Outline

Model Checking
- Model Checking Problem
- Safety Properties and Reachability
- Symbolic Reachability Checking

Putting it All Together

When we design a system, we would like to be sure that it will satisfy all requirements, such as safety.

Putting it All Together

When we design a system, we would like to be sure that it will satisfy all requirements, such as safety.

Now we can treat the safety problem as a mathematical problem. We can

- formally represent our system as a transition system (the symbolic representation);
- express the desired properties of the system in temporal logic.

Putting it All Together

When we design a system, we would like to be sure that it will satisfy all requirements, such as safety.

Now we can treat the safety problem as a mathematical problem. We can

- formally represent our system as a transition system (the symbolic representation);
- express the desired properties of the system in temporal logic.

What is missing?

The Model Checking Problem

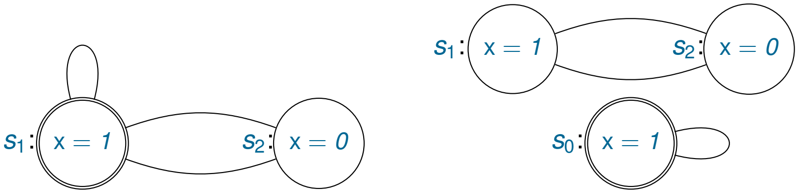
Given

1. a symbolic representation of a transition system;
2. a temporal formula F,

check if every (some) computation of the system satisfies this formula, preferably in a fully automatic way.

Symbolic Representation and Transition Systems

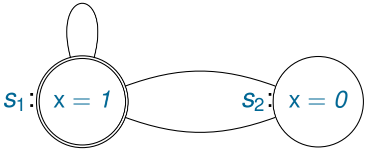
Consider the transition systems with the following state transition graphs:

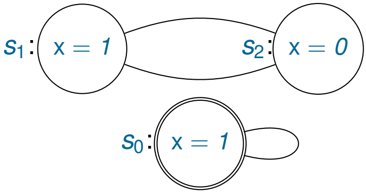


They have the same symbolic representation but satisfy different LTL formulas. For example, $\Diamond \neg x$ is true in the first one but false in the second.

Symbolic Representation and Transition Systems

Consider the transition systems with the following state transition graphs:

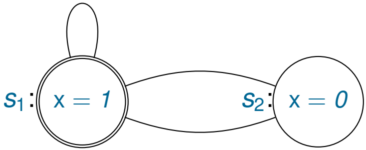


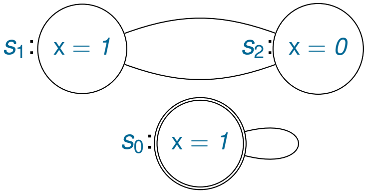


They have the same symbolic representation but satisfy different LTL formulas. For example, $\Diamond\neg x$ is true in the first one but false in the second.
This may happen only if one of the transition systems has more than one different state with the same labelling function (states s_0 and s_1 in the second system).

Symbolic Representation and Transition Systems

Consider the transition systems with the following state transition graphs:





They have the same symbolic representation but satisfy different LTL formulas. For example, $\Diamond\neg x$ is true in the first one but false in the second.

This may happen only if one of the transition systems has more than one different state with the same labelling function (states s_0 and s_1 in the second system).

We call such symbolic representations inadequate: one cannot distinguish two different states by a formula.

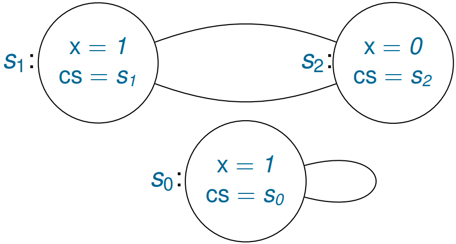
Making an Adequate Representation

If a transition system has different states labeled by the same interpretation, then introduce a new state variable that will distinguish any such pair of states.

Making an Adequate Representation

If a transition system has different states labeled by the same interpretation, then introduce a new state variable that will distinguish any such pair of states.

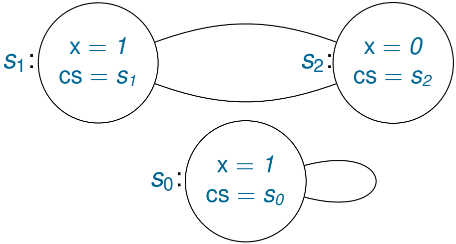
For example, one can add a variable cs (current state) ranging over all states such the value of cs at a state s is s.



Making an Adequate Representation

If a transition system has different states labeled by the same interpretation, then introduce a new state variable that will distinguish any such pair of states.

For example, one can add a variable cs (current state) ranging over all states such the value of cs at a state s is s.



We assume that different states always have different labellings.

Reachability and Safety Properties

A reachability property is expressed by a formula

$$\Diamond F,$$

where F is a propositional formula.

Reachability and Safety Properties

A reachability property is expressed by a formula

$$\Diamond F,$$

where F is a propositional formula.

A safety property is expressed by a formula

$$\Box F,$$

where F is a propositional formula.

Reachability and Safety Properties

A reachability property is expressed by a formula

$$\Diamond F,$$

where F is a propositional formula.

A safety property is expressed by a formula

$$\Box F,$$

where F is a propositional formula.

Reachability and safety properties are the most common problems arising in model checking. They are dual to each other: if we can check one of them, we can check the other one too:

- $\Box F \equiv \neg \Diamond \neg F$;
- $\Diamond F \equiv \neg \Box \neg F$.

Reachability and Safety Properties

A reachability property is expressed by a formula

$$\Diamond F,$$

where F is a propositional formula.

A safety property is expressed by a formula

$$\Box F,$$

where F is a propositional formula.

Reachability and safety properties are the most common problems arising in model checking. They are dual to each other: if we can check one of them, we can check the other one too:

- $\Box F \equiv \neg\Diamond\neg F$;
- $\Diamond F \equiv \neg\Box\neg F$.

We cannot reach an unsafe state if and only if all states we can visit are safe.

