Automated Reasoning

Cynthia Kop & Sebastian Junges
Fall 2024

Lecture 13:
Reasoning about Reachability

Slides inspired by:
Sanjit Seshia, UC Berkeley
Elizabeth Polgreen, Univ Edinburgh

Today's Lecture

- Automated reasoning on graphs:
- (Un)reachability via SAT-solvers
- Symbolic transition systems
- Algorithms for symbolic transition systems:
  - Abstraction
  - Interpolation

Goal:

Learn how SAT and SMT-solvers accelerate reasoning about reachability in graphs

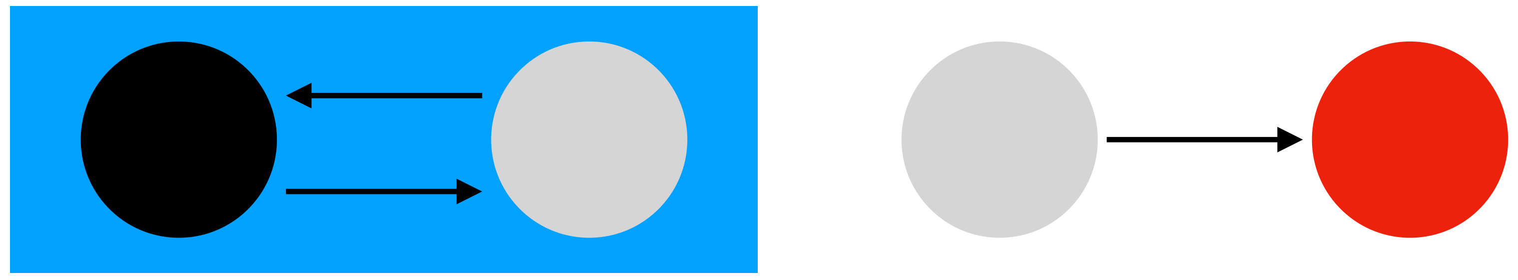
Graphs are everywhere

- Planning problems in robotics, logistics, etc.
- Bug finding (an execution ends in an error-state,
  the course **model checking** treats more complex properties)

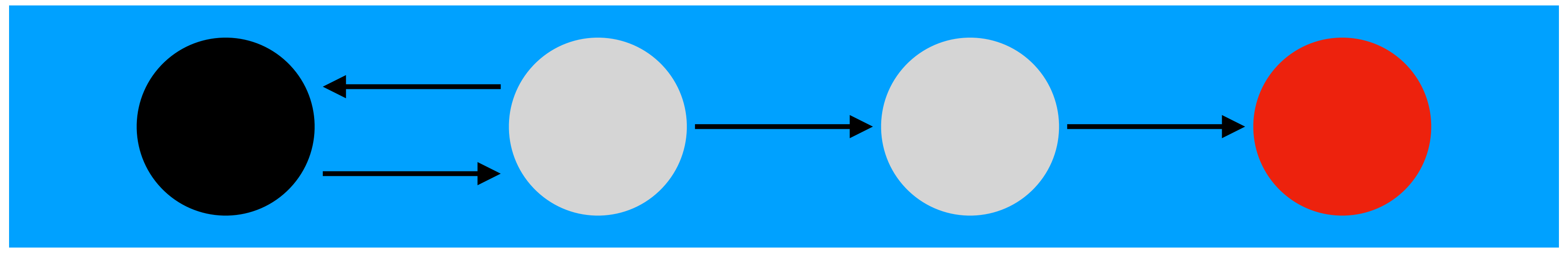
Given a transition system (a directed graph) $\langle S, I, T \rangle$,
is there a **path** from some source state $s \in S$ to a target state $t \in S$?

A path is a sequence of states $s_1 \ldots s_n$ such that $(s_i, s_{i+1}) \in T$ for all $i < n$.

Two properties: (Un)reachability

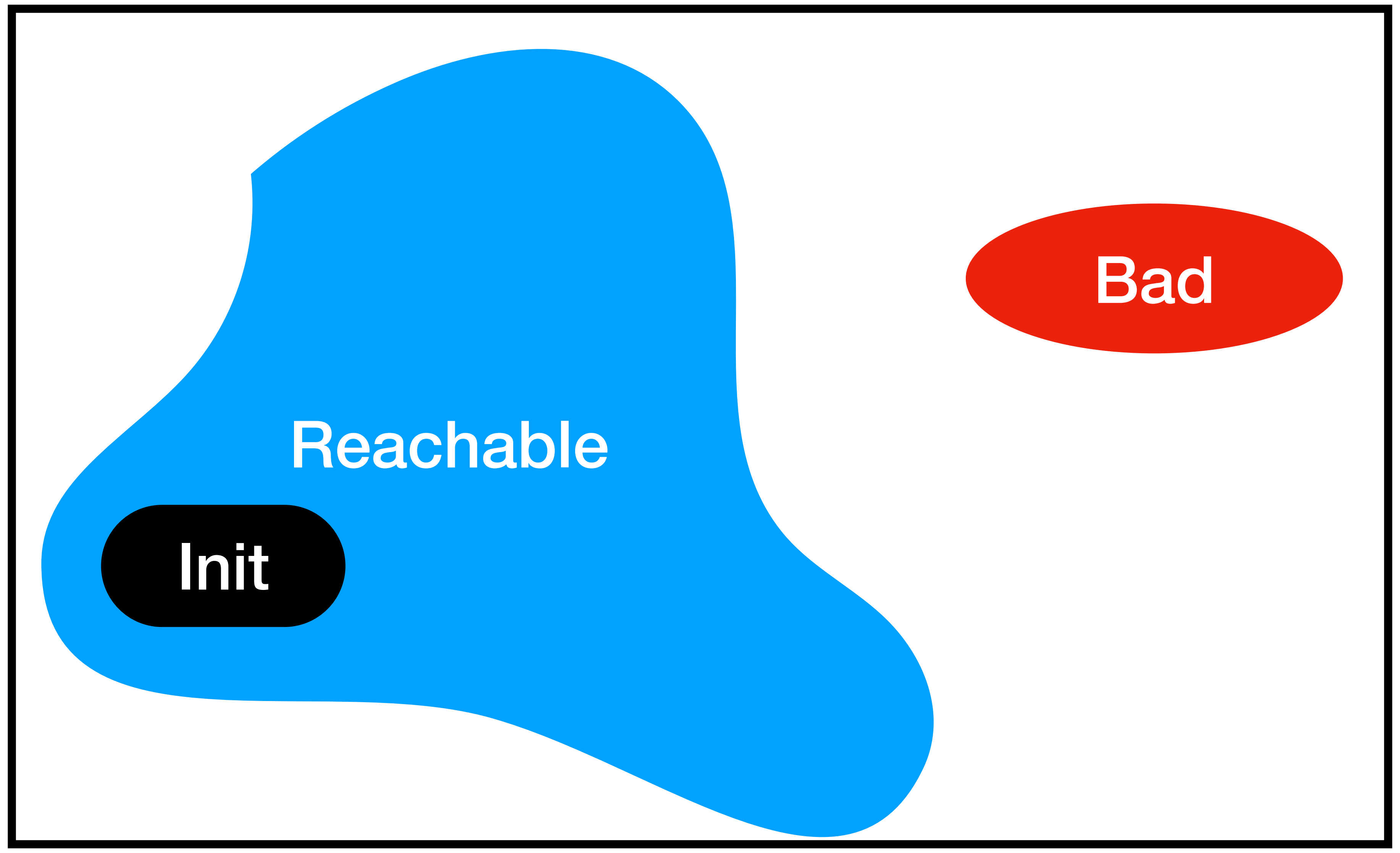


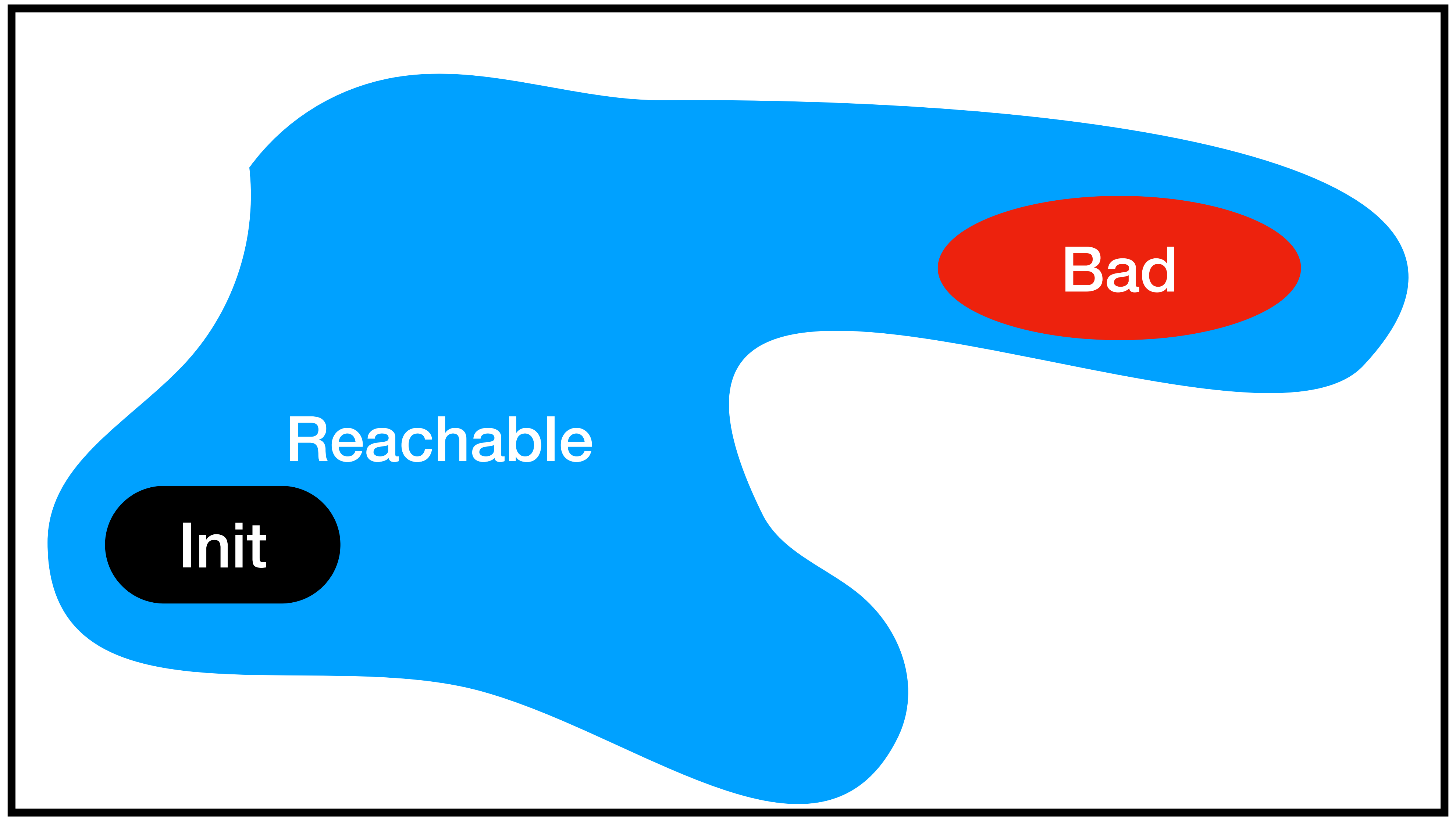
Unreachable



Reachable

A state s is reachable,
iff there exists a path from the initial state to s





Reachability is simple…

- Can be solved via breadth-First or Depth-First Search in $\mathcal{O}(|S| + |T|)$

- What if we have additional constraints
  - Examples: Your homework, building bridges, ….

k-Bounded Reachability

- Is there a path of length up to k between two states?
- One variable per state per time step. Variables: $X = \{x_{s,i} \mid s \in S, 0 \leq i \leq k\}$.
- Idea: Can we reach this state in k steps?
- Constraints:
  - If we can reach state s in i steps, then we can reach the successors in i + 1 steps.
  - We can reach the initial state in 0 steps.
  - We ensure that we reach the target in k steps.

What about unbounded reachability?

- We can set k sufficiently large
  - larger than the number of states,
  - more precisely, larger than the diameter

This quadratic growth in variables is often unacceptable

Notation

Forget about the symbolic aspect for a moment!

