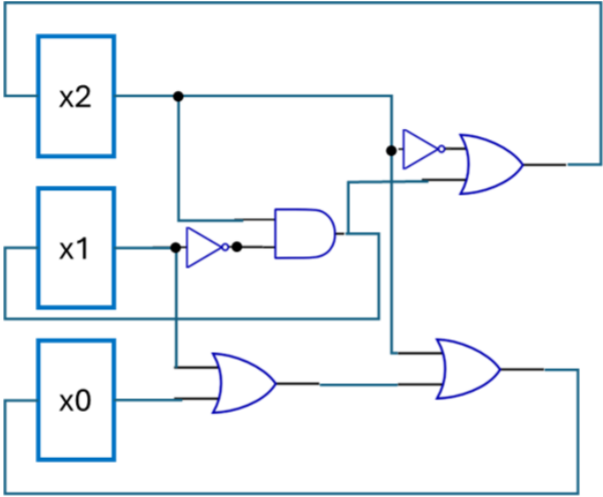


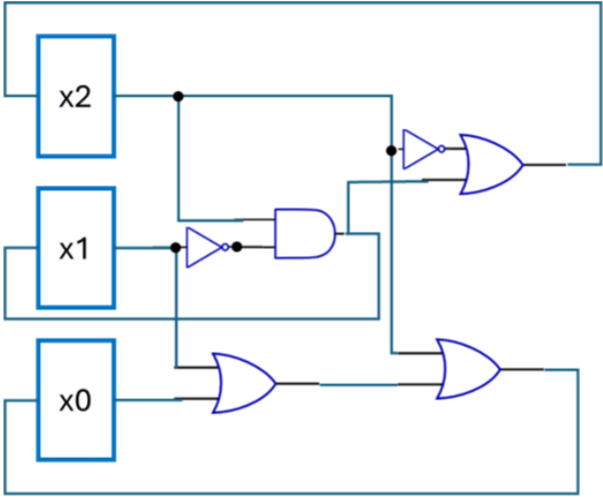
Consider the following synchronous circuit  $C$ . The initial value of the state variable  $x_0$  is true. The initial values of  $x_1$  and  $x_2$  are unknown.



**Task 1. [50 points]** State the formula  $S_0$  that represents the set of initial states and the formula  $R$  that represents the transition relation of  $C$ .

**Task 2. [50 points]** Draw the Kripke structure  $M = (S, S_0, R, AP, L)$  that represents  $C$ . (Add an incoming arrow for each initial state.)

Consider the following synchronous circuit  $C$ . The initial value of the state variable  $x_0$  is true. The initial values of  $x_1$  and  $x_2$  are unknown.



**Task 1. [50 points]** State the formula  $S_0$  that represents the set of initial states and the formula  $R$  that represents the transition relation of  $C$ .

**Task 2. [50 points]** Draw the Kripke structure  $M = (S, S_0, R, AP, L)$  that represents  $C$ . (Add an incoming arrow for each initial state.)

Task 1:

$$S_0(x_0, x_1, x_2) = x_0$$

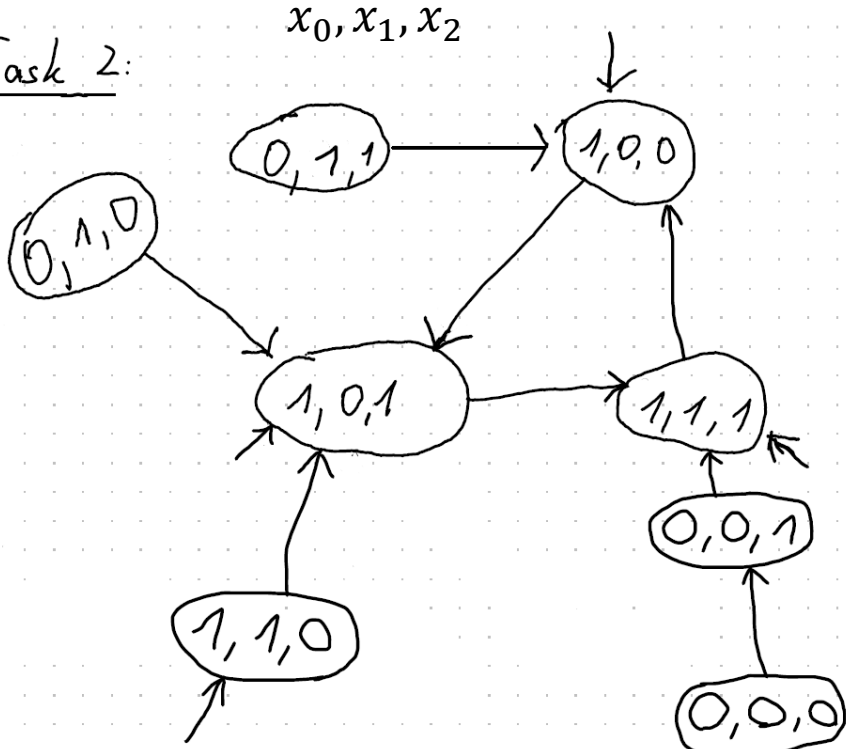
$$R(x_0, x_1, x_2, x_0', x_1', x_2') =$$