Reachability

Fix a transition system $\mathbb{S}$ with the transition relation T. We write $s_0 \to s_1$ for $(s_0, s_1) \in T$ (that is, if there is a transition from s_0 to s_1).

Reachability

Fix a transition system $\mathbb{S}$ with the transition relation T. We write $s_0 \to s_1$ for $(s_0, s_1) \in T$ (that is, if there is a transition from s_0 to s_1).

- A state s is reachable in n steps from a state s_0 if there exists a sequence of states $s_1, \ldots, s_n$ such that $s_n = s$ and

$$s_0 \to s_1 \to \ldots \to s_n.$$

Reachability

Fix a transition system $\mathbb{S}$ with the transition relation T. We write $s_0 \to s_1$ for $(s_0, s_1) \in T$ (that is, if there is a transition from s_0 to s_1).

- A state s is reachable in n steps from a state s_0 if there exists a sequence of states $s_1, \ldots, s_n$ such that $s_n = s$ and

$$s_0 \to s_1 \to \ldots \to s_n.$$

- A state s is reachable from a state s_0 if s is reachable from s_0 in $n \geq 0$ steps.

Reachability Properties and Graph Reachability

Theorem. Let F be a propositional formula. The formula $\Diamond F$ holds on some computation path if and only if there exists an initial state s_0 and a state s such that $s \models F$ and s is reachable from s_0.

Reformulation of Reachability

Given

1. Initial condition I representing a set of initial states;
2. Final condition F representing a set of final states;
3. formula Tr representing the transition relation of a transition system $\mathbb{S}$,

is any final state reachable from an initial state in $\mathbb{S}$?

Reformulation of Reachability

Given

1. Initial condition I representing a set of initial states;
2. Final condition F representing a set of final states;
3. formula Tr representing the transition relation of a transition system $\mathbb{S}$,

is any final state reachable from an initial state in $\mathbb{S}$?

An interesting property of this reformulation is that it does not use temporal logic.

Symbolic Reachability Checking

- Idea: build a symbolic representation of the set of reachable states.

Symbolic Reachability Checking

- Idea: build a symbolic representation of the set of reachable states.
- Two main kinds of algorithm:
  - forward reachability;
  - backward reachability.

Reformulation as a Decision Problem

Given

1. a formula $I(\bar{x})$, called the initial condition;
2. a formula $F(\bar{x})$, called the final condition;
3. formula $T(\bar{x}, \bar{x}')$, called the transition formula

does there exist a sequence of states $s_0, \ldots, s_n$ such that

1. $s_0 \models I(\bar{x})$;
2. $s_n \models F(\bar{x})$;
3. For all $i = 0, \ldots, n-1$ we have $(s_{i-1}, s_i) \models T(\bar{x}, \bar{x}')$.

Note that in this case s_n is reachable from s_0 in n steps.

Idea of Reachability-Checking Algorithms

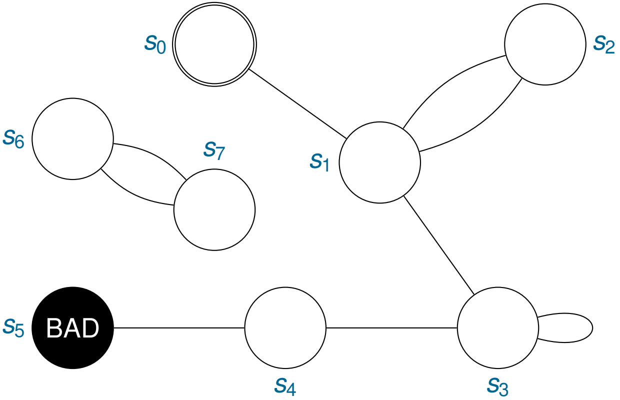
If a final state is reachable from an initial state, then it is reachable from an initial state in some number n of steps.

Idea of Reachability-Checking Algorithms

If a final state is reachable from an initial state, then it is reachable from an initial state in some number n of steps.

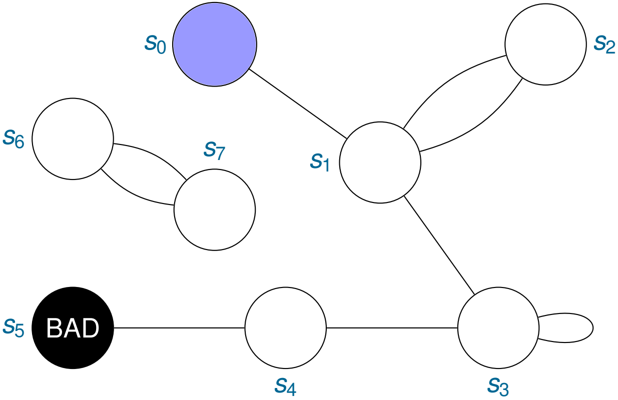
For a given number n, find a symbolic representation of the set of states reachable from from an initial state in n steps. If this formula is not satisfied in a final state, increase n and start again.