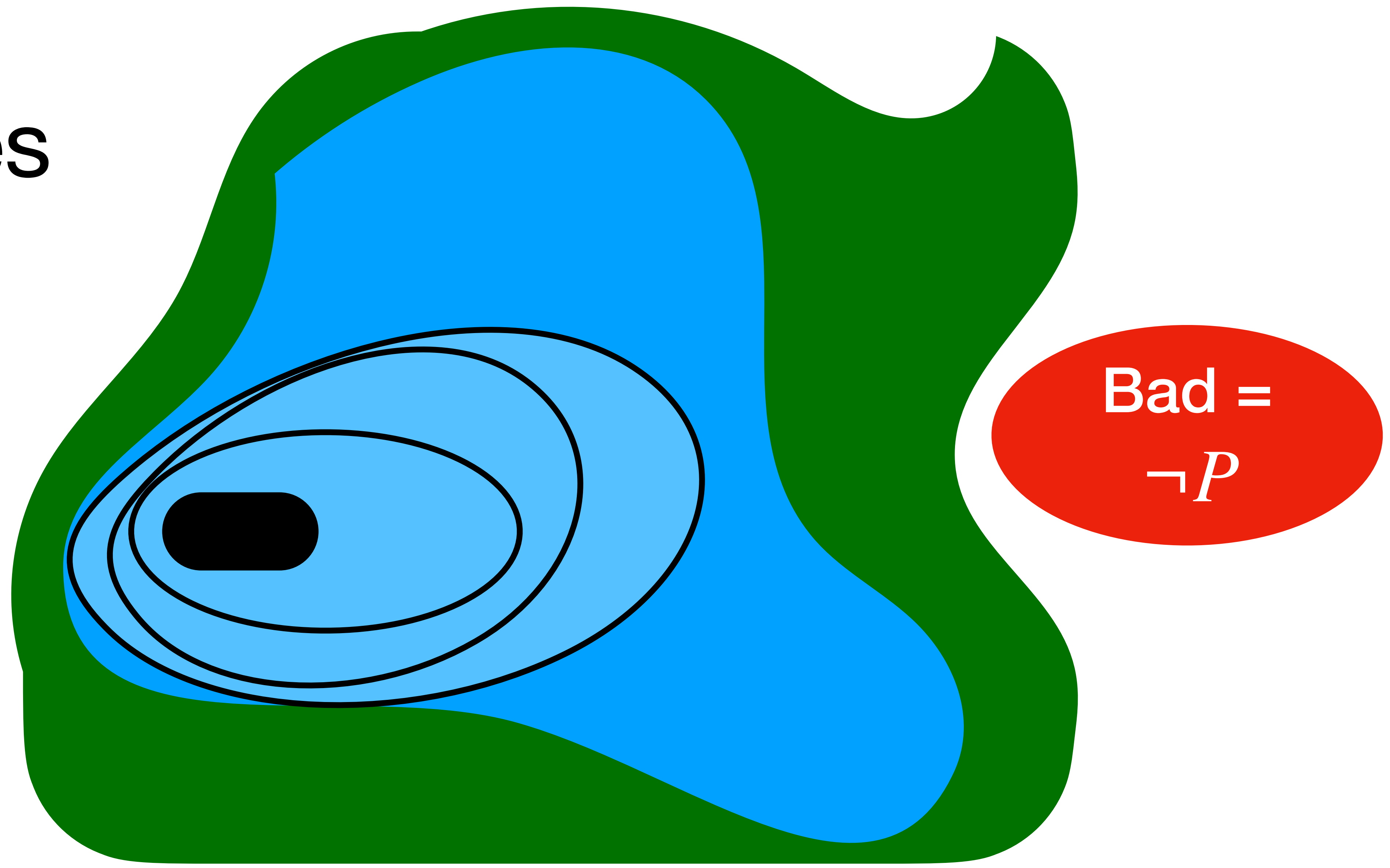
- For any relation $R \in Y \times Y$
  - For $y, y' \in Y$, we write $R(y, y')$ to denote $\langle y, y' \rangle \in R$.
  - For $A \subseteq Y$, we write $R(A)$ to denote $R(A) = \{y' \in Y \mid y \in A \text{ and } R(y, y')\}$.
- Reachable states: $I \cup T(I) \cup T(T(I)) \cup T(\ldots(T(I)))$
- Define $T_+(A) = I \cup A \cup T(A)$
- A Inductive iff $T(A) \subseteq A$

Reachable States as a Fixpoint

- T_+ is an operator on subsets of states
- Reachability is a fixed point of T_+. Which?

Reachable States as a Fixpoint

- T_+ is an operator on subsets of states
- Reachability is a fixed point of T_+. Which?
- Least fixed point! Induces the natural algorithm starting from the initial states
- Any fixed point of T_+ contains all reachable states



Unreachability

- Let the SMT-solver guess (over-approximation of) reachable states
- Prevent including the bad states
- Ensure it is a fixed point to ensure it is an overapproximation
- Variables: $X = \{x_s \mid s \in S\}$.
- Constraints:
  - bad states false, initial states are true
  - If state is in, then also all successors

Idea: Reachability

Guess a subset of the states that can reach the target?

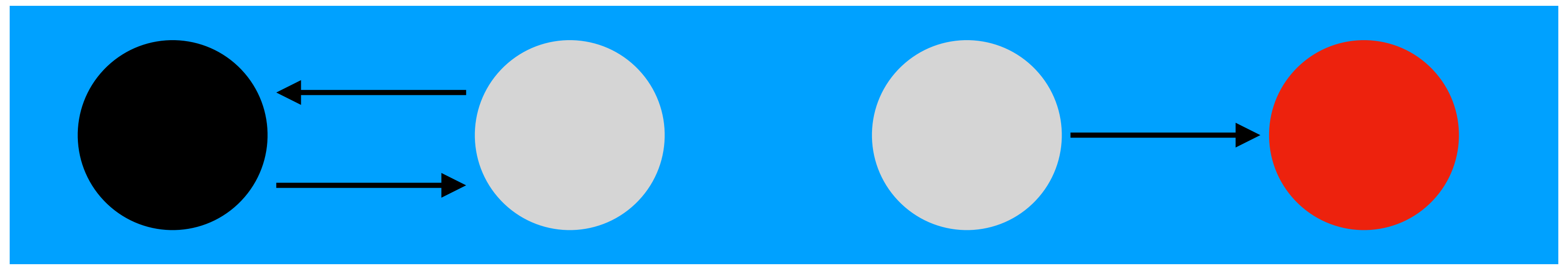
- Benefit: Avoid k copies state variables
- A state can reach the target if the successor can reach the target

  (i.e., the set is closed under the transition relation)
- The target reaches the target

Does not work

- The initial state should reach the target

Unreachable



SAT solution

Idea: Reachability

Fix

- Benefit: Avoid k copies of the transition relation/state variables
- A state can reach the target if a successor can reach the target
  AND is closer to the target
  (i.e., the set is closed under the transition relation)
- The target reaches the target
- The initial state should reach the target

There are some alternatives that all help avoiding cyclic arguments (beyond the scope of this lecture)

Just Reachability is simple…

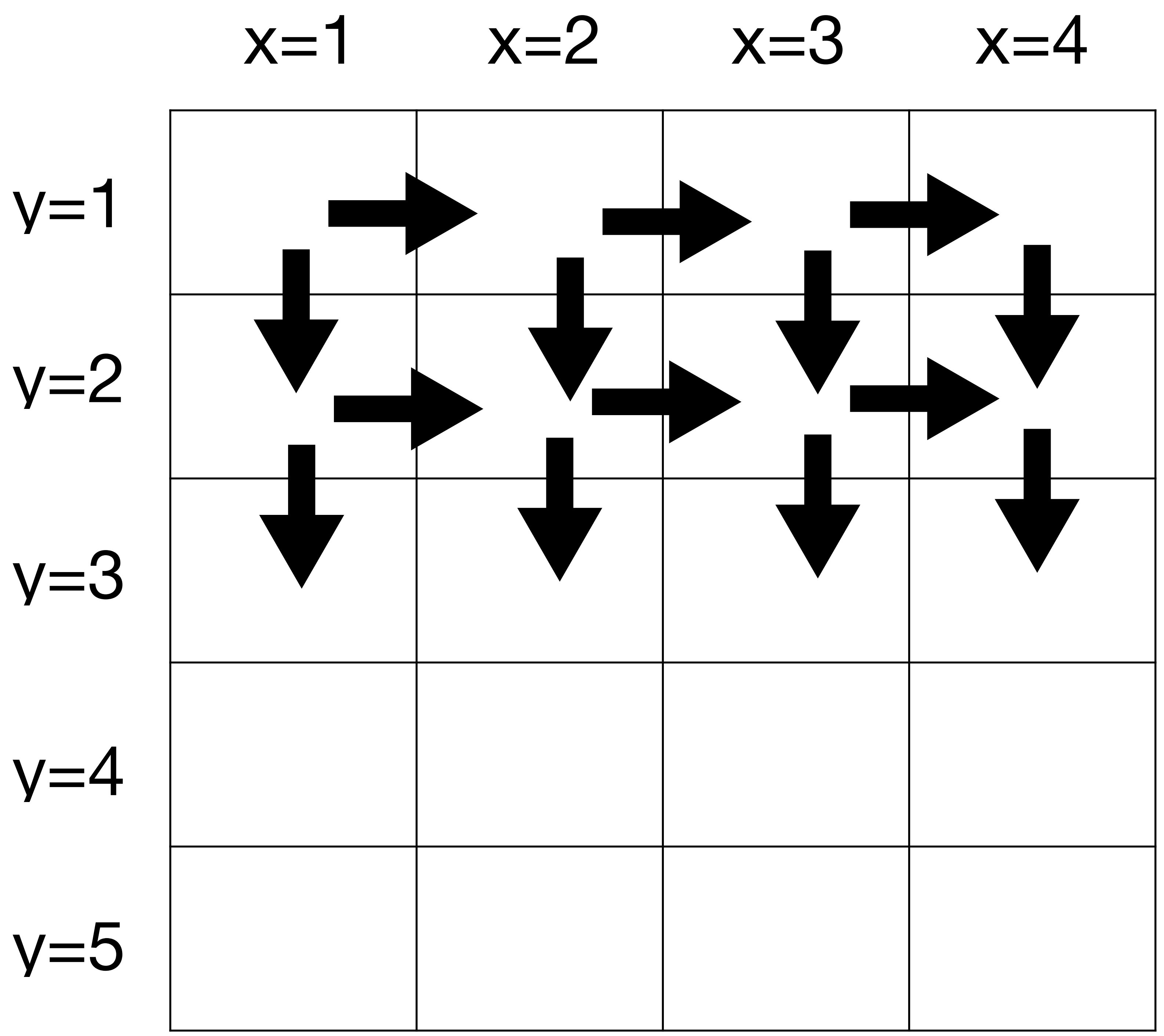
- Breadth-First or Depth-First Search in $\mathcal{O}(|S| + |T|)$
- Let us consider huge graphs…
- Actually, to find a path, it suffices to only guess the correct shortest path
- Complexity: NL (…nondeterministic logspace…)
- Idea: Use an SMT-solver to find such a path

Summary

- Reachability as part of an encoding
- Symbolic transition systems

Symbolically encoding graphs

- Transition systems are typically constructed from a high-level description
- Here: Transition is an assignment to variables integer variables x, y
- Transitions:
  - (only if x < 4) increment x
  - (only if y < 5) increment y
- $T_a = \{(u, v) \mid u(x) < 4 \wedge v(x) = u(x) + 1 \wedge u(y) = v(y)\}$
- $T_b = \{(u, v) \mid u(y) < 5 \wedge v(y) = u(y) + 1 \wedge u(x) = v(x)\}$



