation** iff

for every $(s_1, s_2) \in H$:

- $\forall p \in AP$: $s_2 \models p \Rightarrow s_1 \models p$
  $s_2 \models \neg p \Rightarrow s_1 \models \neg p$

- $\forall t_1 [(s_1,t_1) \in R_1 \Rightarrow \exists t_2 [(s_2,t_2) \in R_2 \wedge (t_1,t_2) \in H]]$

Notation: $\mathbf{s_1 \leq s_2}$

Simulation preorder (cont.)

$\mathbf{H} \subseteq \mathbf{S_1}$ **x** $\mathbf{S_2}$ is a **simulation** from M_1 to M_2 iff

H is a simulation and

for every $\mathbf{s_1} \in I_1$ there is $\mathbf{s_2} \in I_2$ s.t. $(\mathbf{s_1}, \mathbf{s_2}) \in \mathbf{H}$

Notation: $\mathbf{M_1 \leq M_2}$

Bisimulation relation [Park]

For models M_1 and M_2, $\mathbf{H} \subseteq \mathbf{S_1} \times \mathbf{S_2}$ is a **bisimulation**

iff for every $(s_1, s_2) \in H$:

- $\forall p \in AP : p \in L(s_2) \Leftrightarrow p \in L(s_1)$
- $\forall t_1 [(s_1, \mathbf{t_1}) \in R_1 \Rightarrow \exists t_2 [(s_2, \mathbf{t_2}) \in R_2 \wedge \mathbf{(t_1, t_2) \in H}]]$

- $\forall \mathbf{t_2} [(\mathbf{s_2}, \mathbf{t_2}) \in \mathbf{R_2} \Rightarrow \exists \mathbf{t_1} [(\mathbf{s_1}, \mathbf{t_1}) \in \mathbf{R_1} \wedge (\mathbf{t_1}, \mathbf{t_2}) \in \mathbf{H}]]$

Notation: $\mathbf{s_1} \equiv \mathbf{s_2}$

Bisimulation relation (cont.)