Reachability in n steps



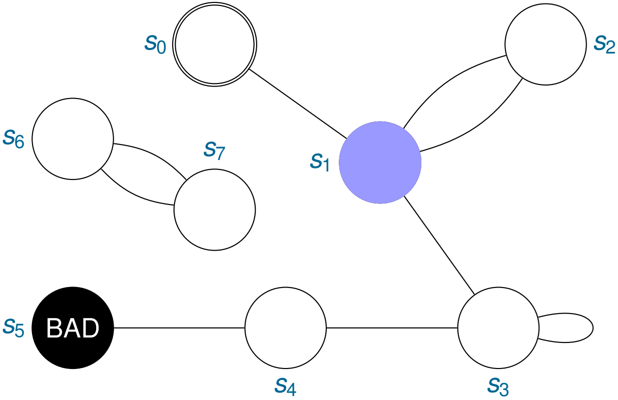
Reachability in n steps

Number of steps: 0



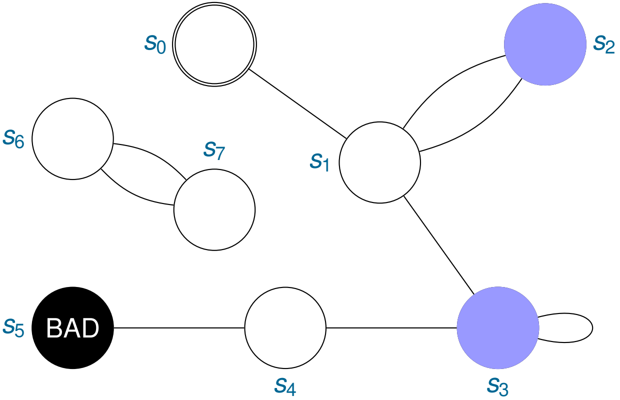
Reachability in n steps

Number of steps: 1



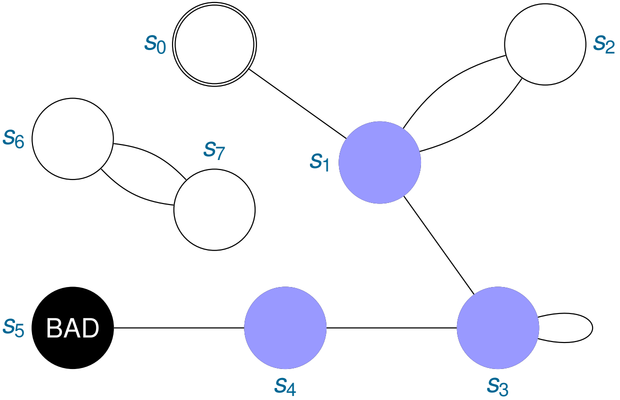
Reachability in n steps

Number of steps: 2



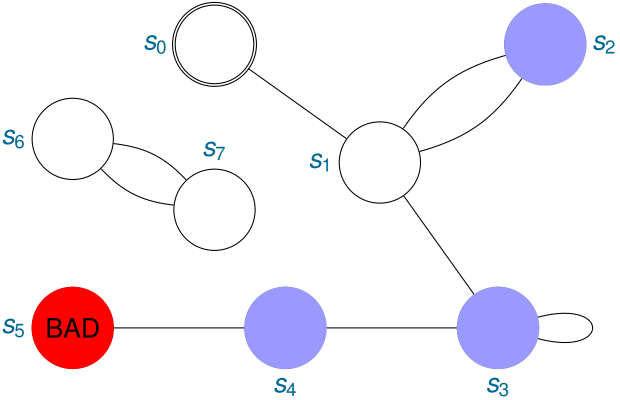
Reachability in n steps

Number of steps: 3



Reachability in n steps

Number of steps: 4



Simple Logical Analysis

Lemma

Let $C(\bar{x})$ symbolically represent a set of states S. Define

$$FR(\bar{x}) \stackrel{\text{def}}{=} \exists \bar{x}_1 (C(\bar{x}_1) \wedge T(\bar{x}_1, \bar{x})).$$

Then $FR(\bar{x})$ represents the set of states reachable from S in one step.

Simple Logical Analysis

Lemma

Let $C(\bar{x})$ symbolically represent a set of states S. Define

$$FR(\bar{x}) \stackrel{\text{def}}{=} \exists \bar{x}_1 (C(\bar{x}_1) \wedge T(\bar{x}_1, \bar{x})).$$

Then $FR(\bar{x})$ represents the set of states reachable from S in one step.

Define a sequence of formulas R_n for reachability in n states:

$$\begin{aligned} R_0(\bar{x}) &\stackrel{\text{def}}{=} I(\bar{x}) \\ R_{n+1}(\bar{x}) &\stackrel{\text{def}}{=} \exists \bar{x}_1 (R_n(\bar{x}_1) \wedge T(\bar{x}, \bar{x}_1)) \end{aligned}$$

End of Lecture 21

Slides for lecture 21 end here . . .

Reachability in n Steps Using SAT

Let $n \geq 0$ and $\bar{x}$ be state variables. Let

1. $I(\bar{x})$ the symbolic representation of the set of initial states;
2. $T(\bar{x}, \bar{x}')$ the symbolic representation of the transition relation;
3. $F(\bar{x})$ be a propositional formula of this variables;

Then a state satisfying $F(\bar{x})$ is reachable in n steps if and only if the following propositional formula is satisfiable:

$$I(\bar{x}_0) \wedge T(\bar{x}_0, \bar{x}_1) \wedge \ldots \wedge T(\bar{x}_{n-1}, \bar{x}_n) \wedge F(\bar{x}_n).$$

Reachability in n Steps Using SAT

Let $n \geq 0$ and $\bar{x}$ be state variables. Let

1. $I(\bar{x})$ the symbolic representation of the set of initial states;
2. $T(\bar{x}, \bar{x}')$ the symbolic representation of the transition relation;
3. $F(\bar{x})$ be a propositional formula of this variables;

Then a state satisfying $F(\bar{x})$ is reachable in n steps if and only if the following propositional formula is satisfiable:

$$I(\bar{x}_0) \land T(\bar{x}_0, \bar{x}_1) \land \ldots \land T(\bar{x}_{n-1}, \bar{x}_n) \land F(\bar{x}_n).$$

Further, take any satisfying assignment $\{\bar{x}_0 \mapsto \bar{v}_0, \ldots, \bar{x}_n \mapsto \bar{v}_n\}$ for this formula and define states $s_0, \ldots, s_n$ by $s_i \stackrel{\text{def}}{=} \{\bar{x} \mapsto \bar{v}_i\}$. Then we have that $s_0 \models I(\bar{x})$, $s_n \models F(\bar{x})$ and

$$s_0 \to s_1 \to \ldots \to s_{n-1} \to s_n$$

In other words, solutions to the formula define paths leading from an initial state to a state satisfying $F(\bar{x})$.