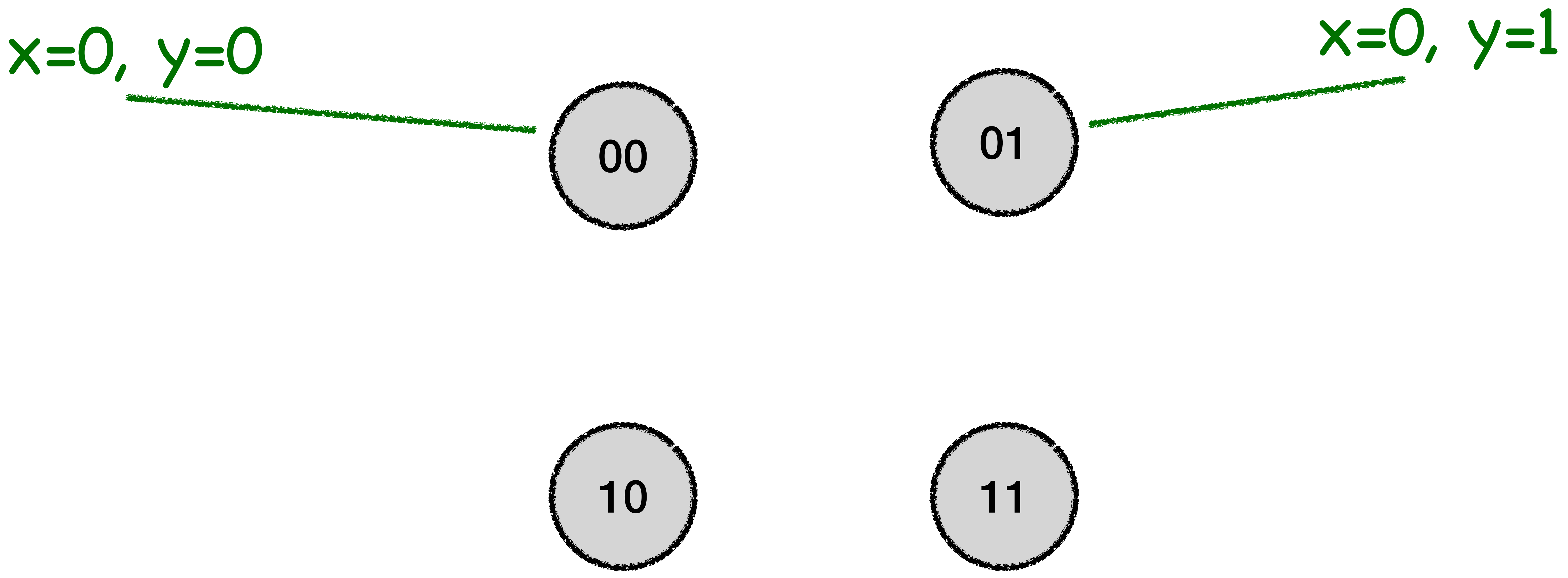
Symbolic Transition Systems

- A set of variables V
- Initial states formula $I(V)$
- Transition relation formula $T(V, V')$ where $V' = \{v' \mid v \in V\}$

Representing Transition Systems

For SAT/SMT-solvers

- States are given as assignments to a set of variables

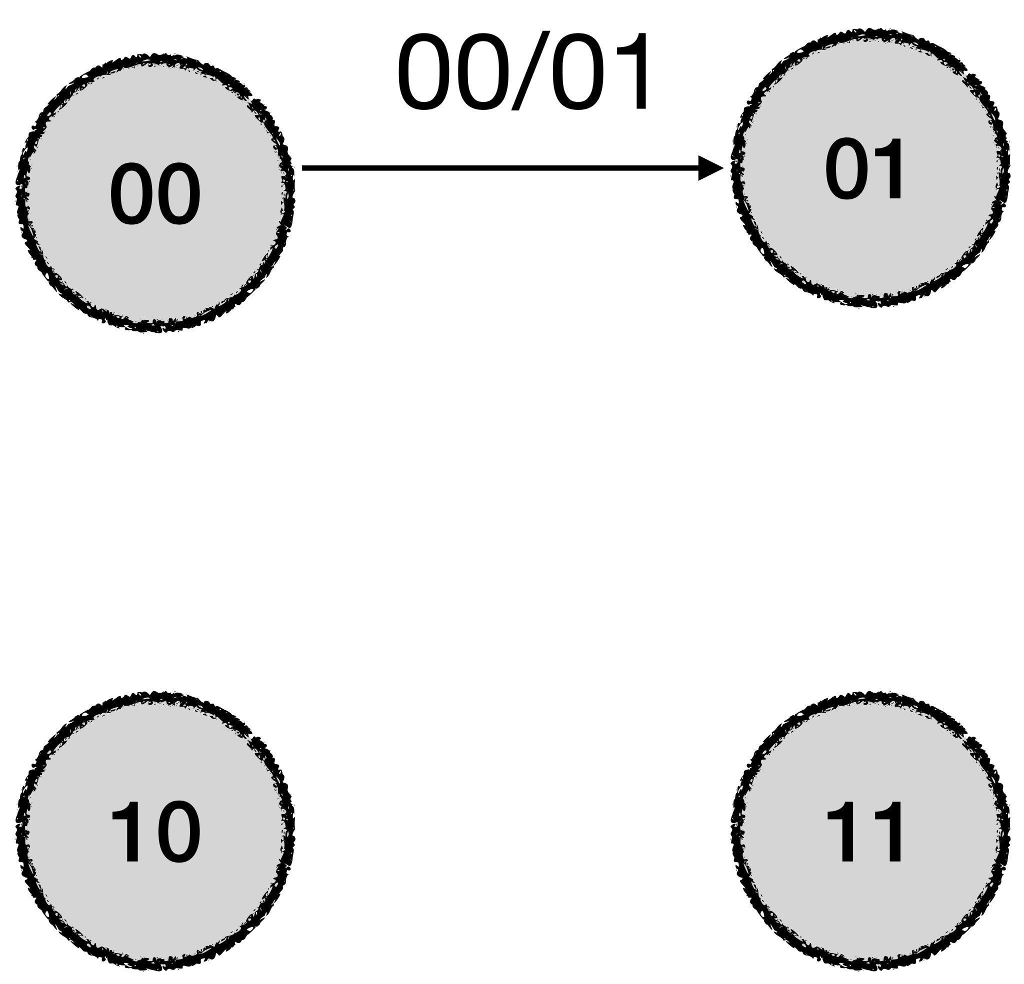


Representing Transition Systems

For SAT/SMT-solvers

- Transitions relate source and target states

x=0, y=0, x′=0, y′=1



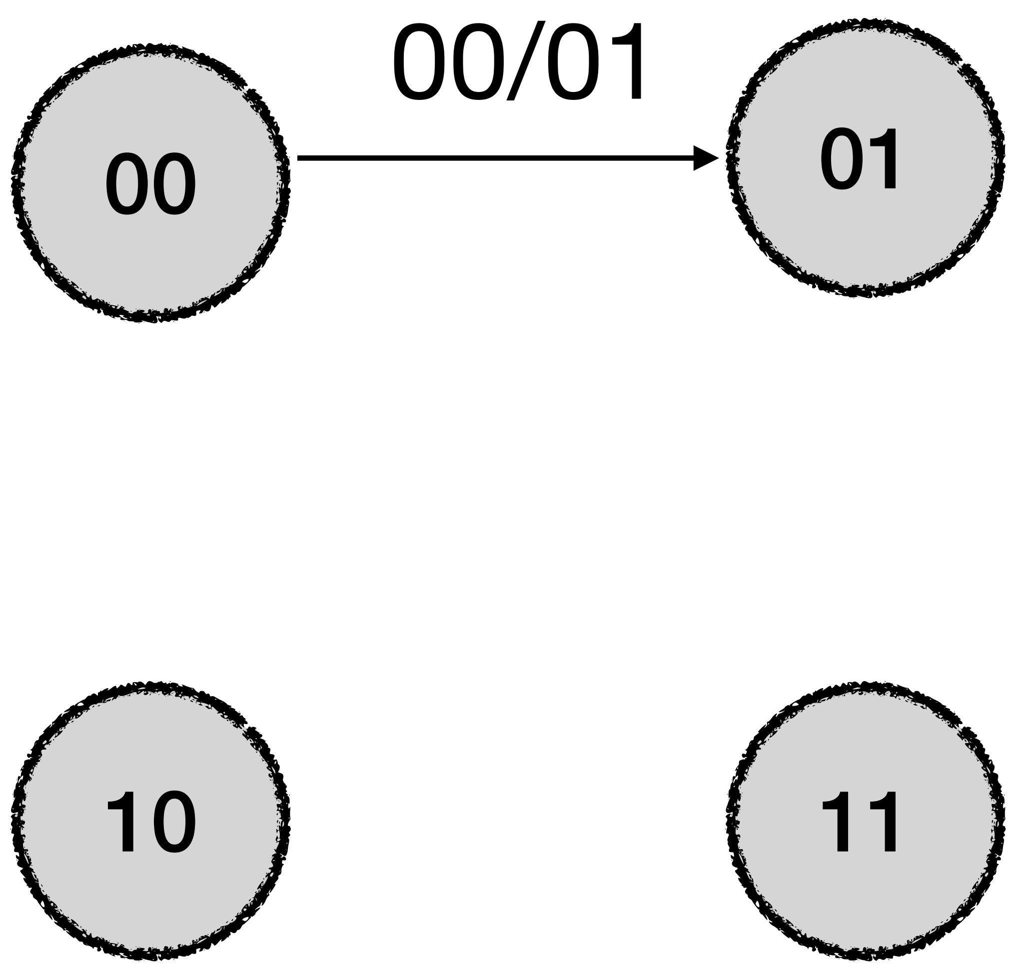
We use 'primed' variables
to describe the target states
of transitions