$\mathbf{H \subseteq S_1 \times S_2}$ is a **Bisimulation** between M_1 and M_2

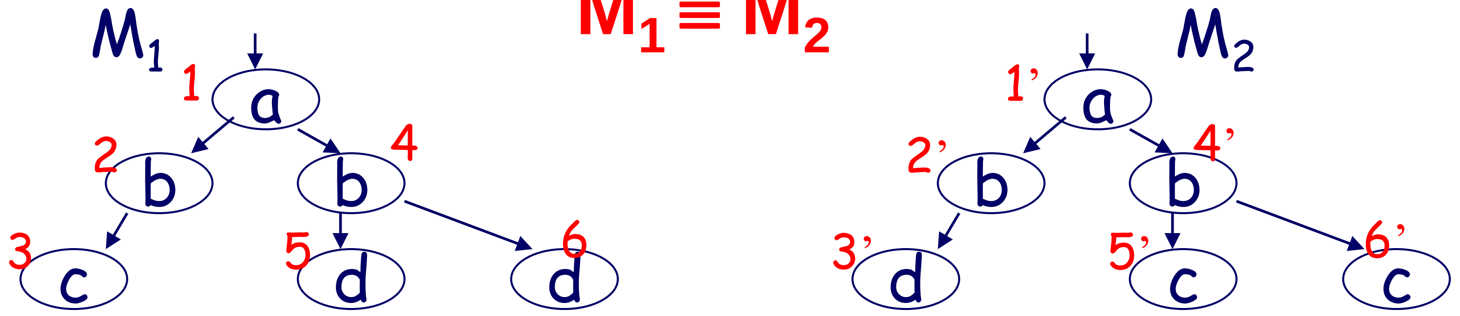
iff H is a bisimulation and

for every $s_1 \in I_1$ there is $s_2 \in I_2$ s.t. $(s_1, s_2) \in H$ and

for every $s_2 \in I_2$ there is $s_1 \in I_1$ s.t. $(s_1, s_2) \in H$

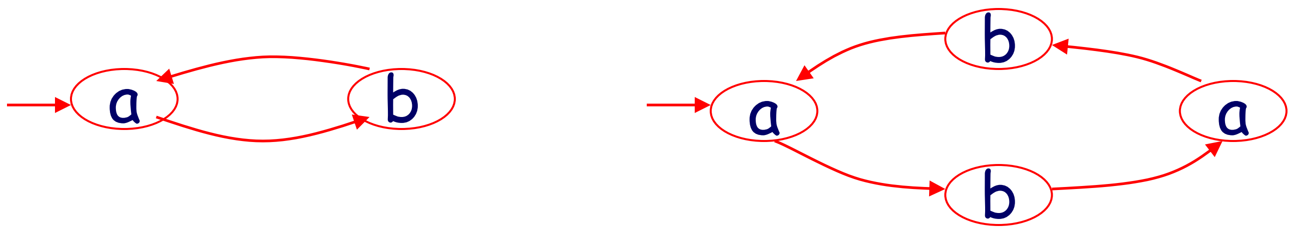
Notation: $\mathbf{M_1 \equiv M_2}$

Bisimulation equivalence



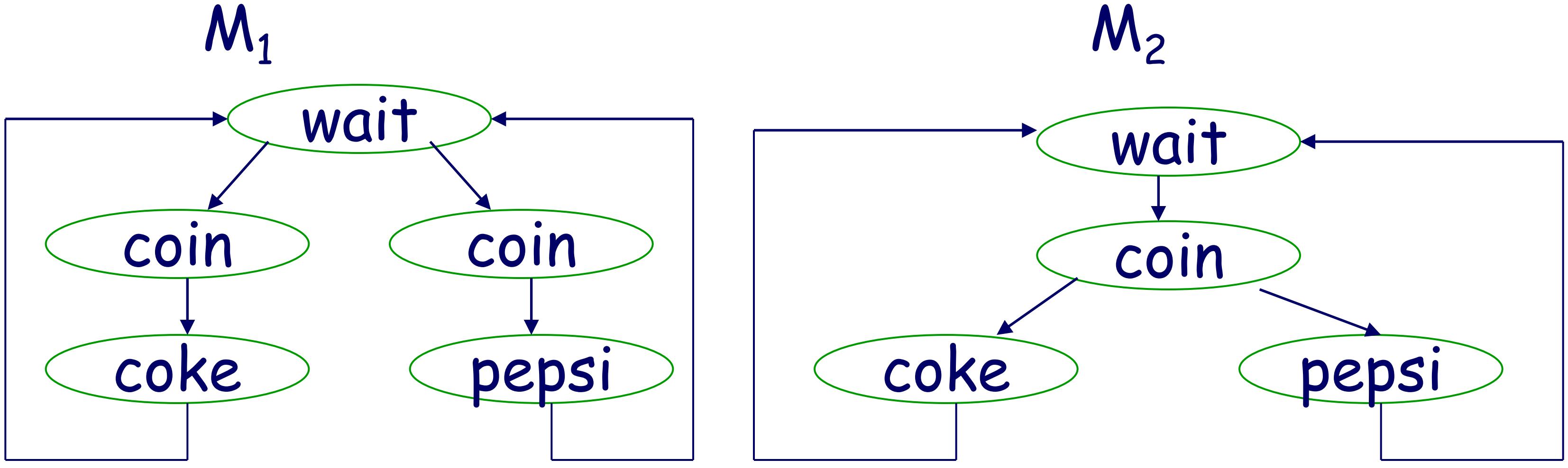
$M_1 \equiv M_2$

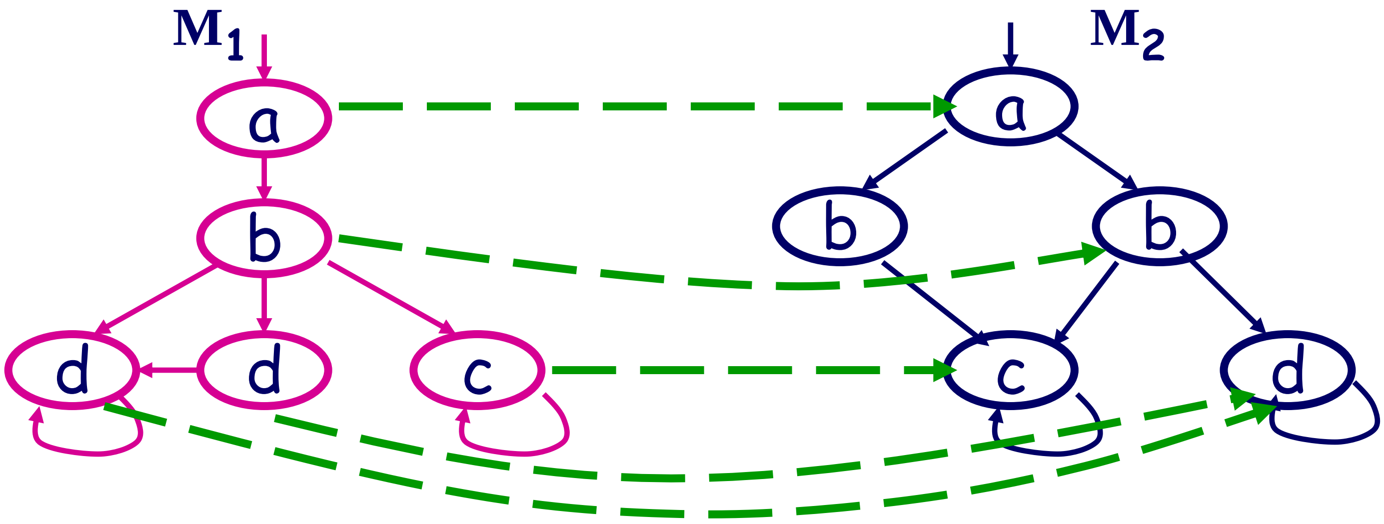
H={ (1,1'), (2,4'), (4,2'), (3,5'), (3,6'), (5,3'), (6,3') }



