

Reachability

Fix a transition system \mathcal{S} with the transition relation T . We write $s_0 \rightarrow s_1$ 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, \dots, s_n such that $s_n = s$ and

$$s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_n.$$

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

Reachability

Fix a transition system \mathcal{S} with the transition relation T . We write $s_0 \rightarrow s_1$ 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, \dots, s_n such that $s_n = s$ and

$$s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_n.$$

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

Reachability

Fix a transition system \mathcal{S} with the transition relation T . We write $s_0 \rightarrow s_1$ 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, \dots, s_n such that $s_n = s$ and

$$s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_n.$$

- ▶ 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 In representing a set of initial states I ;
2. Final condition Fin representing a set of final states F ;
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.

Reformulation of reachability

Given

1. Initial condition In representing a set of initial states I ;
2. Final condition Fin representing a set of final states F ;
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.
- ▶ Two main kinds of algorithm:
 - ▶ forward reachability;
 - ▶ backward reachability.

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, \dots, s_n such that

1. $s_0 \models I(\bar{x})$;
2. $s_n \models F(\bar{x})$;
3. For all $i = 0, \dots, n - 1$ 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.

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.

Termination?

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.

Termination?

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.

Termination?

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.

Using this observation, we can define a sequence of formulas $R_{\leq n}$ for reachability in $\leq n$ states:

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

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.

Using this observation, we can define a sequence of formulas $R_{\leq n}$ for reachability in $\leq n$ states:

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

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

Implementation?

Conjunction and disjunction

Quantification

Equality/inequality

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?

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?

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?

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?

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

Incomplete 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?

Incomplete 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?

Incomplete 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.

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

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

procedure $BReach(I, T, F)$

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” ;

$R'(\bar{x}) := R(\bar{x}) \vee \exists \bar{x}_1 (R(\bar{x}_1) \wedge T(\bar{x}, \bar{x}_1))$;

if $R(\bar{x}) \equiv R'(\bar{x})$ then return “no” ;

$R(\bar{x}) := R'(\bar{x})$

end loop

end

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

$R(\bar{x}) := I(\bar{x})$;

loop

if $R(\bar{x}) \wedge F(\bar{x})$ is satisfiable **then return** “yes” ;

$R'(\bar{x}) := R(\bar{x}) \vee \exists \bar{x}_1 (R(\bar{x}_1) \wedge T(\bar{x}, \bar{x}_1))$;

if $R(\bar{x}) \equiv R'(\bar{x})$ **then return** “no” ;

$R(\bar{x}) := R'(\bar{x})$

end loop

end