Representing Transition Systems

For SAT/SMT-solvers

- Transitions are described by a formula over normal and primed variables

$\neg x \land \neg y \land \neg x' \land y'$

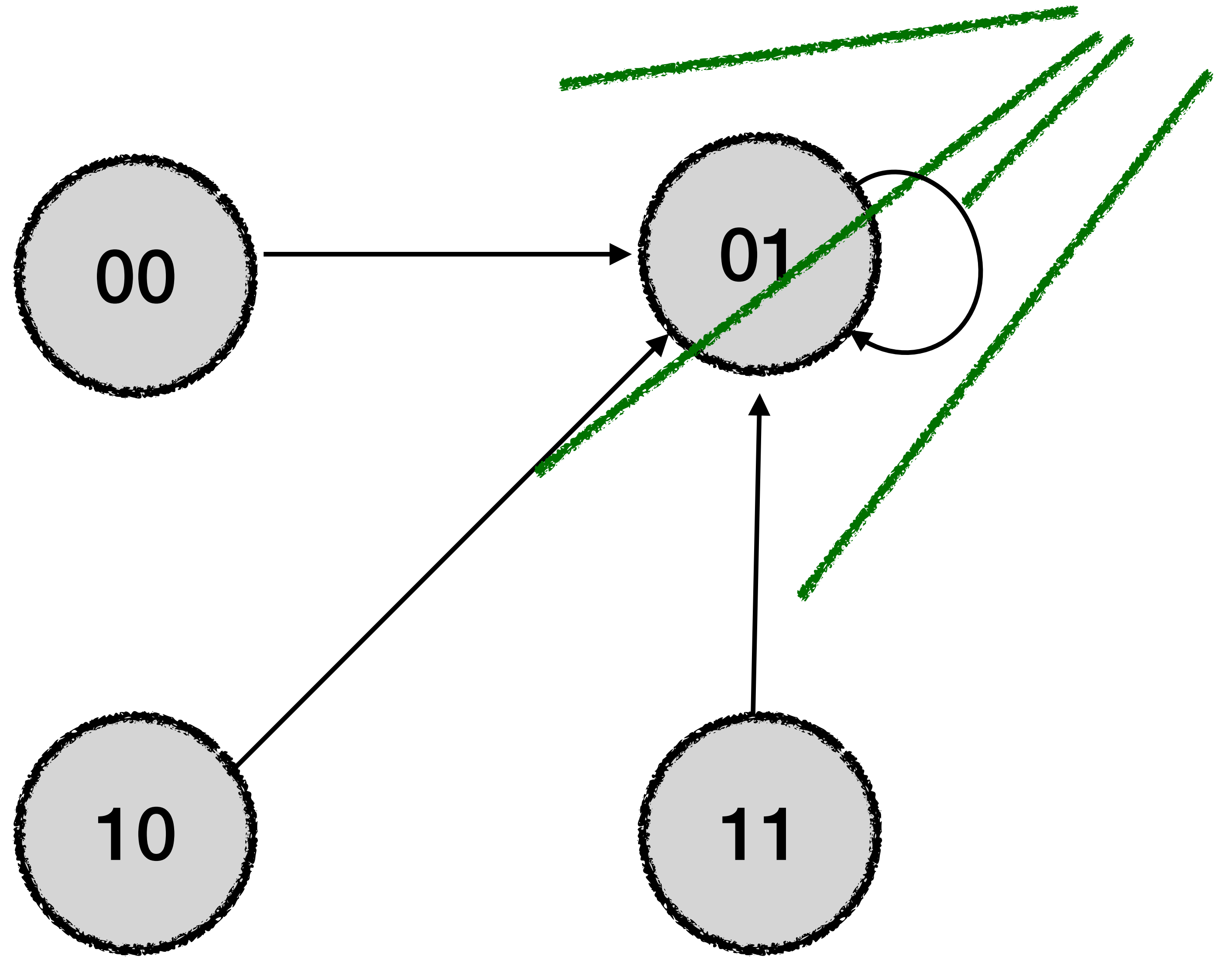


Representing Transition Systems

For SAT/SMT-solvers

- Transitions are described by a formula over normal and primed variables

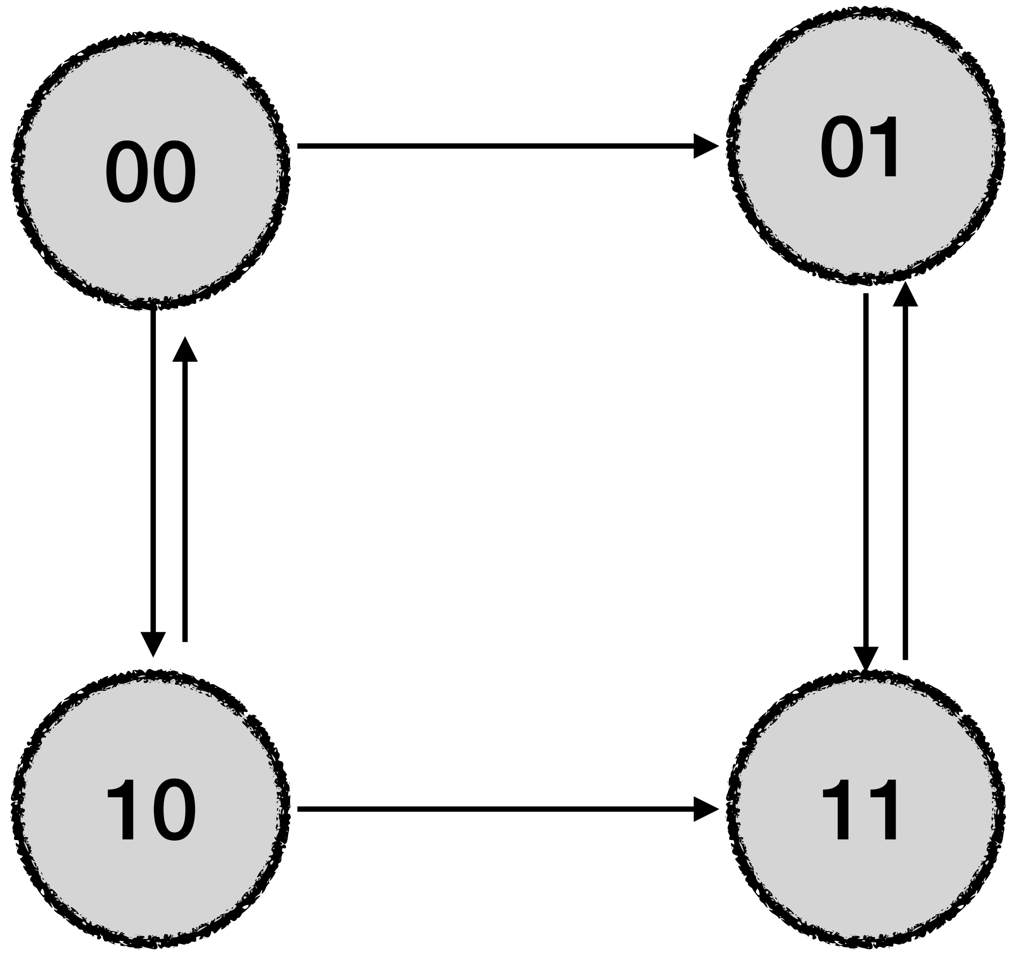
$\neg x' \land y'$



Quiztime

SAT solver guessing paths up to 2 steps

- Symbolic transition system?
- $V = \{x, y\}, I = \neg x \wedge \neg y, T = ?$
- $T = \Big((y \leftrightarrow y') \wedge (x \oplus x') \Big) \vee \Big((x \leftrightarrow x') \wedge \neg y \wedge y' \Big)$



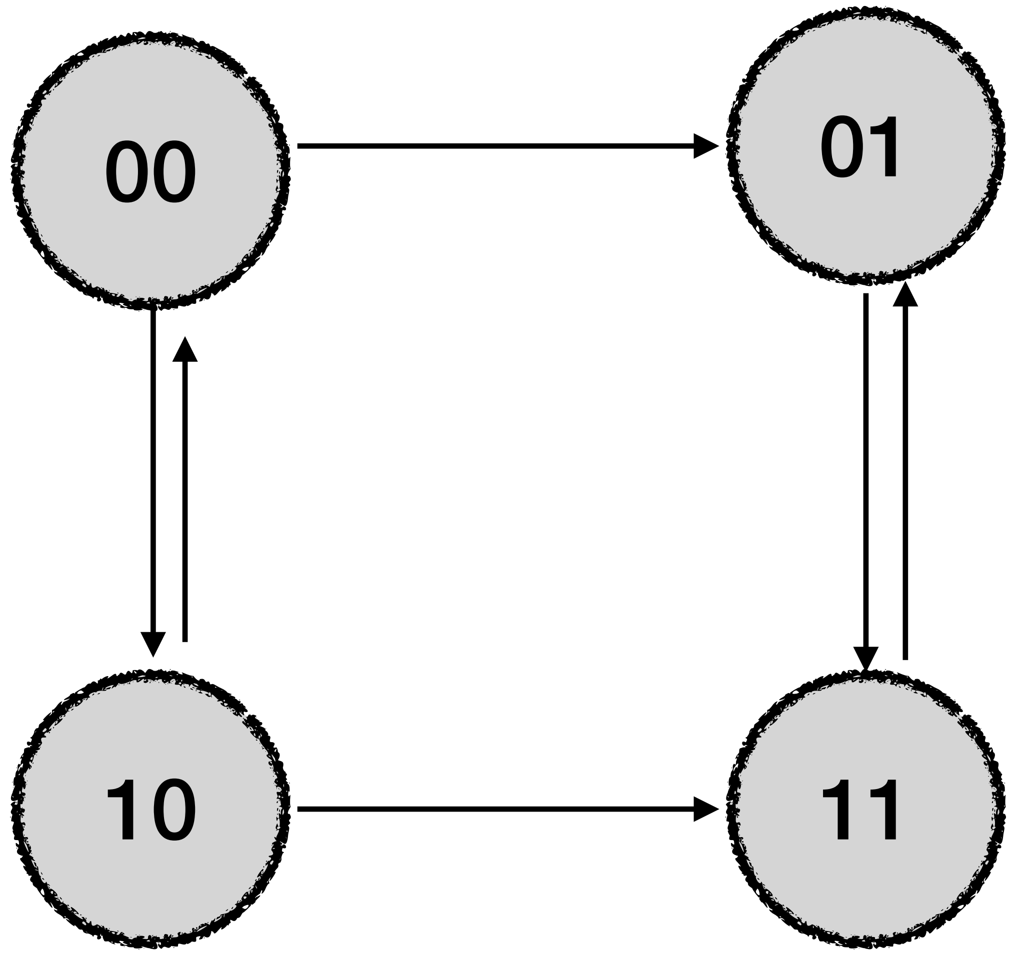
k-Bounded Reachability

- Is there a path of length up to k between two states?
- One variable per variable-set per time step. Variables: $X = \{v_i \mid v \in V, 0 \leq i \leq k\}$.
- Idea: Can we reach this state in k steps?
- Constraints:
  - If we can reach a state in i steps, then we can reach the successors in i + 1 steps.
  - We can reach the initial state in 0 steps.
  - We ensure that we reach the target in k steps.

Quiztime

SAT solver guessing paths up to 2 steps

- Symbolic transition system?
- $V = \{x, y\}, I = \neg x \wedge \neg y, T = ?$
- $T = \Big((y \leftrightarrow y') \wedge (x \oplus x')\Big) \vee \Big((x \leftrightarrow x') \wedge \neg y \wedge y'\Big)$
- Variables for paths of length 2