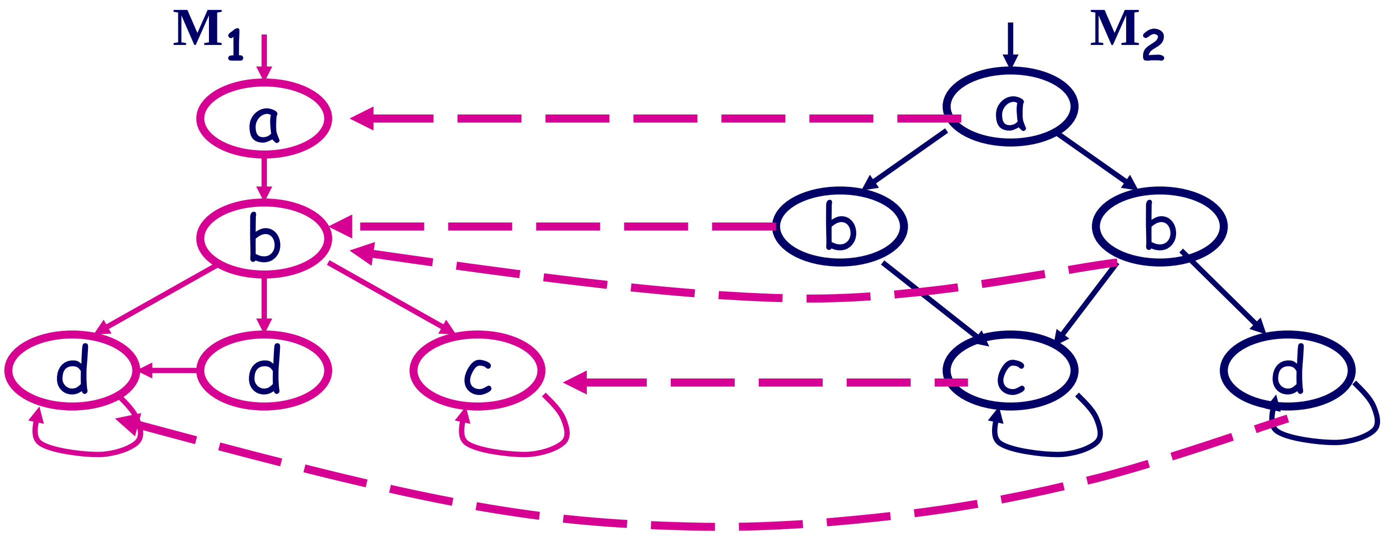
Simulation preorder

$$M_1 \leq M_2$$





$M_1 \leq M_2$



$M_1 \le M_2$ and $M_1 \ge M_2$ but not $M_1 \equiv M_2$

(bi)simulation and logic preservation

Theorem

If $\mathbf{M_1} \equiv \mathbf{M_2}$ then for every **CTL*** formula φ,

$M_1 \models \varphi \Leftrightarrow M_2 \models \varphi$

If $\mathbf{M_2} \geq \mathbf{M_1}$ then for every **ACTL*** formula φ,

$M_2 \models \varphi \Rightarrow M_1 \models \varphi$

Simulation Relation

If M has partial behavior of N, we say that

"N simulates M": $M \leq N$

Let f be an ACTL specification.

If $M \leq N$ and $N \models f$ then $M \models f$.

Simulation and Abstraction

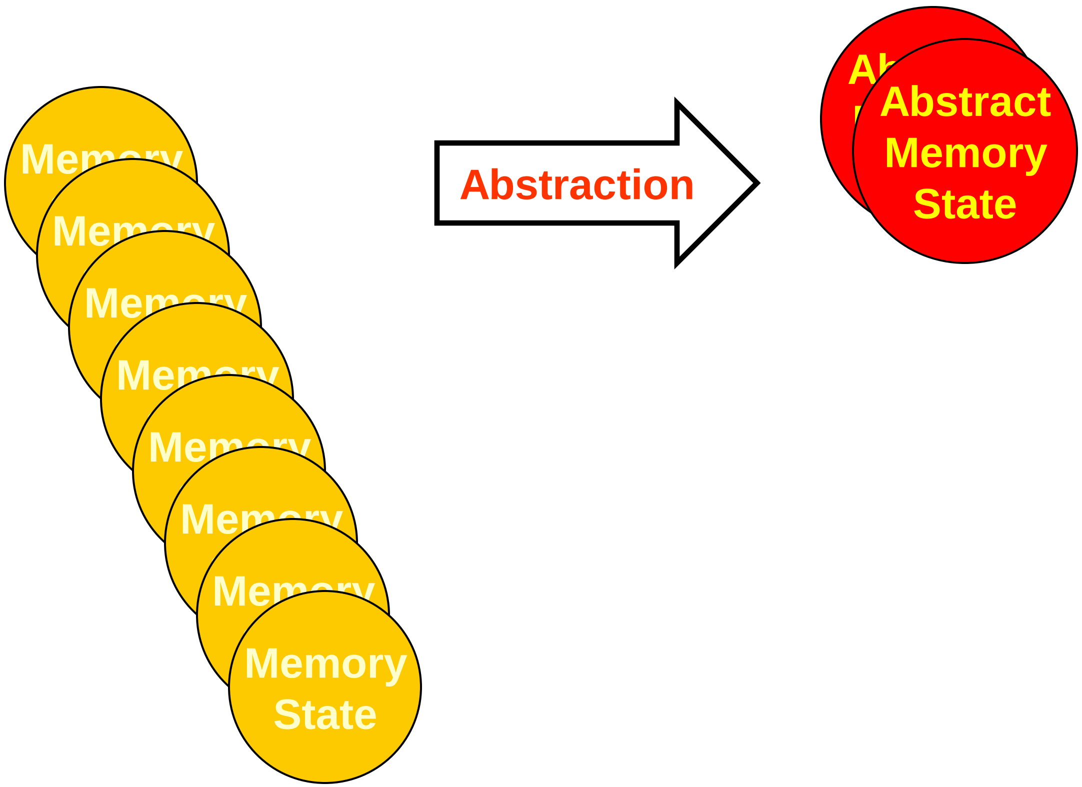
If M has partial behavior of N, we say that