$$= (x_0' \leftrightarrow (x_0 \vee x_1 \vee x_2)) \wedge$$

$$(x_1' \leftrightarrow (\neg x_1 \wedge x_2)) \wedge$$

$$(x_2' \leftrightarrow (\neg x_2 \vee \neg x_1 \wedge x_0))$$

Task 2:



# SAT-Based Model Checking

Chapter 10

# Outline

- Bounded Model Checking
- Verifying Reachability Properties with  $k$ -induction
- Model Checking with Inductive Invariants
- Model Checking with Craig Interpolants
- Property-Directed Reachability

# Overview

- SAT solvers can solve propositional formulas. (See Chapter 9.)
- SAT solvers have become very fast

## Techniques

- Bounded Model Checking: Is there a trace of length  $k$  that violates the property?
- $k$ -induction: Can we prove the property inductively for *any* trace?
- Create an inductive invariant that is stronger than the property
  - Craig Interpolants
  - Property-Directed Reachability

**In this chapter, we will only consider LTL formulas**

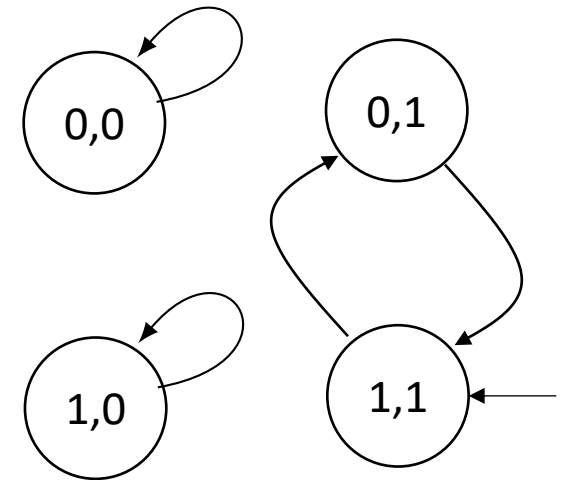
# Preliminaries

# Postimage

$\text{postimg}(S)$  = states reachable from  $S$  in one step

$\text{postimg}(\{ (0,1) \}) =$

$\text{postimg}(\{ (0,0), (1,1) \}) =$



Kripke structure

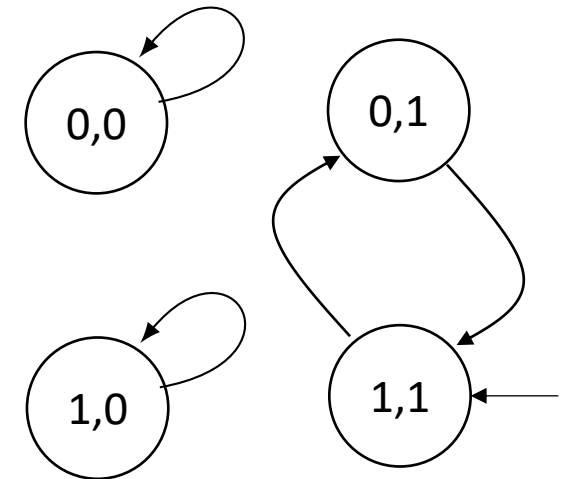
# Paths

Initial states:  $\mathcal{S}_0(x, y) =$

Transitions:  $\mathcal{R}(x, y, x', y') =$

How do we express that there is a path of length 1 from the initial states?

$Path_1(V, V') =$ .