Simple Forward Reachability Algorithm

procedure $FReach(I, T, F)$
input: formulas I, T, F
output: "yes" or no output
begin
  $i := 0$
  $R := I(\bar{x}_0)$;
  loop
    if $R \wedge F(\bar{x}_i)$ is satisfiable **then** **return** "yes" ;
    $R := R \wedge T(\bar{x}_i, \bar{x}_{i+1})$;
    $i := i + 1$
  end **loop**
end

Simple Forward Reachability Algorithm

```
procedure FReach(I, T, F)
input: formulas I, T, F
output: "yes" or no output
begin
  i := 0
  R := I(x̄_0) ;
  loop
    if R ∧ F(x̄_i) is satisfiable then return "yes" ;
    R := R ∧ T(x̄_i, x̄_{i+1}) ;
    i := i + 1
  end loop
end
```

Implementation?

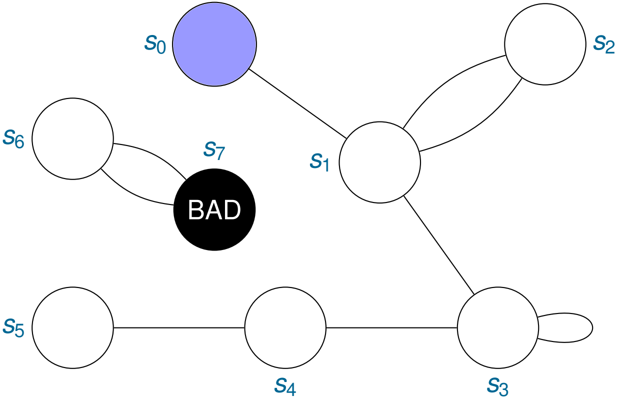
Simple Forward Reachability Algorithm

procedure $FReach(I, T, F)$
input: formulas I, T, F
output: "yes" or no output
begin
  $i := 0$
  $R := I(\bar{x}_0)$;
  loop
    if $R \wedge F(\bar{x}_i)$ is satisfiable **then** **return** "yes" ;
    $R := R \wedge T(\bar{x}_i, \bar{x}_{i+1})$;
    $i := i + 1$
  end **loop**
end

Implementation?
Use SAT solvers.

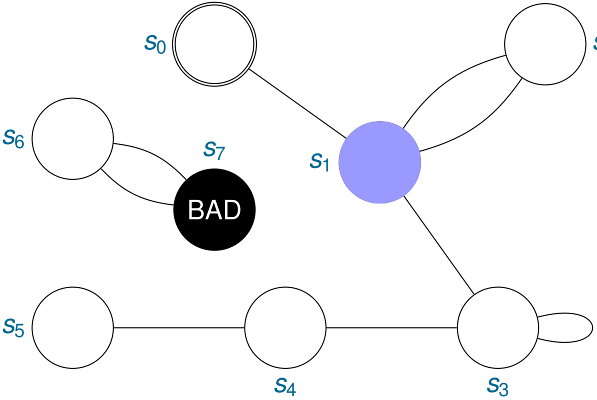
Termination?

Number of steps: 0



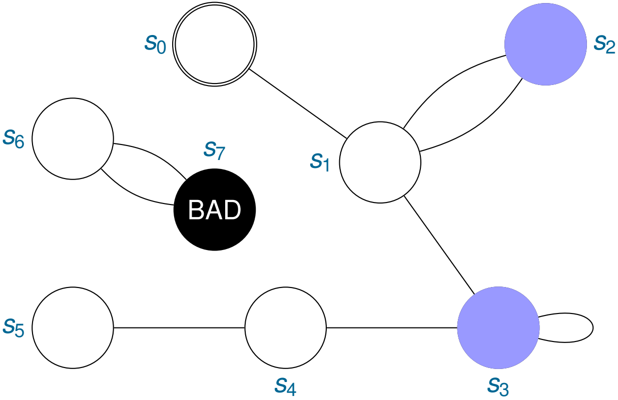
Termination?

Number of steps: 1



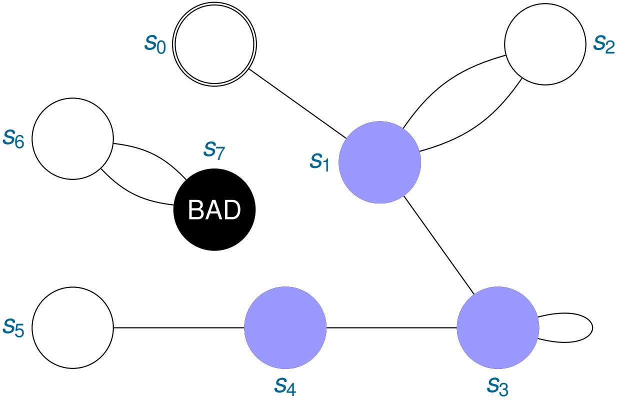
Termination?

Number of steps: 2



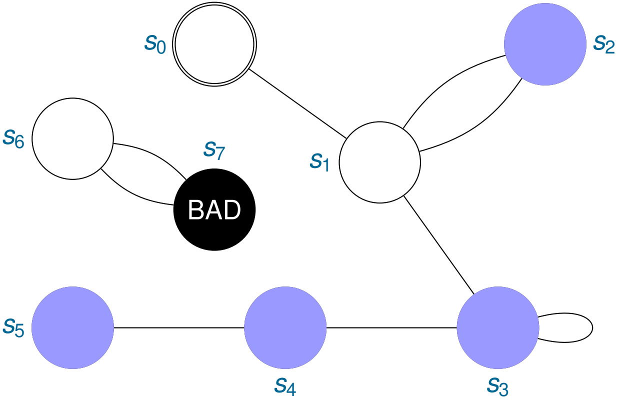
Termination?

Number of steps: 3



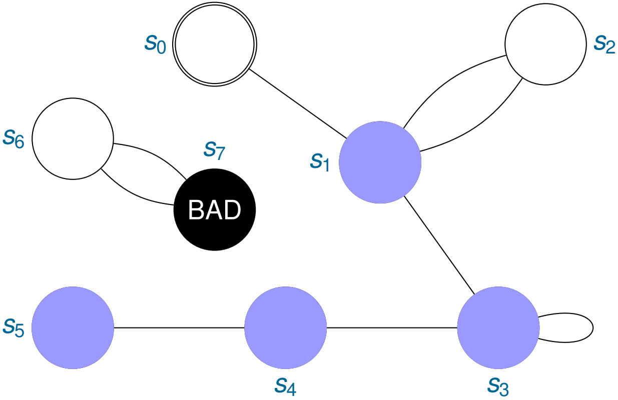
Termination?