"N simulates M": $M \leq N$

$M \models f$ ⟵ $N \models f$

Abstraction

Abstraction



Data Abstraction

Given a program P with variables $x_1,\ldots x_n$, each over domain D,

the **concrete model** of P is defined over states
$(d_1,\ldots,d_n) \in D\times\ldots\times D$

Choosing

- abstract domain **A**
- Abstraction mapping (surjection) **$h: D \rightarrow A$**

we get an **abstract model** over abstract states
$(a_1,\ldots,a_n) \in A\times\ldots\times A$

Example

Given a program P with variable x over the integers

Abstraction 1:

$A_1 = \{ \mathbf{a_{-}}, \mathbf{a_0}, \mathbf{a_+} \}$

$$h_1(d) = \begin{cases} \mathbf{a_+} & \text{if } d>0 \\ \mathbf{a_0} & \text{if } d=0 \\ \mathbf{a_-} & \text{if } d<0 \end{cases}$$

Abstraction 2:

$A_2 = \{ \mathbf{a_{even}}, \mathbf{a_{odd}} \}$

$h_2(d) =$ if even(d) then $\mathbf{a_{even}}$ else $\mathbf{a_{odd}}$

Reduced abstract model
Existential abstraction

Given M, A, $h : D \rightarrow A$

the **reduced model** $M_r = (S_r, I_r, R_r, L_r)$ is

$\mathbf{S_r} = A \times ... \times A$

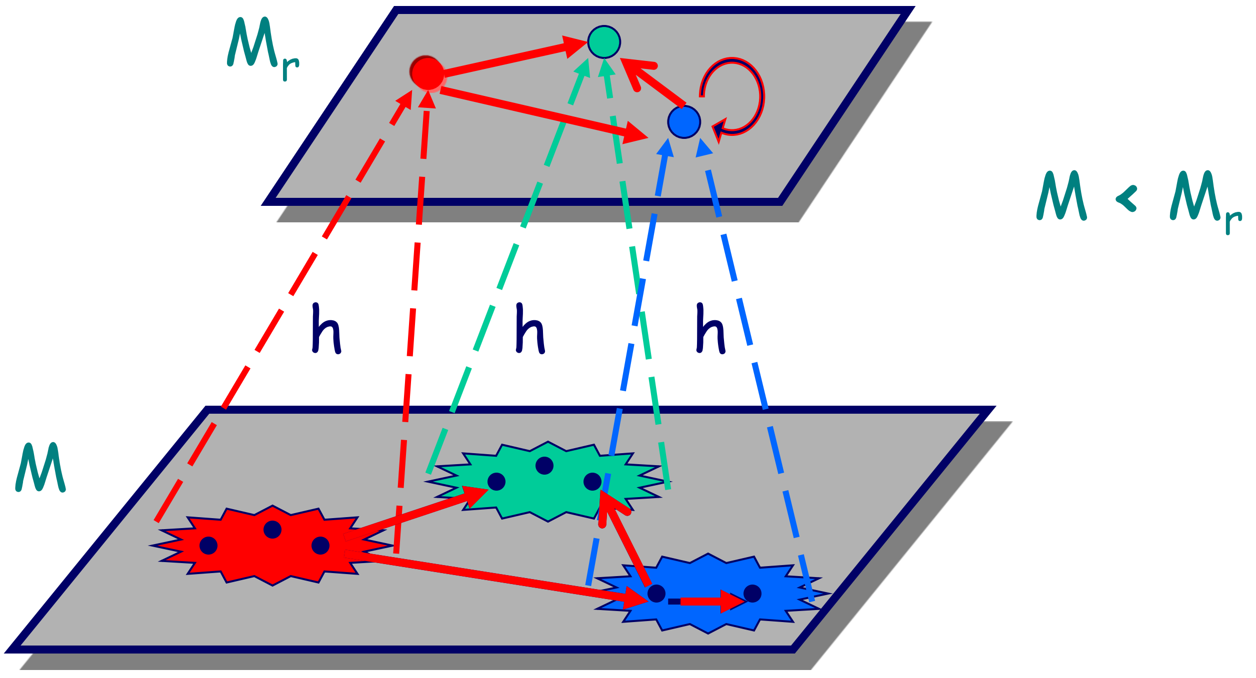
$\mathbf{s_r \in I_r} \Leftrightarrow \exists\, \mathbf{s} \in I : h(\mathbf{s}) = s_r$

$\mathbf{(s_r,t_r) \in R_r} \Leftrightarrow$

$\exists\, \mathbf{s,t}\, [h(\mathbf{s}) = s_r \wedge h(\mathbf{t}) = t_r \wedge (\mathbf{s},\mathbf{t}) \in R]$