Kripke structure



# Paths

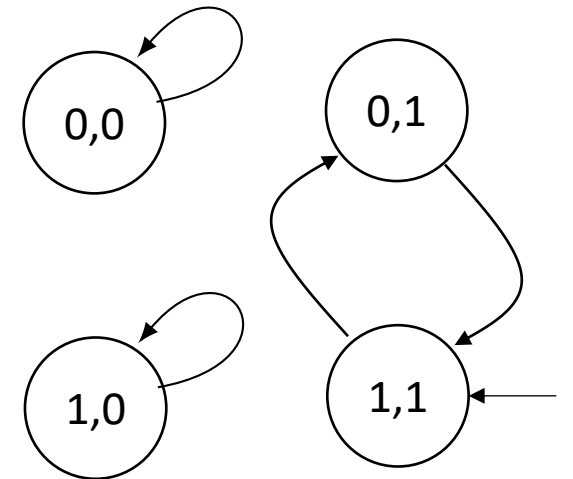
Initial states:  $\mathcal{S}_0(x, y) = (x = 1 \wedge y = 1)$

Transitions:  $\mathcal{R}(x, y, x', y') = (x' = (x + y) \bmod 2) \wedge (y' = y)$

$Path_1(V, V') = S \wedge R(V, V') =$

$x \wedge y \wedge (x' = (x + y) \bmod 2) \wedge (y' = y) =$

$x \wedge y \wedge \neg x' \wedge y'$ .



Kripke structure

# Bounded Model Checking

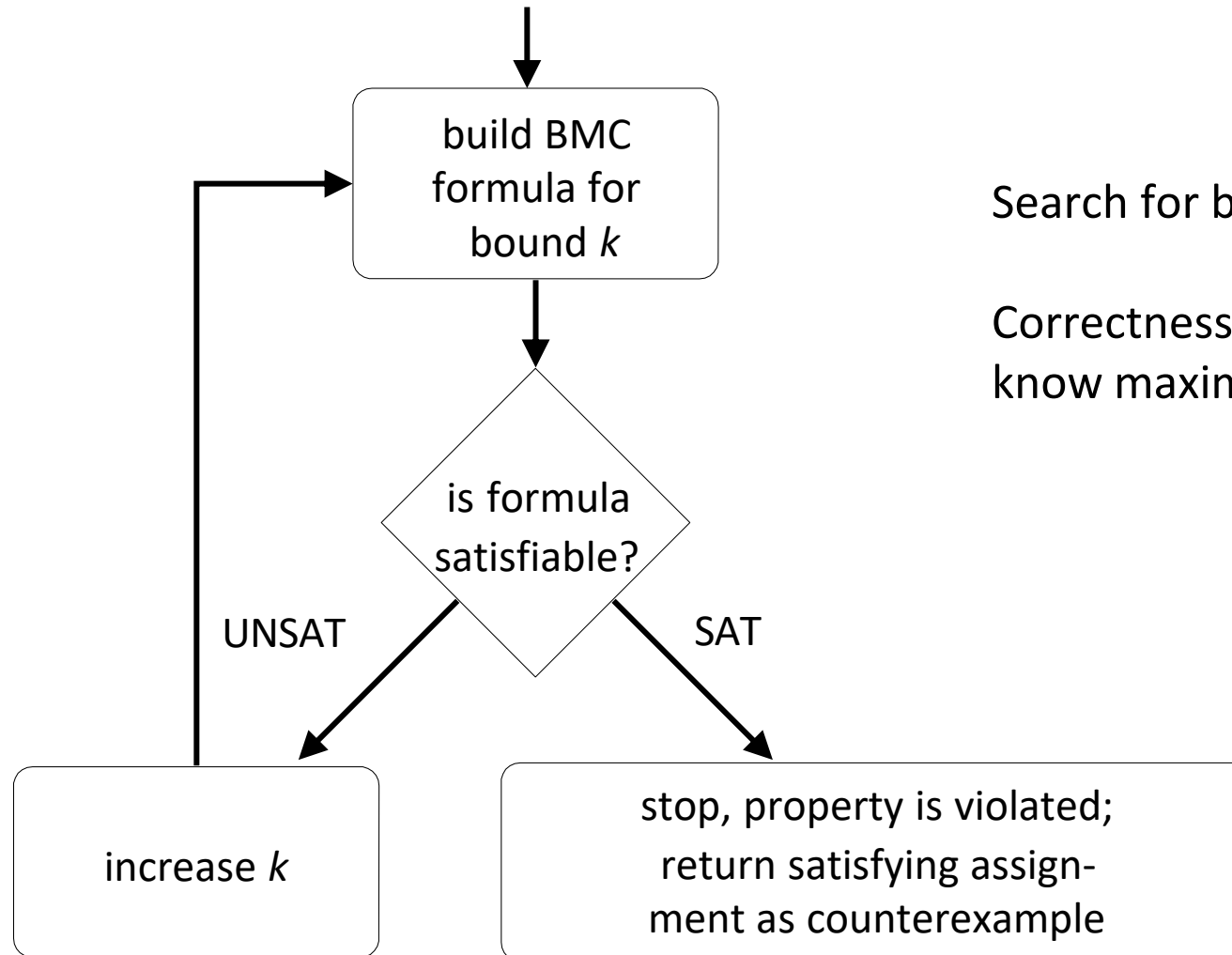
# Computer Aided Verification Award 2018