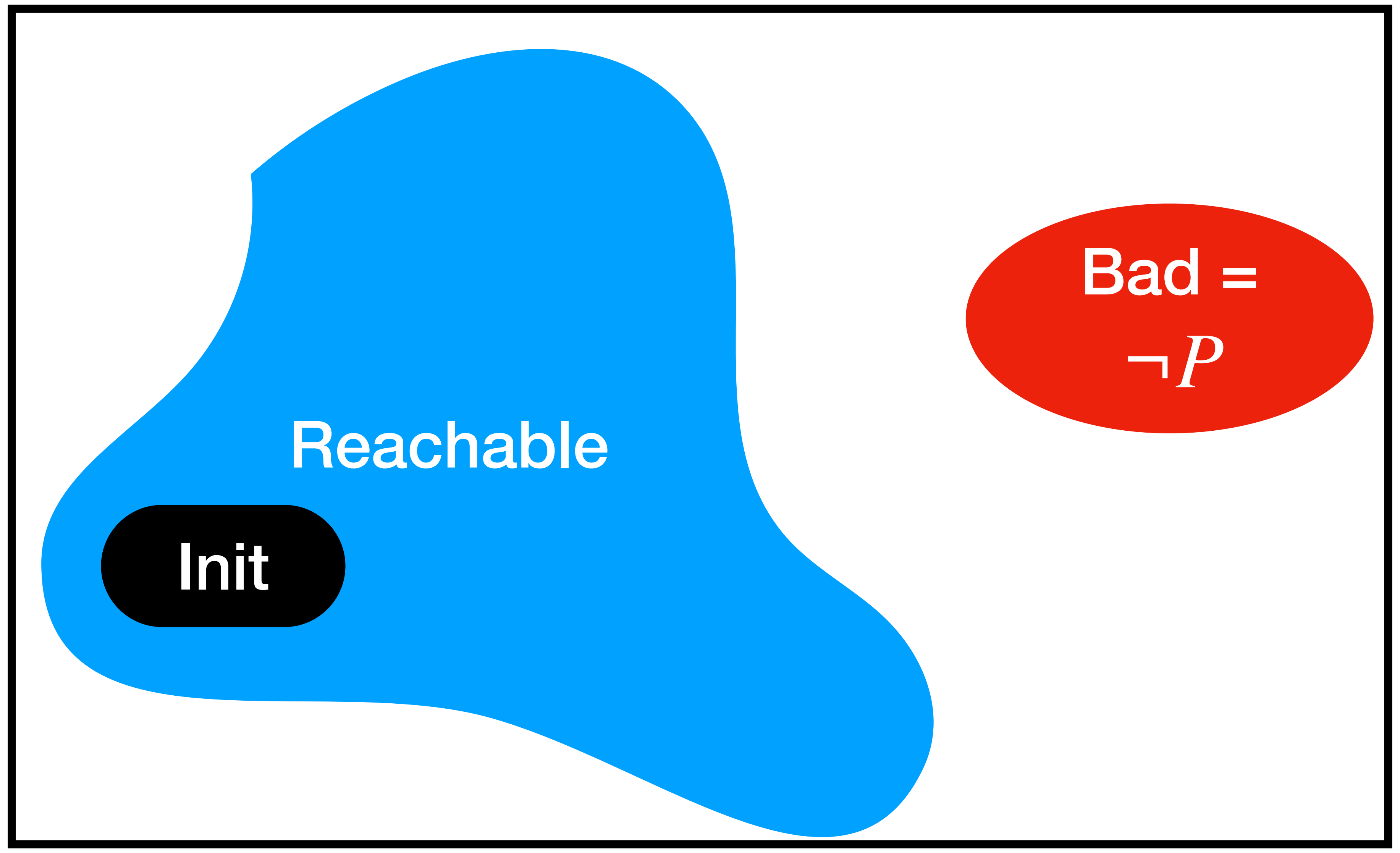


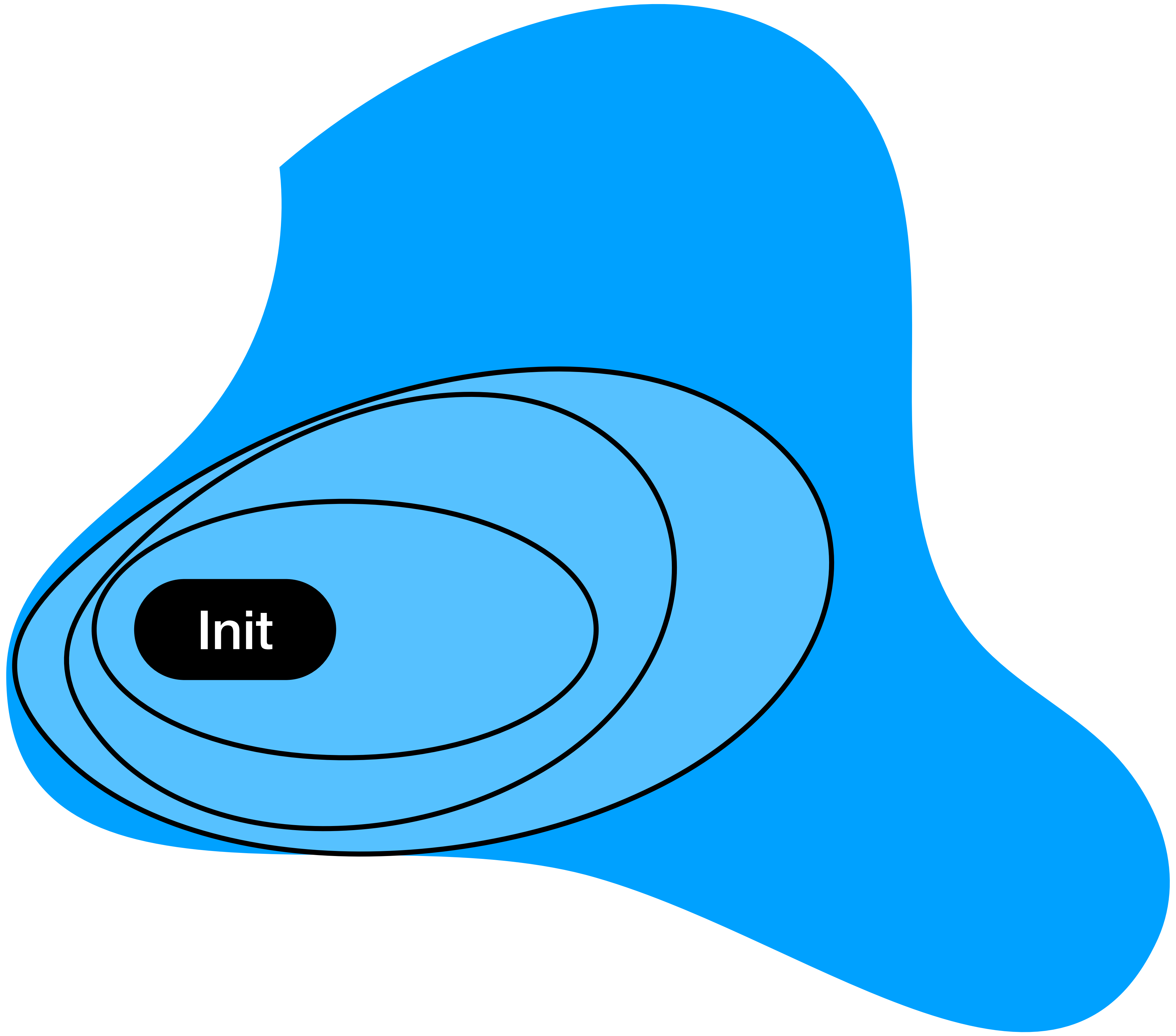
- Reachability constraints for exactly 2 steps? For at most 2 steps?

Summary

- (Un)Reachability as part of an encoding
- Symbolic transition systems

Approximating All Reachable States





Challenge:

- May require many steps to stabilize reachable states
- i-step reachable states may be not concisely representable

Abstraction

Rough idea

- Problem: Long paths are hard to guess, paths can be infinite, …
- Observation: Precise value of ‘data’ is often not important
- Rather than a system M, consider a system $\alpha(M)$ such that
  - $\alpha(M) \vDash \varphi \implies M \vDash \varphi$
  - $\alpha(M)$ is more concise/shorter paths
- General theoretic framework often based on Abstract Interpretation (Cousot & Cousot, 1977)

In this lecture: φ = “never reach a bad state”

Reverse direction does not always need to hold… We will discuss this later

Abstraction of Transition Systems

“Existential Abstraction”

- Let $M = (S, I, R)$ with states S, initial states I, transition relation R
- $\alpha : S \to \hat{S}$ abstracts states
- Obtain $\alpha(M) = (\hat{S}, \{\alpha(s) \mid s \in I\}, \hat{T})$, with
  - $\hat{T}(\hat{s}, \hat{s}')$ iff $\exists s, s'$ s.t. $T(s, s')$ and $\alpha(s) = \hat{s} \wedge \alpha(s') = \hat{s}'$