For $s_r = (a_1,\ldots,a_n)$, $\mathbf{L_r(s_r) =} \{ (x_i^A = a_i) \mid i = 1, \ldots, n \}$

Existential Abstraction



Preservation

Theorem:

$M_r \geq M$ by the simulation preorder

Corollary:

For every ACTL* formula φ:

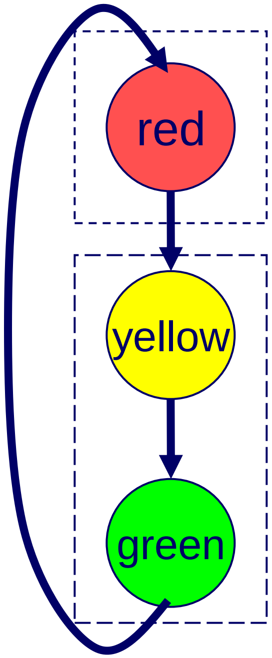
If $M_r \models \varphi$ then $M \models \varphi$

Traffic Light Example

Property:

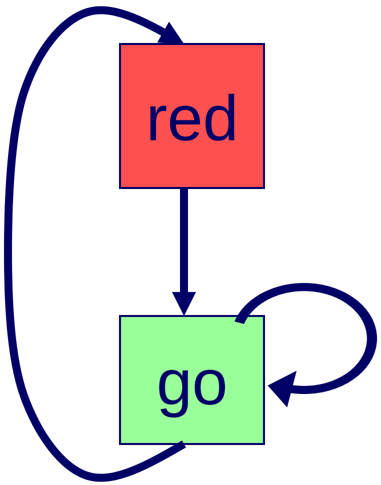
φ = **AG AF** ¬ (**state=red**)

Abstraction function h maps green, yellow to go.



M

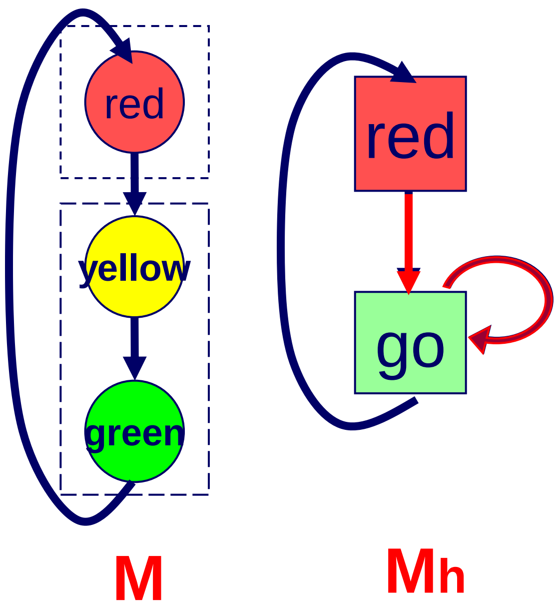
$$M \models \varphi \Leftarrow M_h \models \varphi$$



M_h

Traffic Light Example (Cont)

If the abstract model invalidates a specification, the actual model may still satisfy the specification.



- Property:

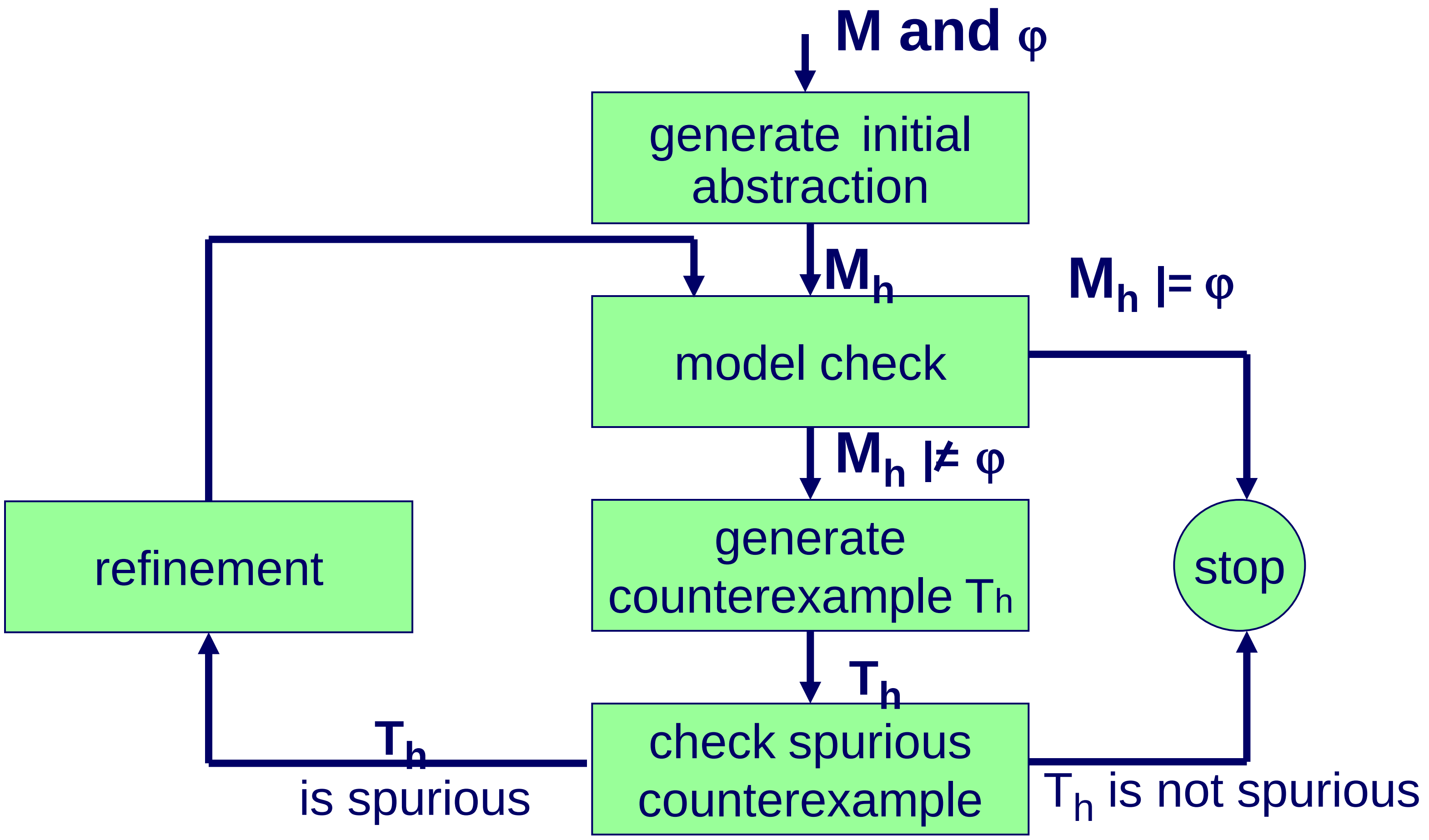
  φ =AG AF (state=red)