**Bounded Model Checking** has revolutionized the way model checking is used and perceived. It has increased the capabilities of model checkers by orders of magnitude, turning them into a standard tool for hardware verification and a very important component of the toolkit available for software verification...

(1999)



# Bounded Model Checking



Search for bugs within  $k$  steps.

Correctness can only be proven if we know maximal value for  $k$

# Reachability Properties

## Property

$p$  always holds iff we cannot reach a state with  $\neg p$

## Kripke Structure

$M = (S, S_0, R, AP, L)$  – represented symbolically (See Chapter 3)

State variables  $V = \{v_1, \dots, v_n\}$

## Reachability – Paths

$$\mathit{path}_0(s_0) =$$

$$\mathit{path}_1(s_0, s_1) =$$

$$\mathit{path}_2(s_0, \dots, s_2) =$$

$$\mathit{path}_k(s_0, \dots, s_k) =$$

## Reachability – Paths

$$path_0(s_0) = S_0(s_0)$$

$$path_1(s_0, s_1) = S_0(s_0) \wedge R(s_0, s_1)$$

$$path_2(s_0, \dots, s_2) = S_0(s_0) \wedge R(s_0, s_1) \wedge R(s_1, s_2)$$

$$path_3(s_0, \dots, s_3) = S_0(s_0) \wedge R(s_0, s_1) \wedge R(s_1, s_2) \wedge R(s_2, s_3)$$

$$path_k(s_0, \dots, s_k) = S_0(s_0) \wedge \bigwedge_{i=0}^{k-1} R(s_i, s_{i+1})$$

# Reachability – Building the Formula



# Reachability – Building the Formula

$$path_k(s_0, \dots, s_k) = S_0(s_0) \wedge \bigwedge_{i=0}^{k-1} R(s_i, s_{i+1})$$

Path starts in initial state      Exists transition from  $s_i$  to  $s_{i+1}$

System is **incorrect** within  $k$  steps if

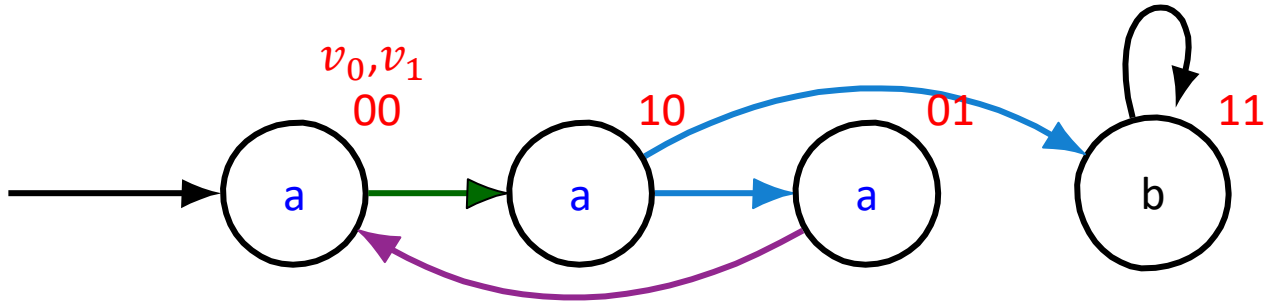
$$path_k(s_0, \dots, s_k) \wedge \bigvee_{i=0}^k \neg p(s_i)$$

There is a path to  $s_k$       One of the state violates  $p$

## Reachability – Correctness

**Theorem 10.1.**  $path_k(s_0, \dots, s_k) \wedge \bigvee_{i=0}^k \neg p(s_i)$  is satisfiable iff there is a counterexample to  $AG\ p$  of length  $\leq k$ .