Every path in M is reflected by a path in $\alpha(M)$

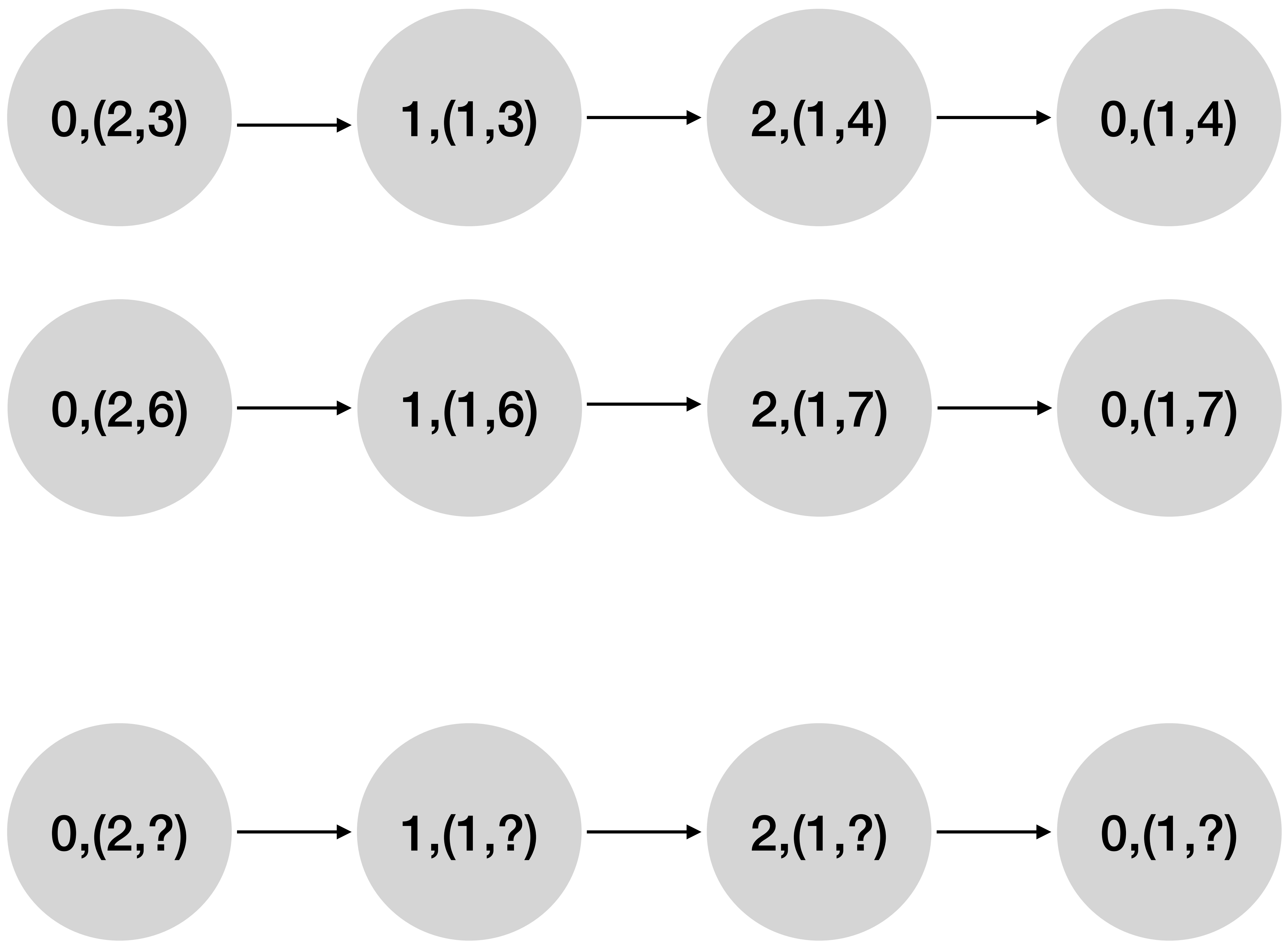
Example

Dead variable elimination

- initial = x > 0, y > 0
- 0: while (x >= 0)
  - 1: x = x - 1
  - 2: y = y + 1

Can remove this line

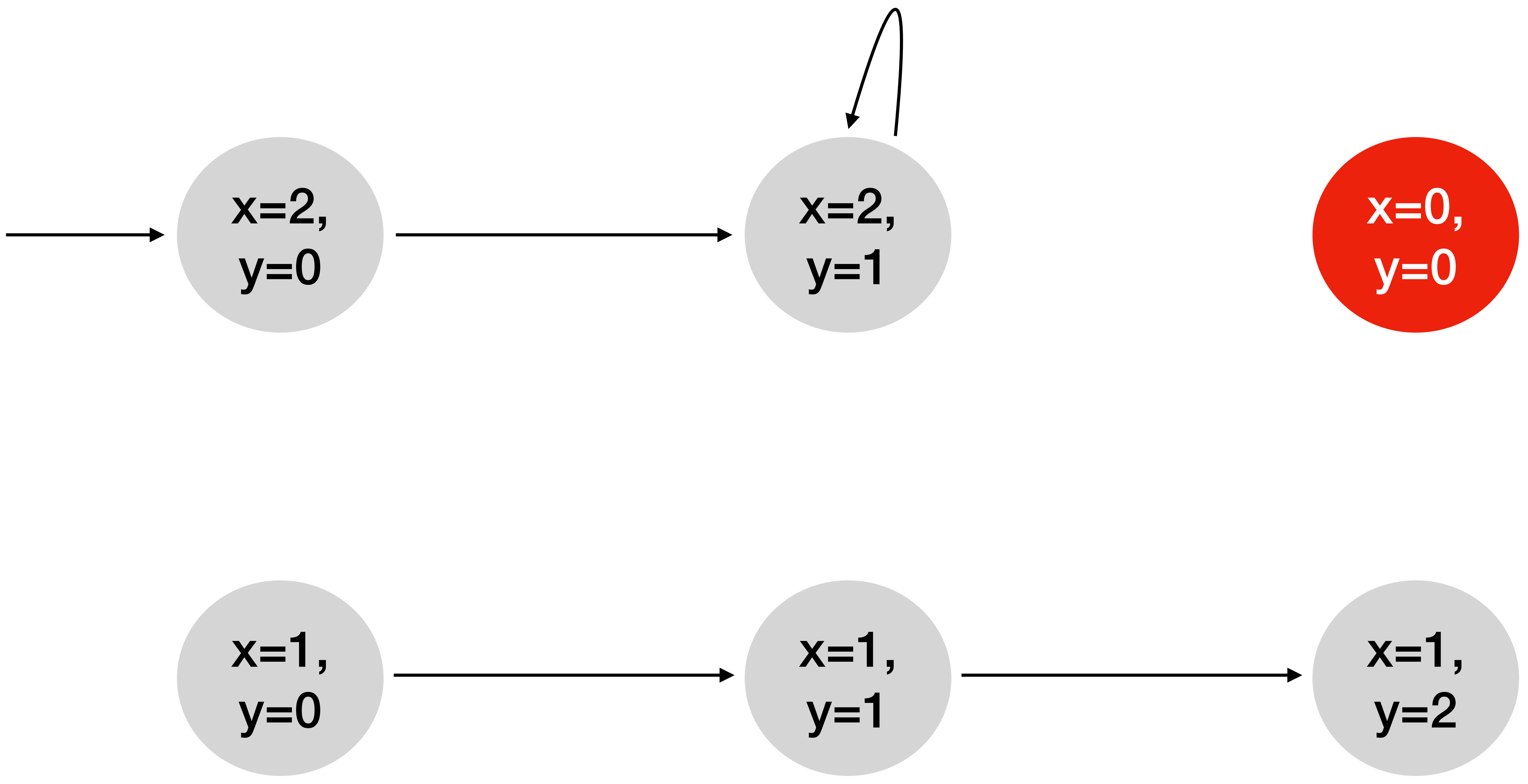
- return: x >= 0



Predicate Abstraction

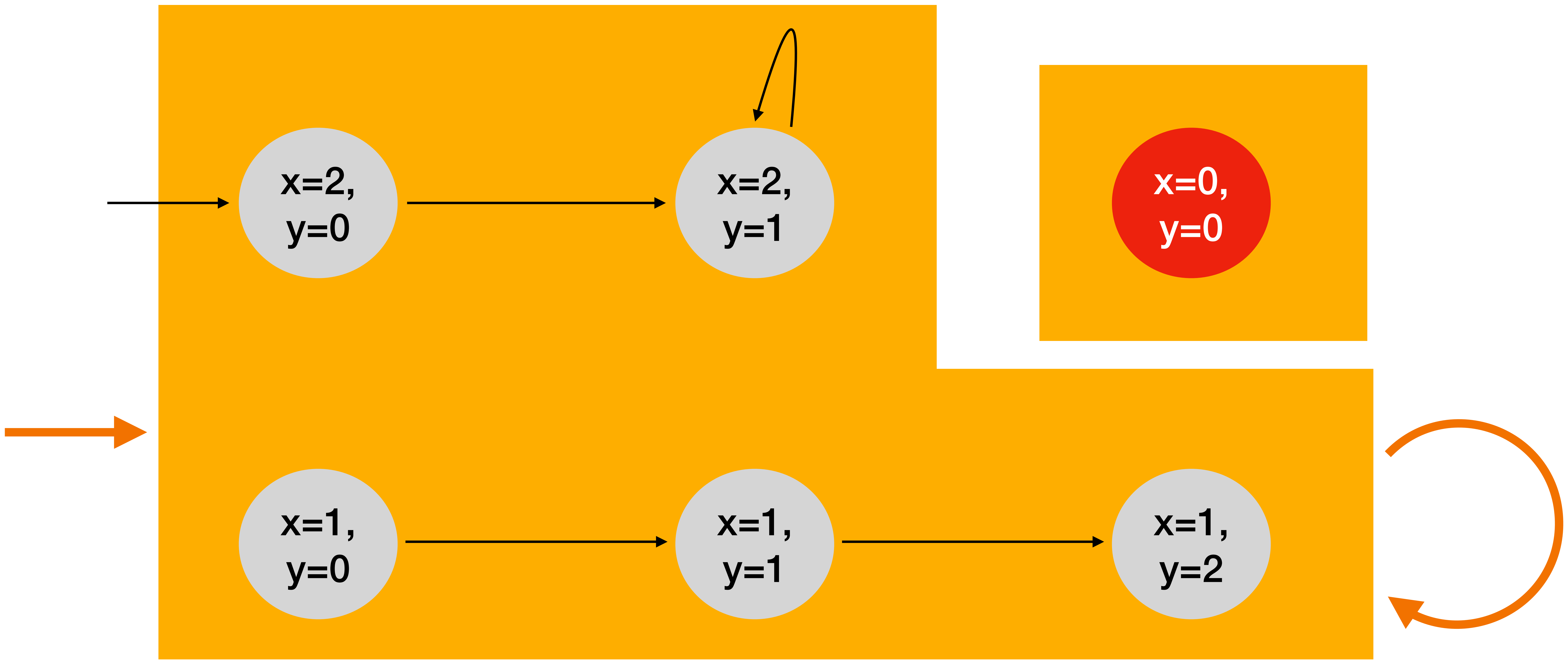
- States S given by an assignment to a set of N-valued variables $V = \{v_1, \ldots, v_n\}$
- Each predicate over variables V partitions the state space, $\beta : S \rightarrow \{0,1\}$ e.g., $x_1 > 2 \wedge x_4 \leq 3$
- For a set of predicates $\{\beta_1, \ldots, \beta_m\}$, $\hat{S} = \mathbb{B}^n$, $\alpha : S \rightarrow \hat{S}$, $\alpha(s) = [\beta_1(s), \ldots \beta_m]$
- Reduces the state space form N^n to 2^m