- **M |= φ but M_h |≠ φ**

- **Spurious** Counterexample:

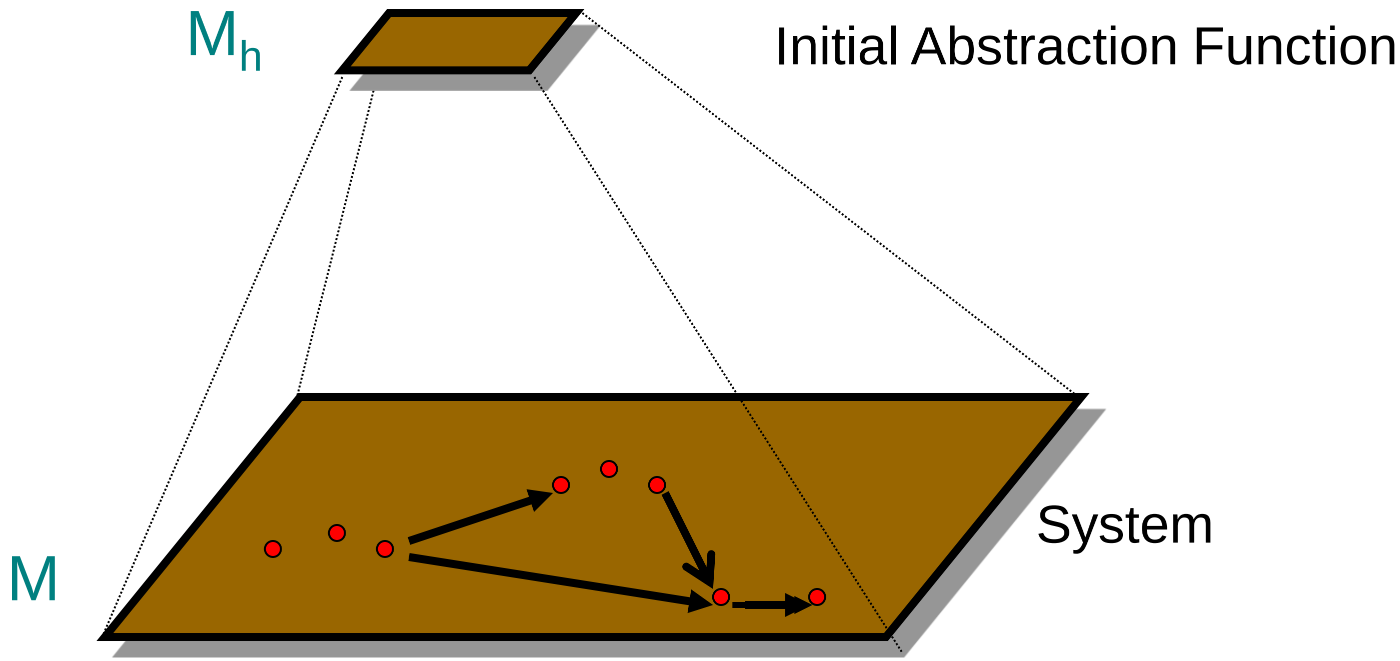
  ⟨red,go,go, ...⟩

CEGAR Methodology



CEGAR (Counterexample-Guided Abstraction Refinement)