# Example



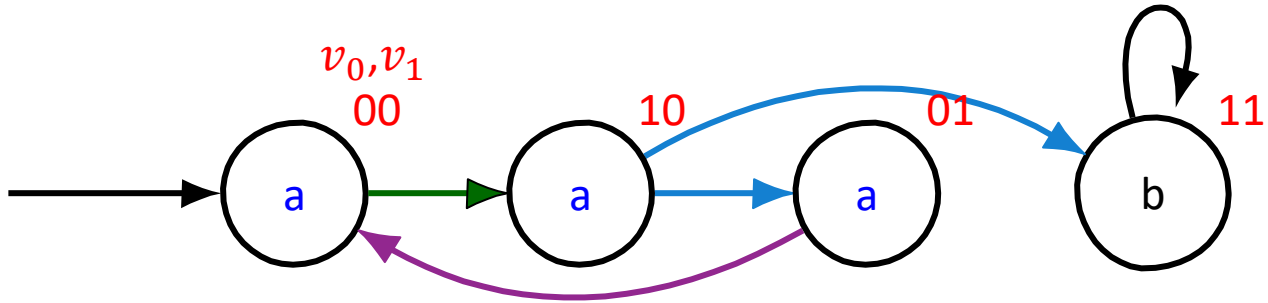
**AG a**

$S_0(v_0, v_1) = .$

$R(v_0, v_1, v'_0, v'_1) =$

$a(v_0, v_1) = .$

# Example



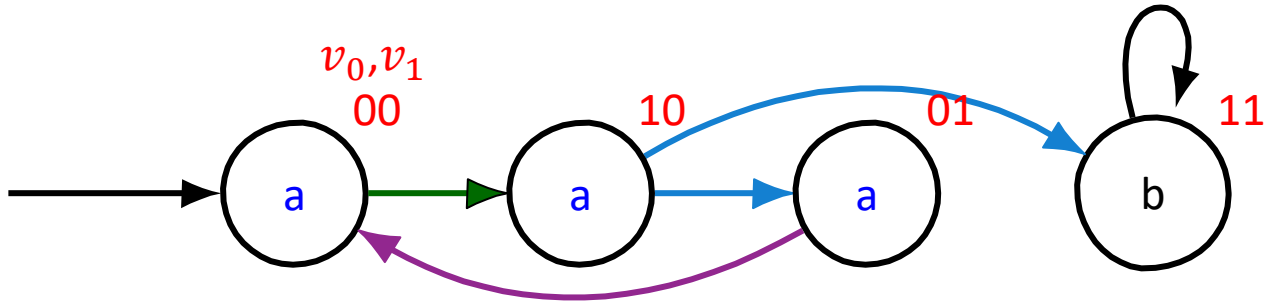
**AG a**

$$S_0(v_0, v_1) = \neg v_0 \wedge \neg v_1.$$

$$\begin{aligned} R(v_0, v_1, v'_0, v'_1) \\ &= \neg v_0 \wedge \neg v_1 \wedge v'_0 \wedge \neg v'_1 \vee v_0 \wedge \neg v_1 \wedge v'_1 \\ &\vee \neg v_0 \wedge v_1 \wedge \neg v'_0 \wedge \neg v'_1 \vee v_0 \wedge v_1 \wedge v'_0 \wedge v'_1 \end{aligned}$$