Number of steps: 4



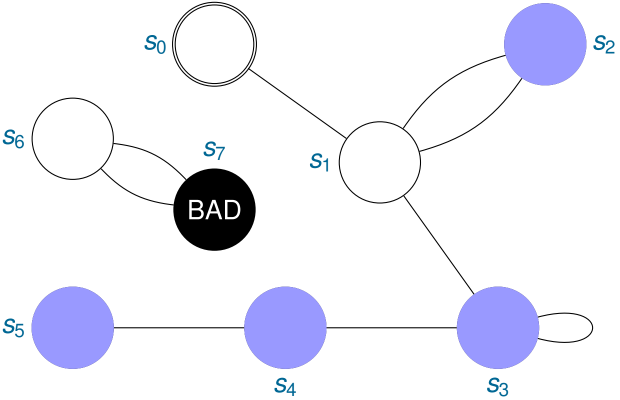
Termination?

Number of steps: 5



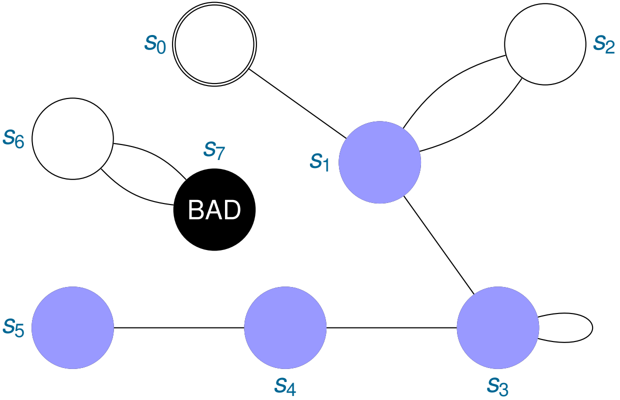
Termination?

Number of steps: 6



Termination?

Number of steps: 7



When no final state is reachable, the algorithm does not terminate.

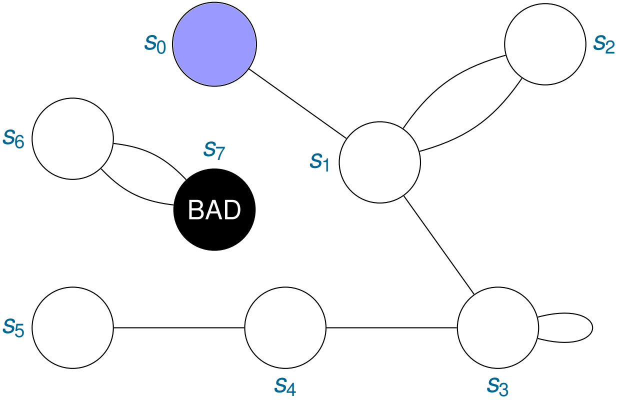
Reachability in $\leq n$ steps

Define a sequence of formulas $R_{\leq n}$ for reachability in $\leq n$ states:

$$
\begin{array}{rcl}
R_{\leq 0}(\bar{x}) & \stackrel{\text{def}}{=} & I(\bar{x}) \\
R_{\leq n+1}(\bar{x}) & \stackrel{\text{def}}{=} & R_{\leq n}(\bar{x}) \vee \exists \bar{x}_1 (R_{\leq n}(\bar{x}_1) \wedge T(\bar{x}, \bar{x}_1))
\end{array}
$$