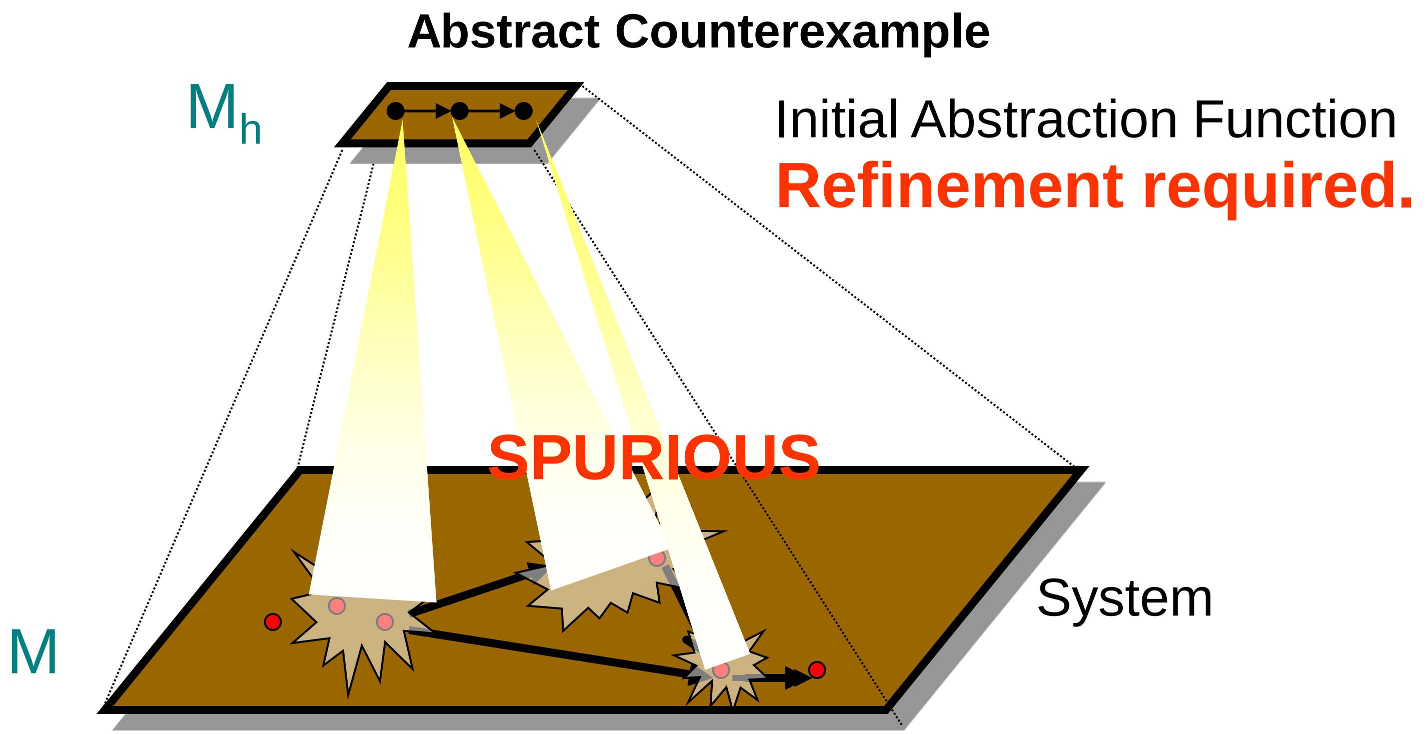
Adaptive Strategy



Counterexample-Guided Abstraction Refinement
Clarke, Grumberg, Jha, Lu, Veith'00

CEGAR (Counterexample-Guided Abstraction Refinement)