$$a(v_0, v_1) = \neg v_0 \vee \neg v_1.$$

# Example: k=0



$$S_0(v_0, v_1) = \neg v_0 \wedge \neg v_1.$$

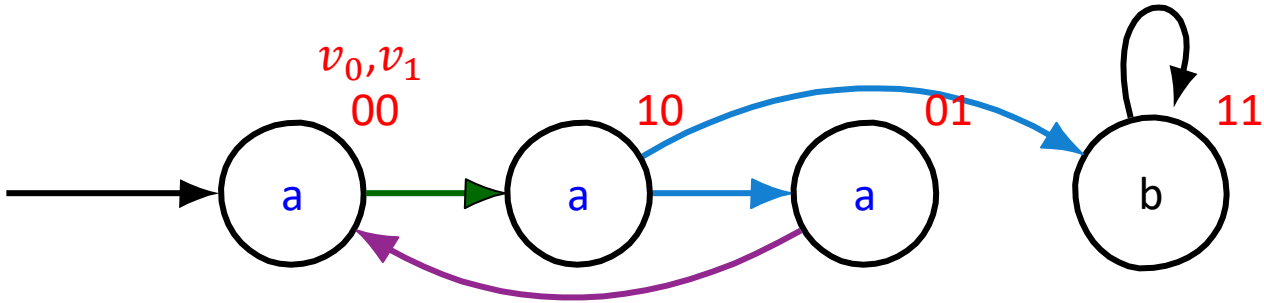
$$\begin{aligned} path_0(s_0, s_1, s_2) &= S_0(s_0) \\ &= \neg v_{0,0} \wedge \neg v_{1,0} \end{aligned}$$

$$\begin{aligned} R(v_0, v_1, v'_0, v'_1) &= \neg v_0 \wedge \neg v_1 \wedge v'_0 \wedge \neg v'_1 \vee v_0 \wedge \neg v_1 \wedge v'_1 \\ &\vee \neg v_0 \wedge v_1 \wedge \neg v'_0 \wedge \neg v'_1 \vee v_0 \wedge v_1 \wedge v'_0 \wedge v'_1 \end{aligned}$$

$$a(v_0, v_1) = \neg v_0 \vee \neg v_1.$$

$$\bigvee_{i=0}^k \neg a(s_i) = \neg(\neg v_{00} \vee \neg v_{10})$$

# Example: k=1



$$S_0(v_0, v_1) = \neg v_0 \wedge \neg v_1$$

$$R(v_0, v_1, v'_0, v'_1) = \neg v_0 \wedge \neg v_1 \wedge v'_0 \wedge \neg v'_1 \vee v_0 \wedge \neg v_1 \wedge v'_1 \vee \neg v_0 \wedge v_1 \wedge \neg v'_0 \wedge \neg v'_1 \vee v_0 \wedge v_1 \wedge v'_0 \wedge v'_1$$

$$a(v_0, v_1) = \neg v_0 \vee \neg v_1$$

$v_{10}$  is variable  $v_1$  in time 0

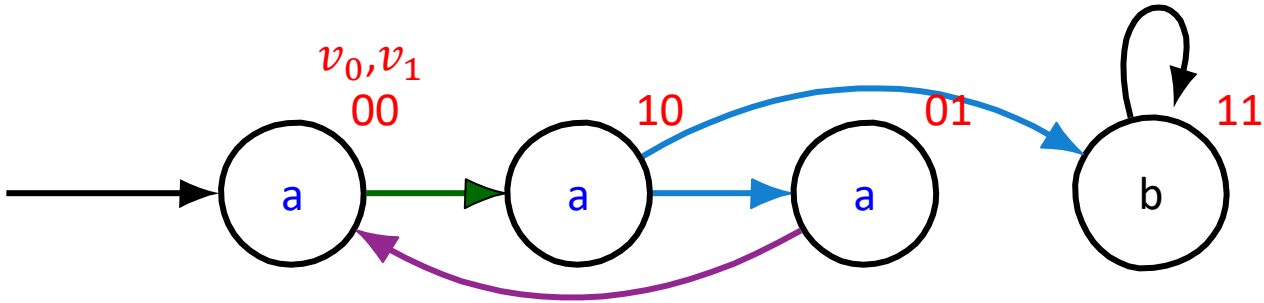
$$path_1(s_0, s_1, s_2) = S_0(s_0) \wedge R(s_0, s_1)$$

$$= \neg v_{0,0} \wedge \neg v_{1,0} \wedge$$

$$\left( \begin{array}{l} \neg v_{00} \wedge \neg v_{10} \wedge v_{01} \wedge \neg v_{11} \\ \vee v_{00} \wedge \neg v_{10} \wedge v_{11} \\ \vee \neg v_{00} \wedge v_{10} \wedge \neg v_{01} \wedge \neg v_{11} \\ \vee v_{00} \wedge v_{10} \wedge v_{01} \wedge v_{11} \end{array} \right)$$

$$\bigvee_{i=0}^k \neg a(s_i) = \neg(\neg v_{00} \vee \neg v_{10}) \vee \neg(\neg v_{01} \vee \neg v_{11})$$