Example



Bad states: x = 0

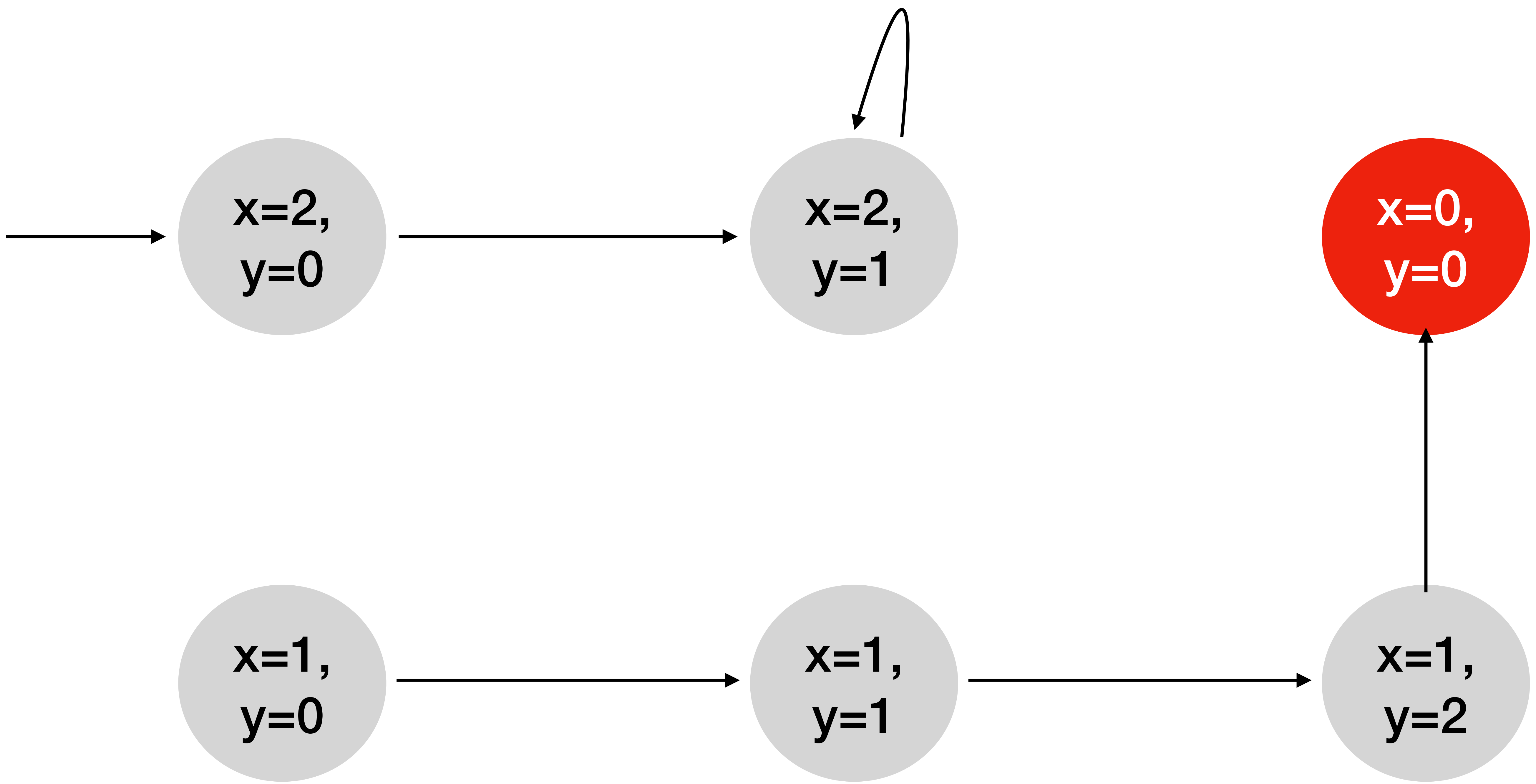
Example



Bad states: x = 0

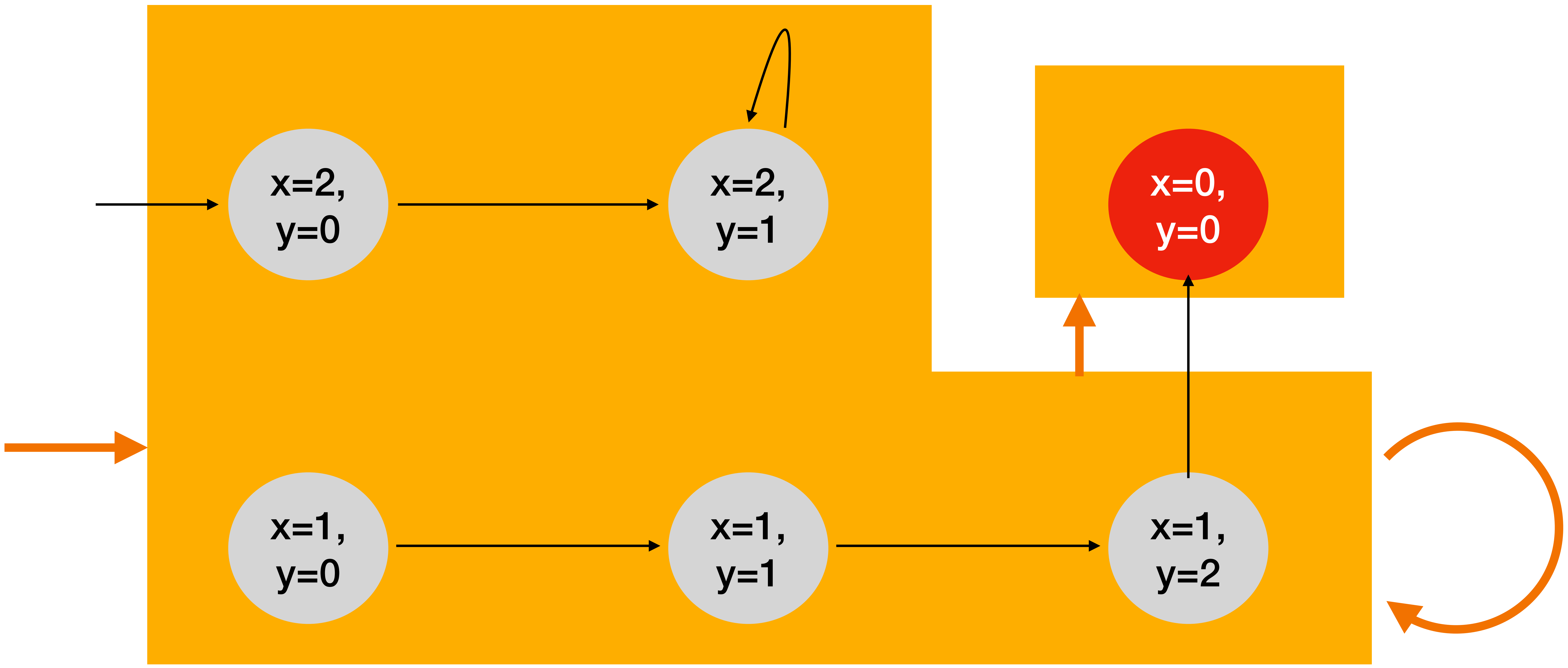
Predicates: { x = 0 }

Example



Bad states: x = 0

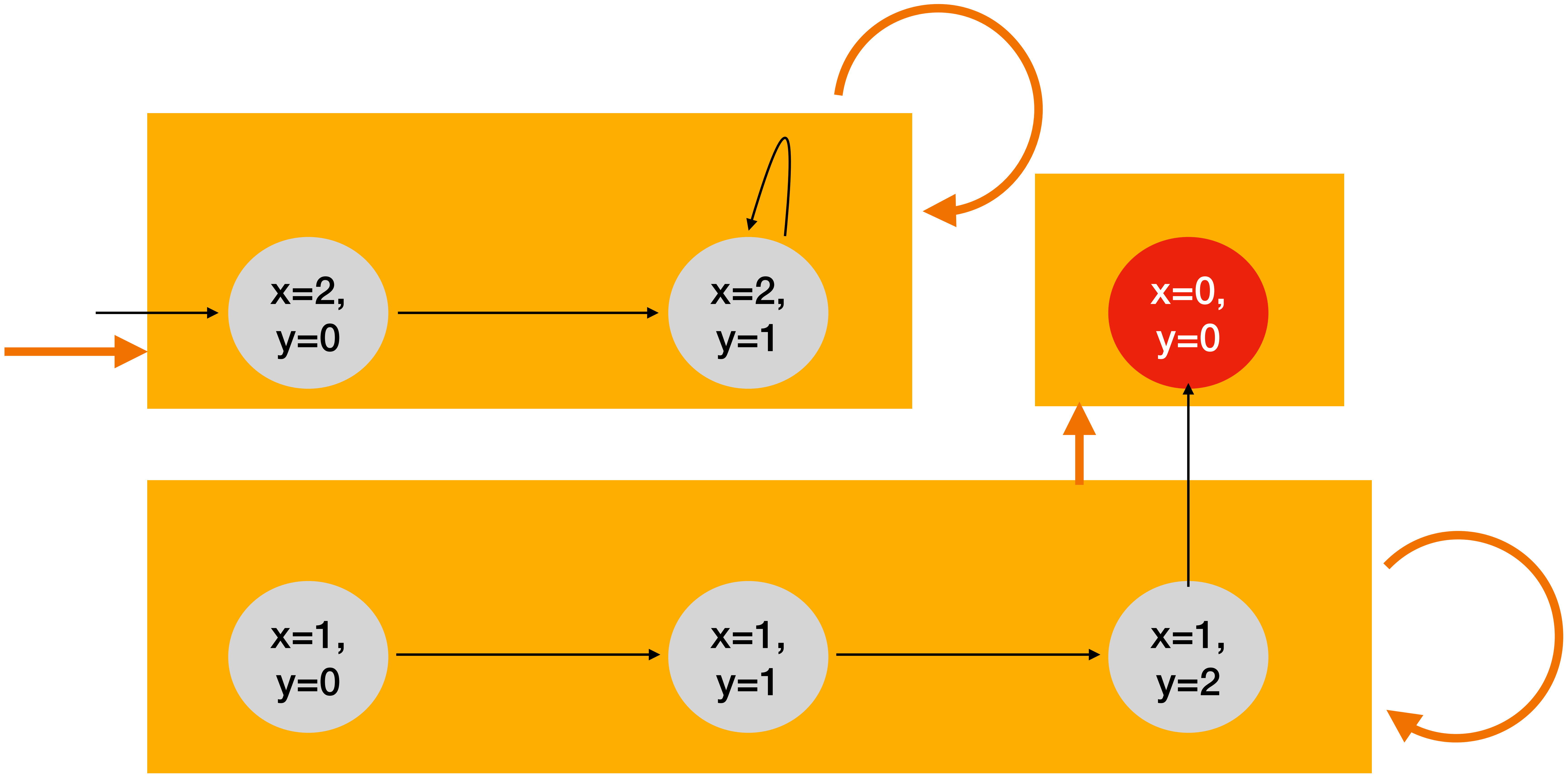
Example



Bad states: x = 0

Predicates: { x = 0 }

Example



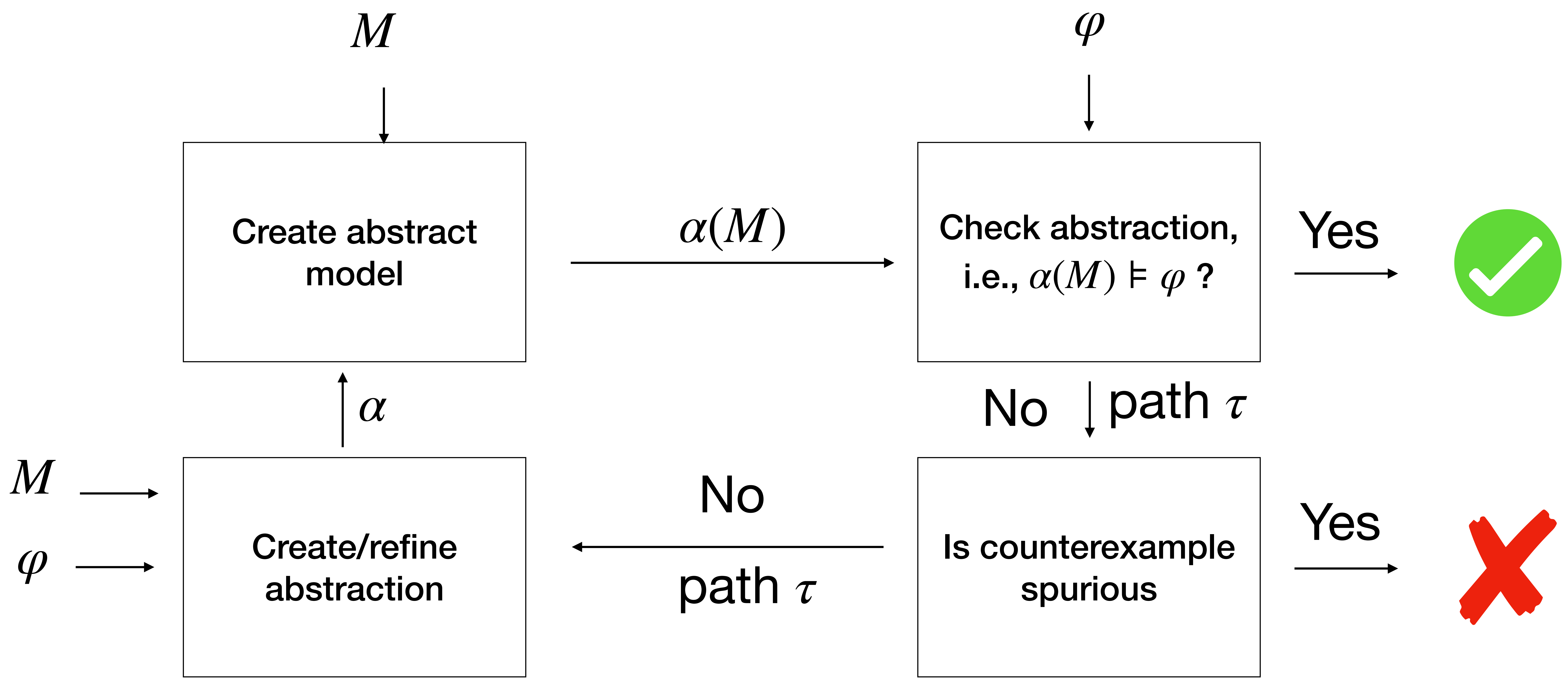
Bad states: x = 0

Predicates: { x = 0, x = 2 }

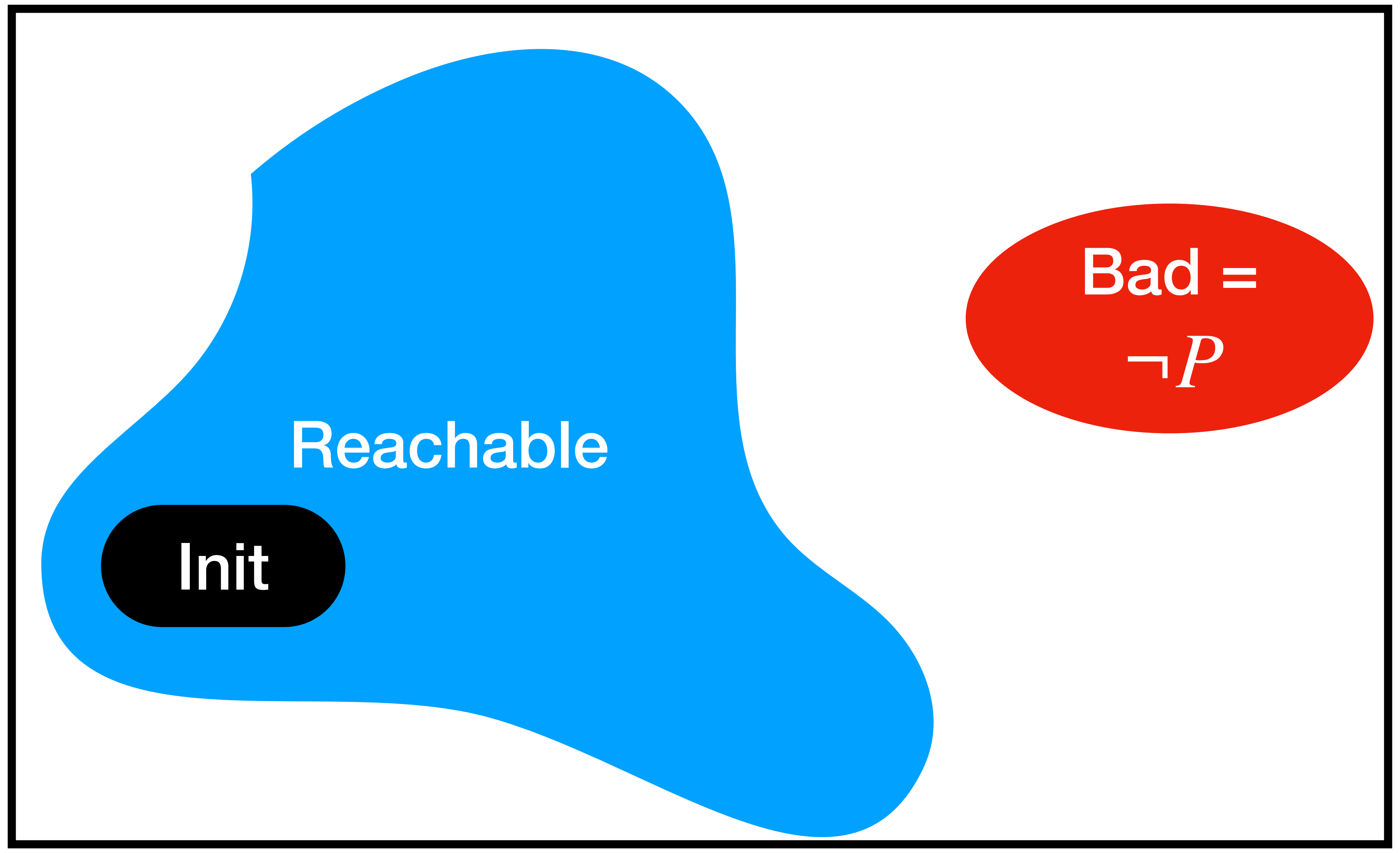
Abstraction-Refinement

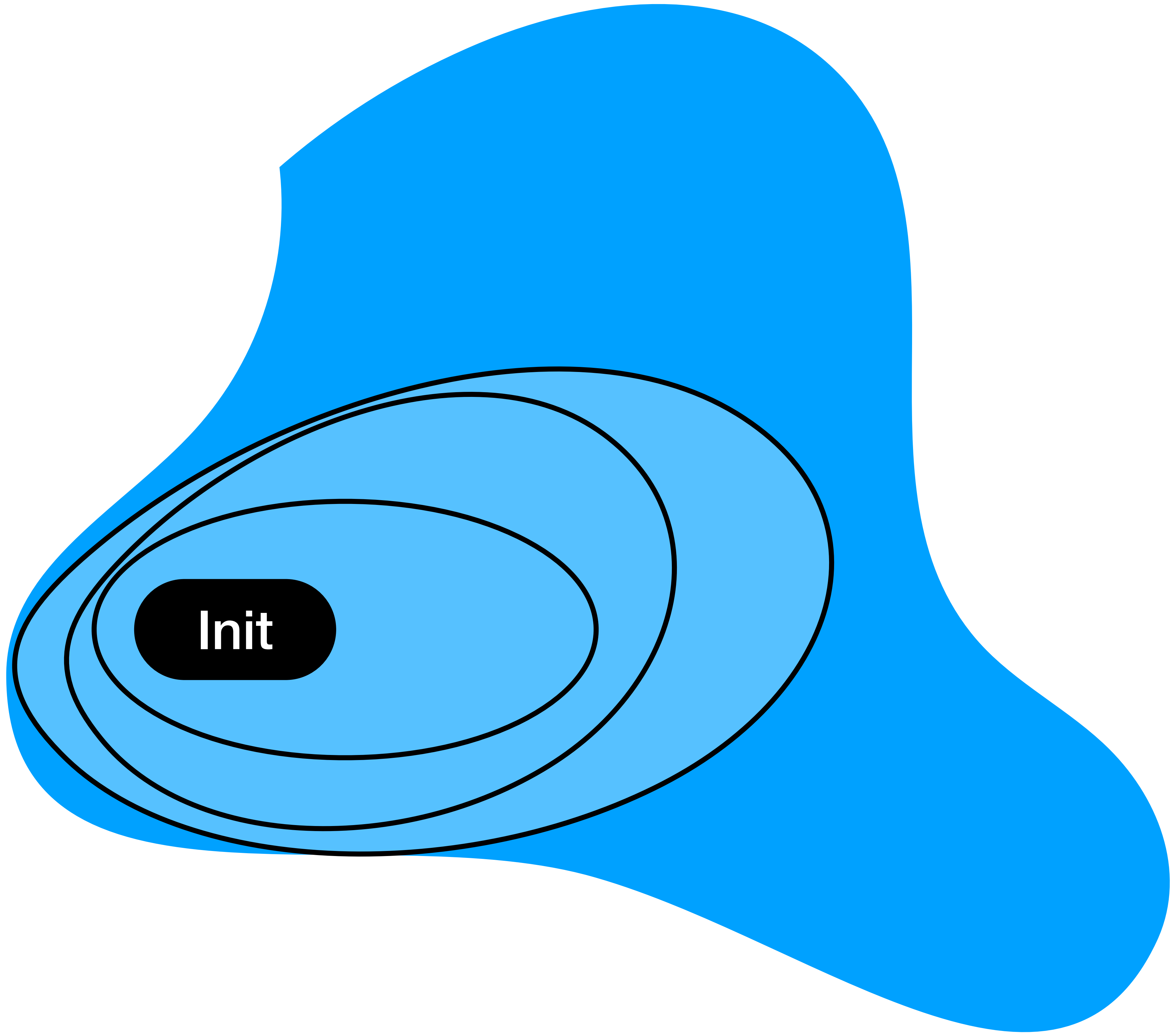
- Iteraterively adding predicates guaranteed to terminate
  - often, few predicates suffice.
- How to know which predicates to select…?
- To select optimal predicates is intractable, but..
  - looking at property and structure helps
  - and: looking at spurious counterexamples helps **a lot**

Counterexample-Guided Abstraction Refinement



Approximating All Reachable States





Challenge:

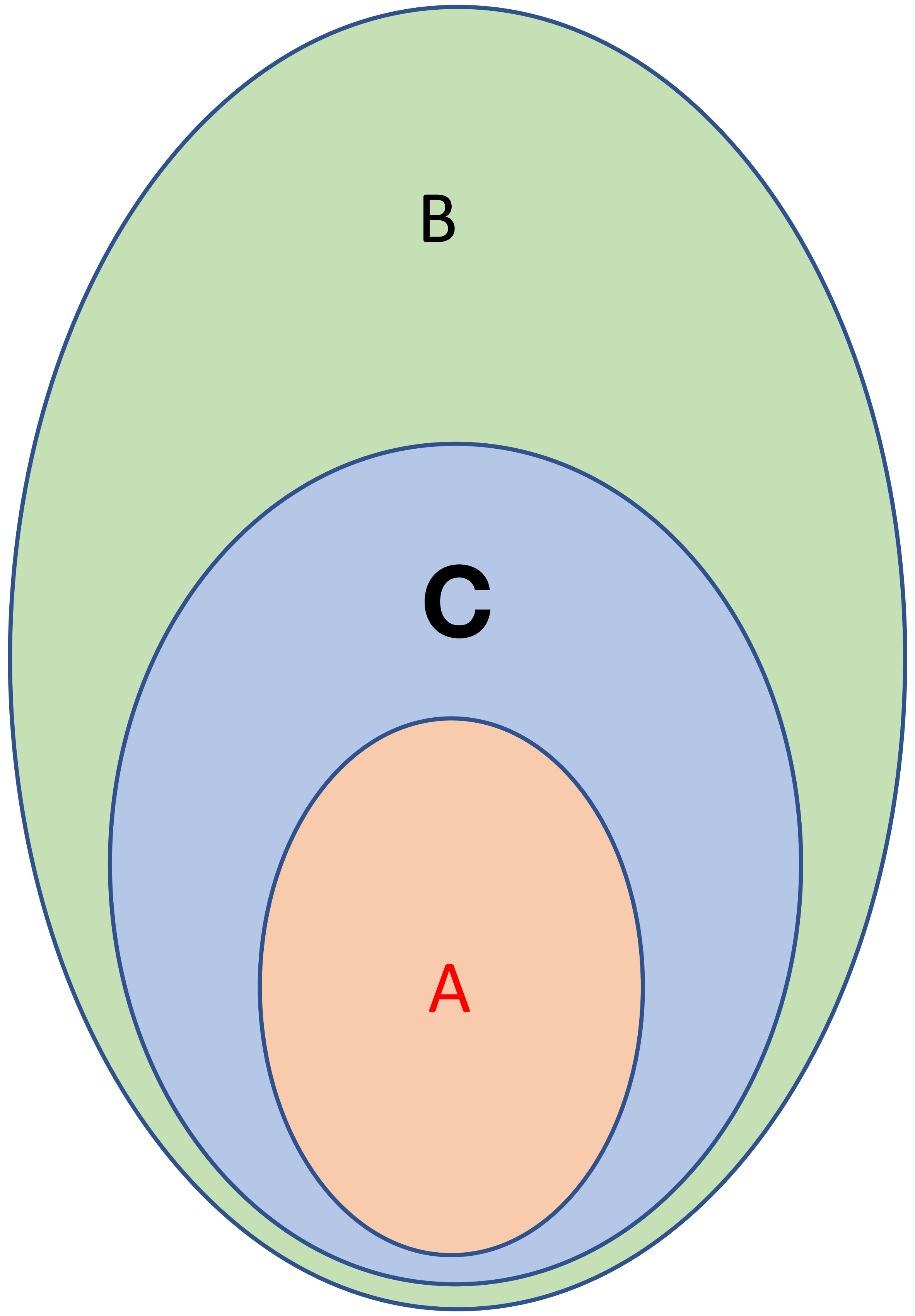
- May require many steps to stabilize reachable states
- i-step reachable states may be not concisely representable

Craig-Interpolants

Definition

- Given two (first-order) formulas A and B
  - with variables var(A) and var(B)
- C is a **Craig-Interpolant**, iff
  - A implies C
  - C implies B
  - var(C) = var(A) ∩ var(B)

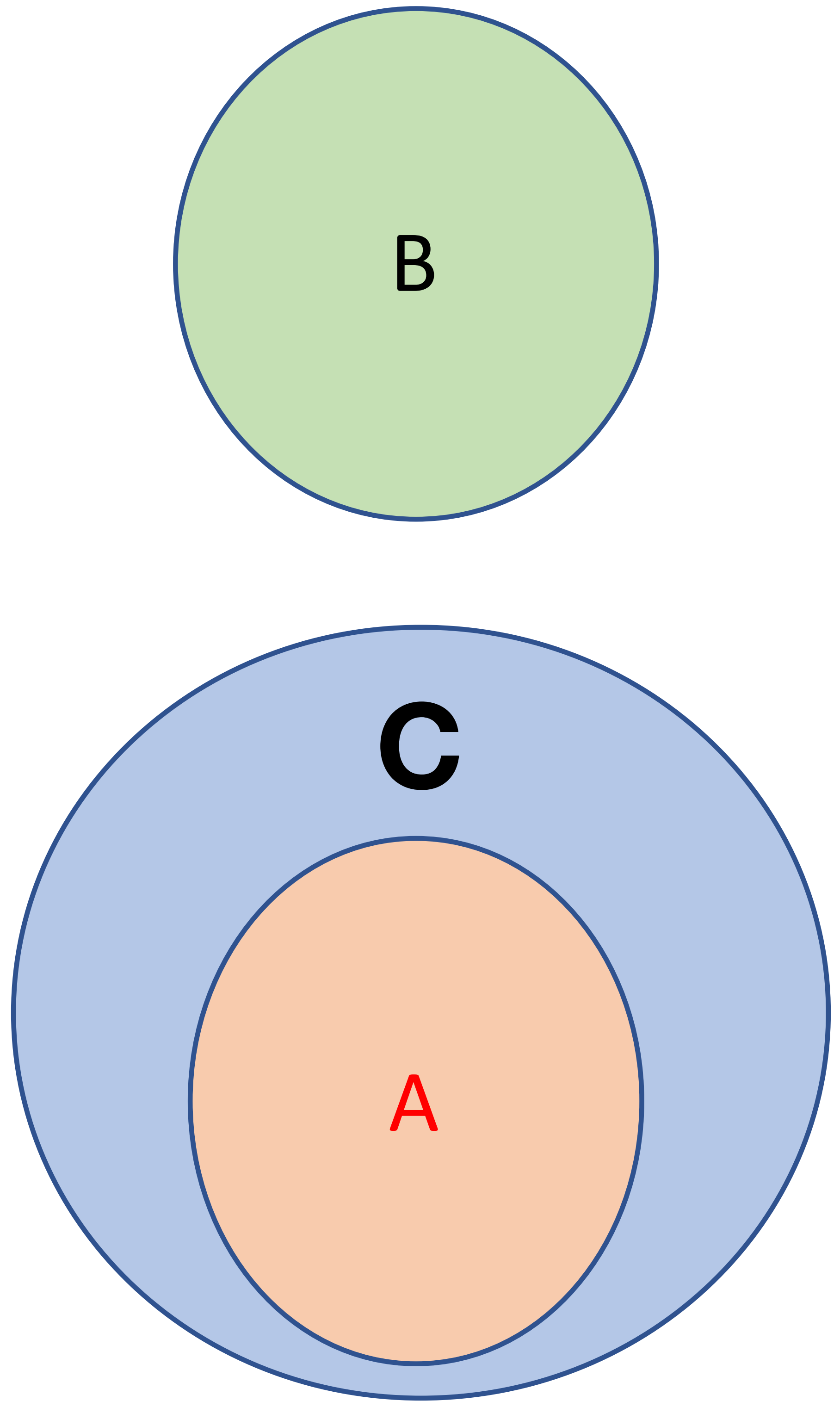
In general, replace variables with non-logical symbols



(Reverse) Interpolants

McMillans Formulation (used hereafter)

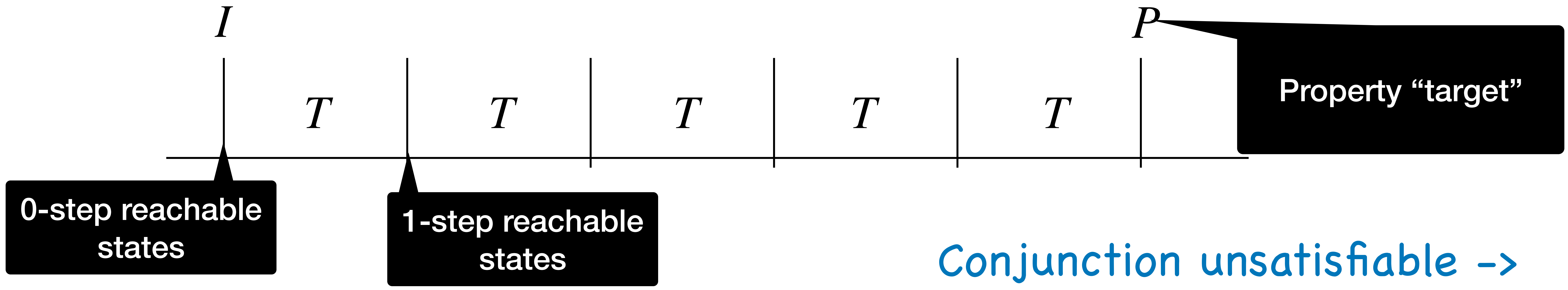
- Given two (first-order) formulas A and B
  - with variables var(A) and var(B)
  - With $A \wedge B$ unsatisfiable
- C is an **interpolant**, iff
  - A implies C
  - C implies **not** B = not(B and C)
  - var(C) = var(A) ∩ var(B)



Computing Interpolants

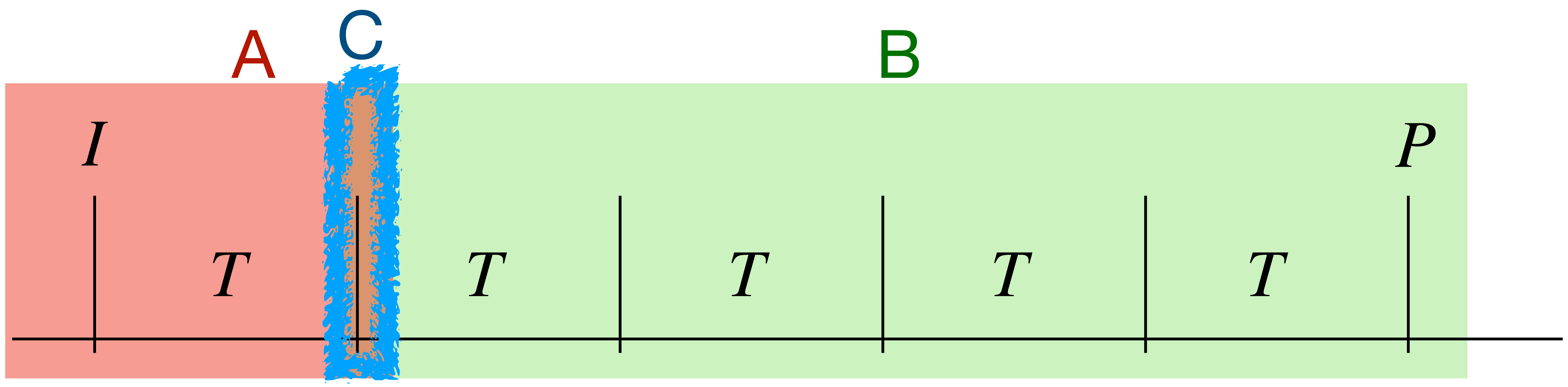
- For many SMT theories (including UF, EQ, LRA) interpolants can efficiently be computed from a resolution proof
- Details are beyond the scope of this lecture

Interpolants on with k-bounded reachability



Conjunction unsatisfiable -> no path of length k

Partition conjunction into A and B



Interpolant C overapproximates reachable states without admitting an k-1 path to P

Interpolation-Based Reachability

Extension to bounded reachability

For a fixed k:

1. Set REACH initially to I
2. Ask for k-bounded reachability from REACH
   - If SAT: have we found a counterexample?
   - If UNSAT, continue
3. Update REACH:
   - Use interpolation to compute over-approximation of one-step reachable states;
   - add them to REACH
     - Can newly added states lead to error states in k-1 steps?
     - In k steps?
4. If REACH does not increase, we've reached a fixed point!
   - Is the property true?
5. Otherwise, back to step 2

Increment k to resolve spurious counterexamples

Only if REACH = initial states

Summary

- Bounded Reachability, Reachability, and Unreachability with a SAT-solver
- Symbolic transition systems
- Interpolants
- Abstraction

Questions

Make sure you can answer the following questions!

A. How can a SAT-solver prove unreachability? What is an inductive state set?

B. Why is inductivity not enough to prove reachability? What idea can we use to fix this?

C. How do we represent symbolic transition systems? How can a SAT-solver find paths up to length k?

D. ~~What is an interpolant? (NOT DISCUSSED IN LECTURE)~~

E. What is existential abstraction and how does abstraction-refinement work?

See you next week!