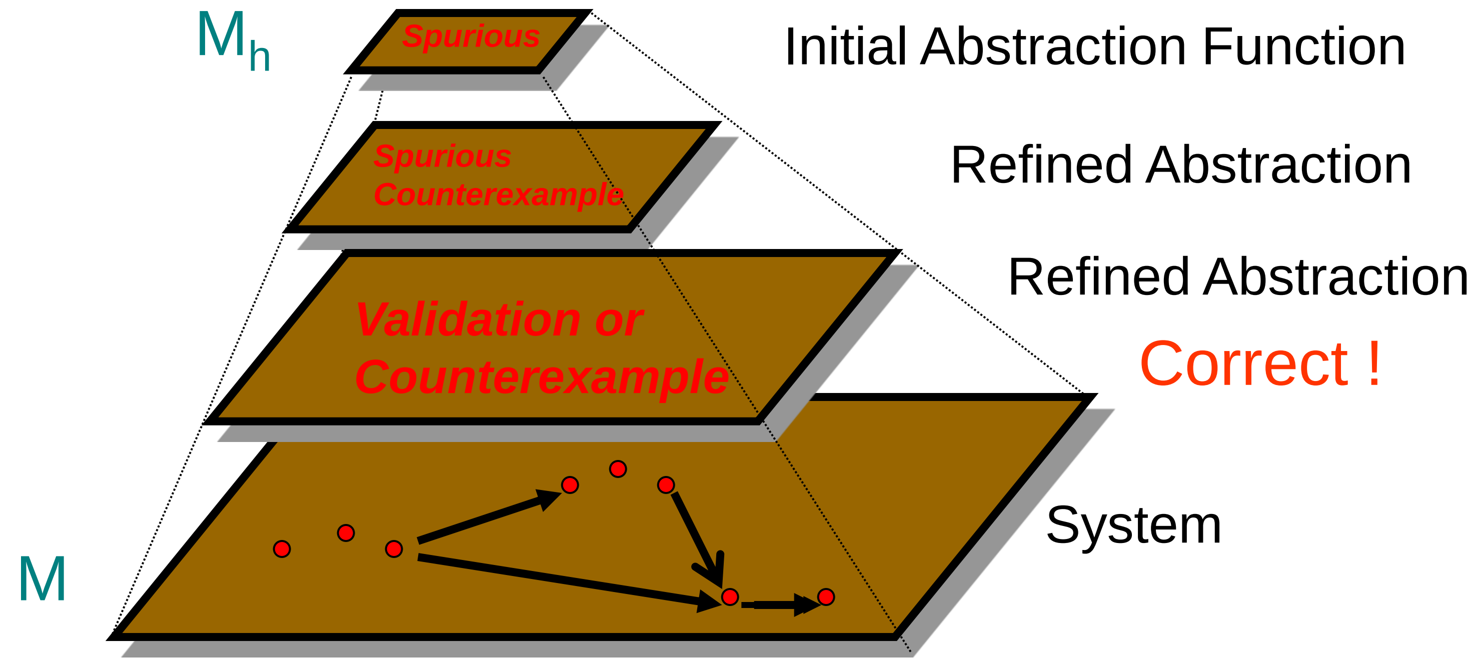
Adaptive Strategy



Counterexample-Guided Abstraction Refinement
Clarke, Grumberg, Jha, Lu, Veith'00

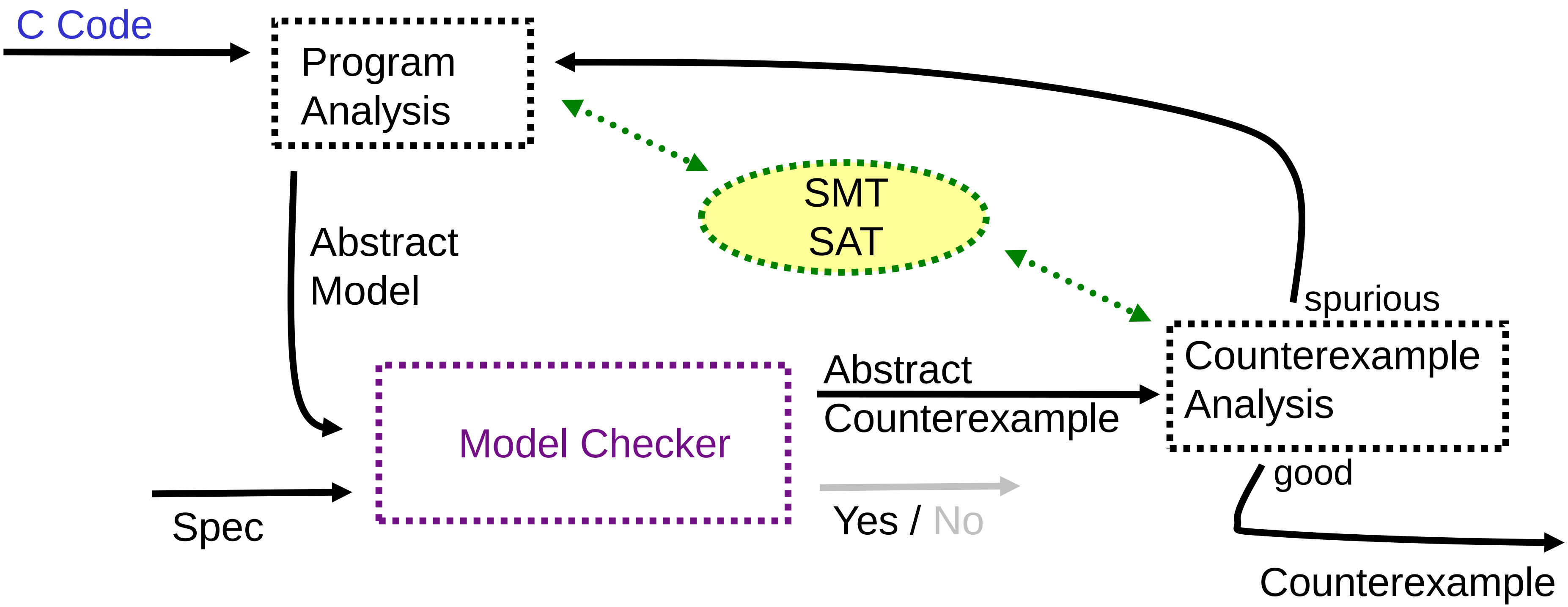
CEGAR (Counterexample-Guided Abstraction Refinement)

Adaptive Strategy



Counterexample-Guided Abstraction Refinement
Clarke, Grumberg, Jha, Lu, Veith'00

Software Model Checking



CEGAR + Predicate Abstraction

Integration of Theorem Proving / Decision Procedures / SMT

SIGSOFT Distinguished Paper Award (ICSE 2003)