# Example: k=2



$$S_0(v_0, v_1) = \neg v_0 \wedge \neg v_1$$

$$R(v_0, v_1, v'_0, v'_1) = \neg v_0 \wedge \neg v_1 \wedge v'_0 \wedge \neg v'_1 \vee v_0 \wedge \neg v_1 \wedge v'_1 \vee \neg v_0 \wedge v_1 \wedge \neg v'_0 \wedge \neg v'_1 \vee v_0 \wedge v_1 \wedge v'_0 \wedge v'_1$$

$$a(v_0, v_1) = \neg v_0 \vee \neg v_1$$

$$path_2(s_0, s_1, s_2) = S_0(s_0) \wedge R(s_0, s_1) \wedge R(s_1, s_2)$$

$$= \neg v_{0,0} \wedge \neg v_{1,0} \wedge$$

$$\left( \begin{array}{l} \neg v_{00} \wedge \neg v_{10} \wedge v_{01} \wedge \neg v_{11} \\ \vee v_{00} \wedge \neg v_{10} \wedge v_{11} \\ \vee \neg v_{00} \wedge v_{10} \wedge \neg v_{01} \wedge \neg v_{11} \\ \vee v_{00} \wedge v_{10} \wedge v_{01} \wedge v_{11} \end{array} \right) \wedge$$

$$\left( \begin{array}{l} \neg v_{01} \wedge \neg v_{11} \wedge v_{02} \wedge \neg v_{12} \\ \vee v_{01} \wedge \neg v_{11} \wedge v_{12} \\ \vee \neg v_{01} \wedge v_{11} \wedge \neg v_{02} \wedge \neg v_{12} \\ \vee v_{01} \wedge v_{11} \wedge v_{02} \wedge v_{12} \end{array} \right)$$

$$\bigvee_{i=0}^k \neg a(s_i) = \neg(\neg v_{00} \vee \neg v_{10}) \vee \neg(\neg v_{01} \vee \neg v_{11}) \vee \neg(\neg v_{02} \vee \neg v_{12})$$

# Try it with Z3!

```
(declare-const v00 Bool)
(declare-const v10 Bool)
(declare-const v01 Bool)
(declare-const v11 Bool)
(declare-const v02 Bool)
(declare-const v12 Bool)

(define-fun S0 ((v0 Bool) (v1 Bool)) Bool
  (and (not v0) (not v1)))

(define-fun R ((v0 Bool) (v1 Bool) (w0 Bool) (w1 Bool))
  Bool
  (or
   (and (not v0) (not v1) w0 (not w1))
   (and v0 (not v1) w1)
   (and (not v0) v1 (not w0) (not w1))
   (and v0 v1 w0 w1))
  )

(define-fun a ((v0 Bool) (v1 Bool)) Bool
  (or (not v0) (not v1))
  )

(define-fun path1 ((v00 Bool) (v10 Bool) (v01 Bool) (v11
  Bool)) Bool
  (and (S0 v00 v10)
   (R v00 v10 v01 v11))
  )
)
```

```
(define-fun path2 ((v00 Bool) (v10 Bool) (v01 Bool) (v11
  Bool) (v02 Bool) (v12 Bool)) Bool
  (and (S0 v00 v10)
   (R v00 v10 v01 v11)
   (R v01 v11 v02 v12))
  )

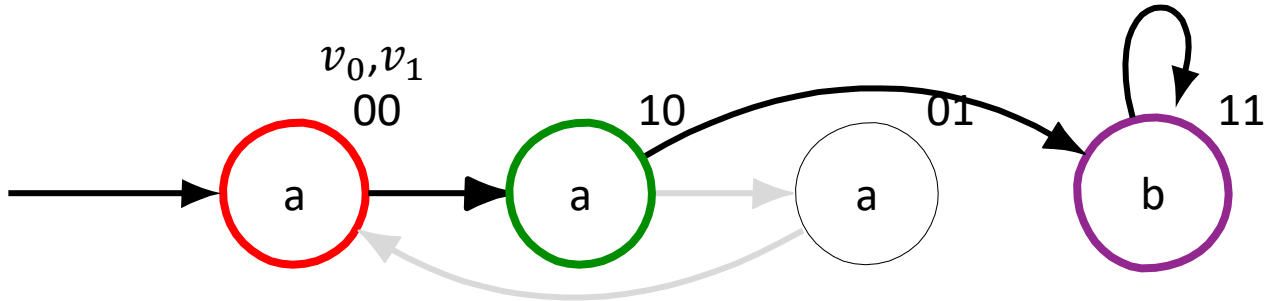
; k = 1
; (assert
;   (and (path1 v00 v10 v01 v11)
;        (or (not (a v00 v10))
;            (not (a v01 v11))
;            )
;   )
;)

; k = 2
; (assert
;   (and (path2 v00 v10 v01 v11 v02 v12)
;        (or (not (a v00 v10))
;            (not (a v01 v11))
;            (not (a v02 v12))
;            )
;   )
;)

(check-sat)
(get-model)
```