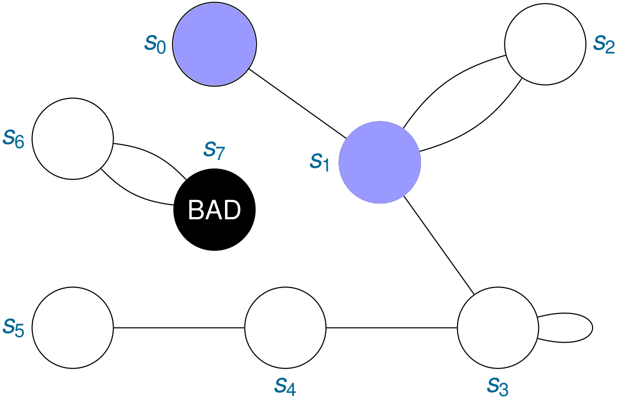
Reachability in $\leq n$ steps

Number of steps: 0



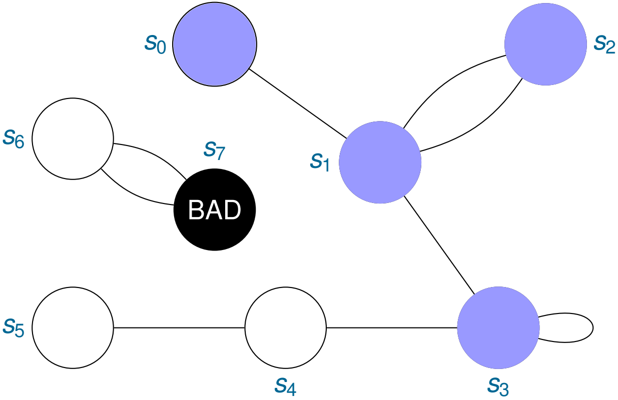
Reachability in $\leq n$ steps

Number of steps: 1



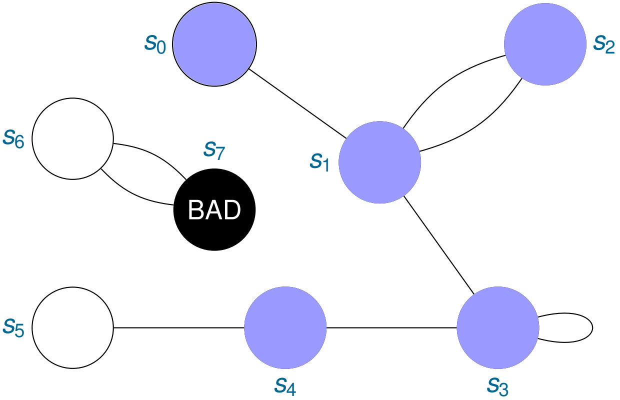
Reachability in $\leq n$ steps

Number of steps: 2



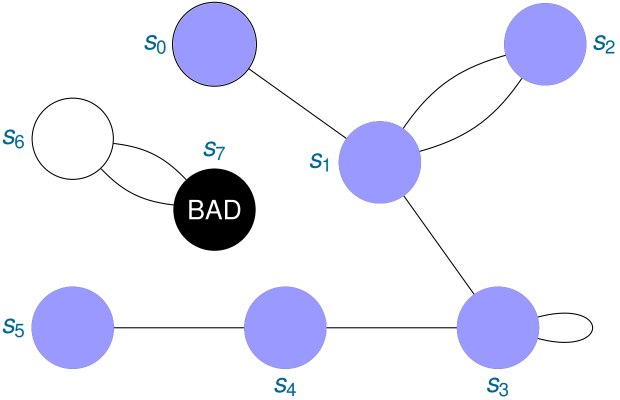
Reachability in $\leq n$ steps

Number of steps: 3



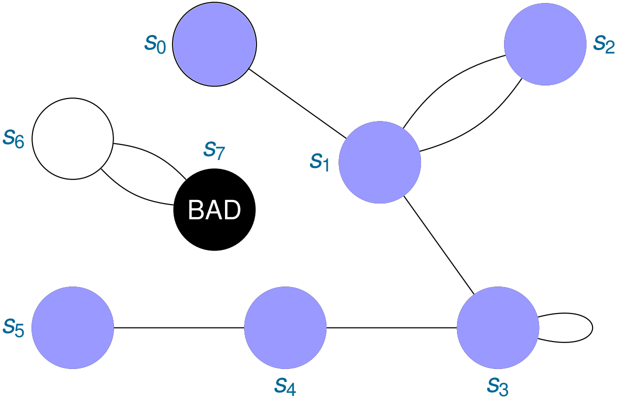
Reachability in $\leq n$ steps

Number of steps: 4



Reachability in $\leq n$ steps

Number of steps: 5



The set of states will change no more.

Termination

Denote by S_n the set of states reachable from an initial state in $\leq n$ steps.
Key properties for termination.

- $S_i \subseteq S_{i+1}$ for all i;
- the system has a finite number of states;
- therefore, there exists a number k such that $S_k = S_{k+1}$;
- for such k we have $R_{\leq k}(\bar{x}) \equiv R_{\leq k+1}(\bar{x})$.

Forward Reachability Algorithm

procedure $FReach(I, T, F)$
input: formulas I, T, F
output: "yes" or "no"
begin
  $R(\bar{x}) \; := \; I(\bar{x})$;
  loop
    if $R(\bar{x}) \wedge F(\bar{x})$ is satisfiable **then** **return** "yes" ;
    $R'(\bar{x}) \; := \; R(\bar{x}) \vee \exists \bar{x}_1 (R(\bar{x}_1) \wedge T(\bar{x}_1, \bar{x}))$;
    if $R(\bar{x}) \equiv R'(\bar{x})$ **then** **return** "no" ;
    $R(\bar{x}) \; := \; R'(\bar{x})$
  end **loop**
end

Forward Reachability Algorithm

procedure *FReach*(I, T, F)
input: formulas I, T, F
output: "yes" or "no"
begin
  $R(\bar{x}) \ := \ I(\bar{x})$;
  loop
    if $R(\bar{x}) \wedge F(\bar{x})$ is satisfiable **then** **return** "yes" ;
    $R'(\bar{x}) \ := \ R(\bar{x}) \vee \exists \bar{x}_1 (R(\bar{x}_1) \wedge T(\bar{x}_1, \bar{x}))$;
    if $R(\bar{x}) \equiv R'(\bar{x})$ **then** **return** "no" ;
    $R(\bar{x}) \ := \ R'(\bar{x})$
  end **loop**
end

Implementation?

Forward Reachability Algorithm

procedure $FReach(I, T, F)$
input: formulas I, T, F
output: "yes" or "no"
begin
  $R(\bar{x}) \quad := \quad I(\bar{x})$;
  loop
    if $R(\bar{x}) \wedge F(\bar{x})$ is satisfiable **then** **return** "yes" ;
    $R'(\bar{x}) \quad := \quad R(\bar{x}) \vee \exists \bar{x}_1 (R(\bar{x}_1) \wedge T(\bar{x}_1, \bar{x}))$;
    if $R(\bar{x}) \equiv R'(\bar{x})$ **then** **return** "no" ;
    $R(\bar{x}) \quad := \quad R'(\bar{x})$
  end **loop**
end

Implementation?

Conjunction and disjunction

Forward Reachability Algorithm

procedure *FReach*(I, T, F)
input: formulas I, T, F
output: "yes" or "no"
begin
  $R(\bar{x}) \ := \ I(\bar{x})$;
  loop
    if $R(\bar{x}) \wedge F(\bar{x})$ is satisfiable **then** **return** "yes" ;
    $R'(\bar{x}) \ := \ R(\bar{x}) \vee \exists \bar{x}_1 (R(\bar{x}_1) \wedge T(\bar{x}_1, \bar{x}))$;
    if $R(\bar{x}) \equiv R'(\bar{x})$ **then** **return** "no" ;
    $R(\bar{x}) \ := \ R'(\bar{x})$
  end **loop**
end

Implementation?

Conjunction and disjunction
Quantification

Forward Reachability Algorithm

procedure $FReach(I, T, F)$
input: formulas I, T, F
output: "yes" or "no"
begin
  $R(\bar{x}) \ := \ I(\bar{x})$;
  loop
    if $R(\bar{x}) \wedge F(\bar{x})$ is satisfiable **then** **return** "yes" ;
    $R'(\bar{x}) \ := \ R(\bar{x}) \vee \exists \bar{x}_1 (R(\bar{x}_1) \wedge T(\bar{x}_1, \bar{x}))$;
    if $R(\bar{x}) \equiv R'(\bar{x})$ **then** **return** "no" ;
    $R(\bar{x}) \ := \ R'(\bar{x})$
  end loop
end

Implementation?

Conjunction and disjunction
Quantification
Satisfiability checking

Forward Reachability Algorithm

procedure $FReach(I, T, F)$
input: formulas I, T, F
output: "yes" or "no"
begin
$R(\bar{x}) \ := \ I(\bar{x})$;
loop
if $R(\bar{x}) \wedge F(\bar{x})$ is satisfiable **then** **return** "yes" ;
$R'(\bar{x}) \ := \ R(\bar{x}) \vee \exists \bar{x}_1 (R(\bar{x}_1) \wedge T(\bar{x}_1, \bar{x}))$;
if $R(\bar{x}) \equiv R'(\bar{x})$ **then** **return** "no" ;
$R(\bar{x}) \ := \ R'(\bar{x})$
end **loop**
end

Implementation?

Conjunction and disjunction
Quantification
Satisfiability checking
Equivalence checking

Forward Reachability Algorithm

procedure $FReach(I, T, F)$
input: formulas I, T, F
output: "yes" or "no"
begin
  $R(\bar{x}) \ := \ I(\bar{x})$;
  loop
    if $R(\bar{x}) \wedge F(\bar{x})$ is satisfiable **then** **return** "yes" ;
    $R'(\bar{x}) \ := \ R(\bar{x}) \vee \exists \bar{x}_1 (R(\bar{x}_1) \wedge T(\bar{x}_1, \bar{x}))$;
    if $R(\bar{x}) \equiv R'(\bar{x})$ **then** **return** "no" ;
    $R(\bar{x}) \ := \ R'(\bar{x})$
  end **loop**
end

Implementation?
Use OBDDs and OBDD algorithms

Conjunction and disjunction
Quantification
Satisfiability checking
Equivalence checking

Main Problems with the Forward Reachability Algorithms

Forward reachability behave in the same way independently of the set of final states.

In other words, they are not goal oriented.

Backward Reachability

Idea:

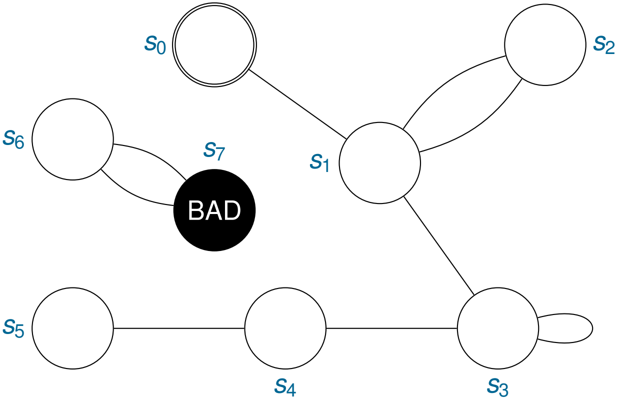
- instead of going forward in the state transition graph, go backward;
- swap initial and final states and invert the transition relation.

Backward Reachability in $\leq n$ steps

Idea:

- instead of going forward in the state transition graph, go backward;
- swap initial and final states and invert the transition relation.

Number of backward steps: 0

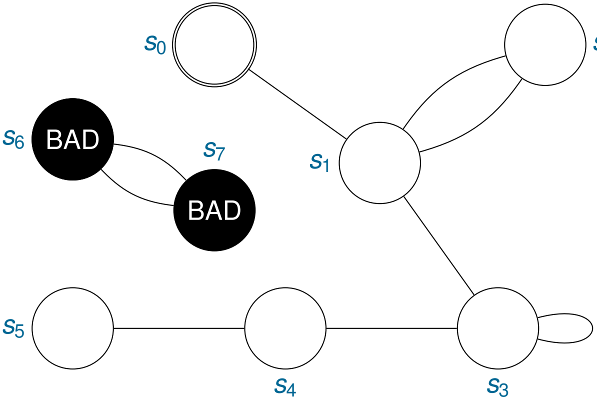


Backward Reachability in $\leq n$ steps

Idea:

- instead of going forward in the state transition graph, go backward;
- swap initial and final states and invert the transition relation.

Number of backward steps: 1

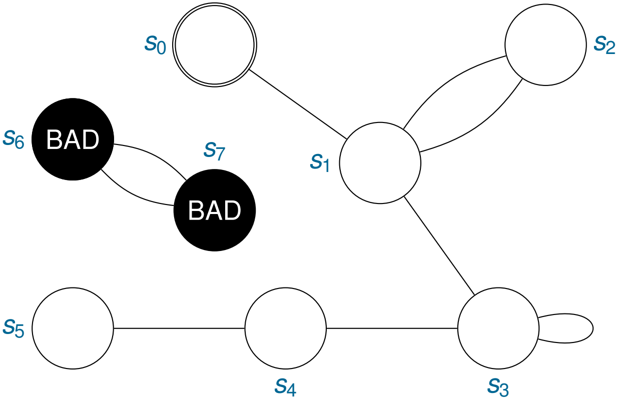


Backward Reachability in $\leq n$ steps

Idea:

- instead of going forward in the state transition graph, go backward;
- swap initial and final states and invert the transition relation.

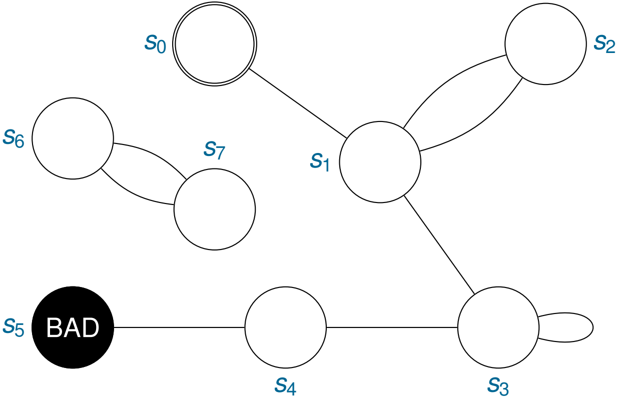
Number of backward steps: 1



Unreachable!

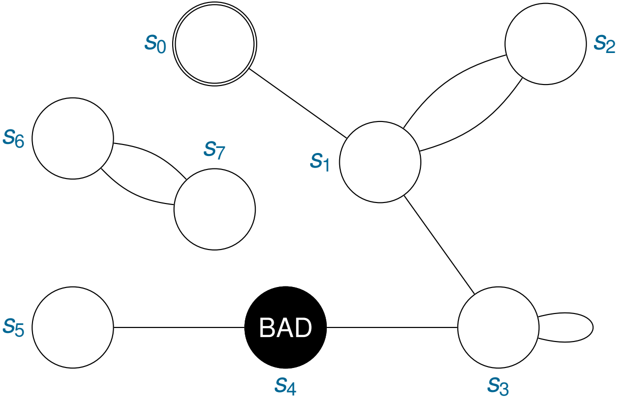
Backward Reachability in *n* steps

Number of backward steps: 0



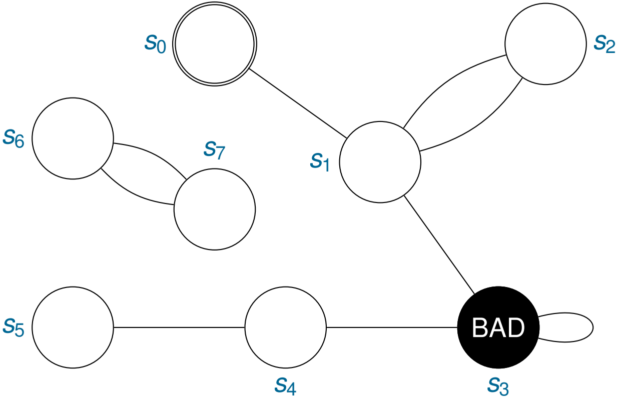
Backward Reachability in n steps

Number of backward steps: 1



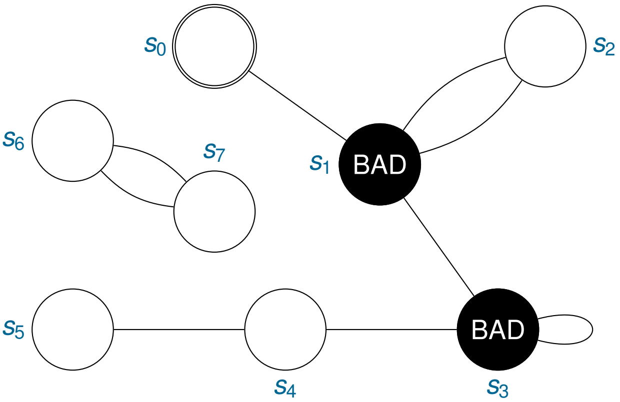
Backward Reachability in *n* steps

Number of backward steps: 2



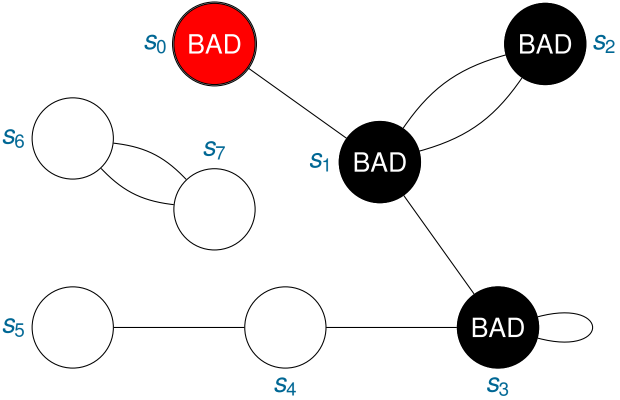
Backward Reachability in *n* steps

Number of backward steps: 3



Backward Reachability in n steps

Number of backward steps: 4



Reachable!

Backward Reachability

If S_n is reachable from S_0 in n steps, we say that S_0 is backward reachable from S_0 in n steps.

Backward Reachability

If S_n is reachable from S_0 in n steps, we say that S_0 is backward reachable from S_0 in n steps.

Lemma

Let $C(\bar{x})$ symbolically represent a set of states S. Define

$$BR(\bar{x}) \stackrel{\text{def}}{=} \exists \bar{x}_1 (C(\bar{x}_1) \wedge T(\bar{x}, \bar{x}_1)).$$

Then $BR(\bar{x})$ represents the set of states backward reachable from S in one step.

Backward Reachability Algorithm

Same as the forward reachability algorithms, but

- Swap I with F;
- Use the inverse of the transition relation T.

Backward Reachability Algorithm

Same as the forward reachability algorithms, but

- Swap I with F;
- Use the inverse of the transition relation T.

procedure $BReach(I, T, F)$
input: formulas I, T, F
output: "yes" or "no"
begin
  $R(\bar{x}) \ := \ F(\bar{x})$;
  loop
    if $R(\bar{x}) \wedge I(\bar{x})$ is satisfiable **then** **return** "yes" ;
    $R'(\bar{x}) \ := \ R(\bar{x}) \vee \exists \bar{x}_1 (R(\bar{x}_1) \wedge T(\bar{x}, \bar{x}_1))$;
    if $R(\bar{x}) \equiv R'(\bar{x})$ **then** **return** "no" ;
    $R(\bar{x}) \ := \ R'(\bar{x})$
  end **loop**
end

Other Properties

- There are model-checking algorithms for properties other than reachability;

Other Properties

- There are model-checking algorithms for properties other than reachability;
- there is even a general model-checking algorithm for arbitrary LTL properties;

Other Properties

- There are model-checking algorithms for properties other than reachability;
- there is even a general model-checking algorithm for arbitrary LTL properties;
- these algorithms will not be considered in this course;

End of Lecture 22

Slides for lecture 22 end here . . .