# Example



**AG a**

$$\begin{aligned}
 path_2(s_0, s_1, s_2) &= S_0(s_0) \wedge R(s_0, s_1) \wedge R(s_1, s_2) \\
 &= \neg v_{0,0} \wedge \neg v_{1,0} \wedge \\
 &\quad \left( \begin{array}{l} \neg v_{00} \wedge \neg v_{10} \wedge v_{01} \wedge \neg v_{11} \\ \vee v_{00} \wedge \neg v_{10} \wedge v_{11} \\ \vee \neg v_{00} \wedge v_{10} \wedge \neg v_{01} \wedge \neg v_{11} \\ \vee v_{00} \wedge v_{10} \wedge v_{01} \wedge v_{11} \end{array} \right) \wedge \\
 &\quad \left( \begin{array}{l} \neg v_{01} \wedge \neg v_{11} \wedge v_{02} \wedge \neg v_{12} \\ \vee v_{01} \wedge \neg v_{11} \wedge v_{12} \\ \vee \neg v_{01} \wedge v_{11} \wedge \neg v_{02} \wedge \neg v_{12} \\ \vee v_{01} \wedge v_{11} \wedge v_{02} \wedge v_{12} \end{array} \right)
 \end{aligned}$$

Satisfying assignment

i	0	1	2
$v_{0i}$	0	1	1
$v_{1i}$	0	0	1

$$\bigvee_{i=0}^k \neg a(s_i) = \neg(\neg v_{00} \vee \neg v_{10}) \vee \neg(\neg v_{01} \vee \neg v_{11}) \vee \neg(\neg v_{02} \vee \neg v_{12})$$

## Another Example

Does modulo-8 counter of Chapter 3.5 ever reach 4?

$$\neg p = \neg v_0 \wedge \neg v_1 \wedge v_2.$$

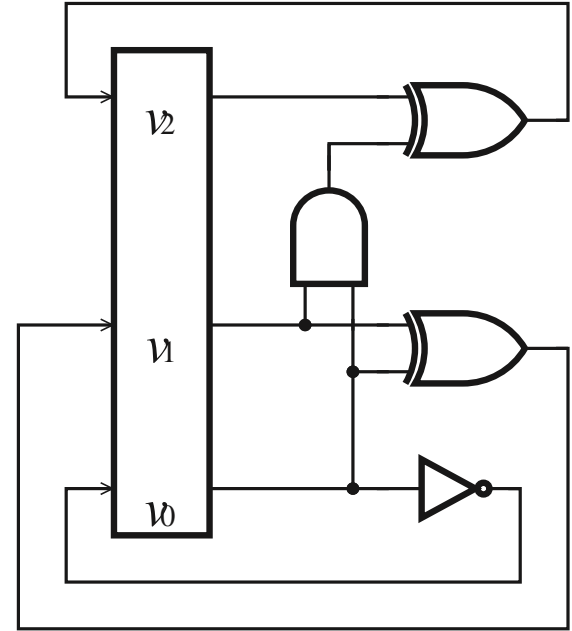
$$S_0(V) = \neg v_0 \wedge \neg v_1 \wedge \neg v_2.$$

$$R(V, V') = (v'_0 \leftrightarrow \neg v_0) \wedge (v'_1 \leftrightarrow v_0 \oplus v_1) \wedge (v'_2 \leftrightarrow (v_0 \wedge v_1) \oplus v_2).$$

**$k = 0$ :**

$$S_0(v) \quad \neg v_0 \wedge \neg v_1 \wedge \neg v_2 \wedge$$

$$\neg p(v) \quad \neg v_0 \wedge \neg v_1 \wedge v_2.$$



# Another Example

Does module-8 counter of Chapter 3.5 ever reach 4?

$$\neg p = \neg v_0 \wedge \neg v_1 \wedge v_2.$$

$$S_0(V) = \neg v_0 \wedge \neg v_1 \wedge \neg v_2.$$

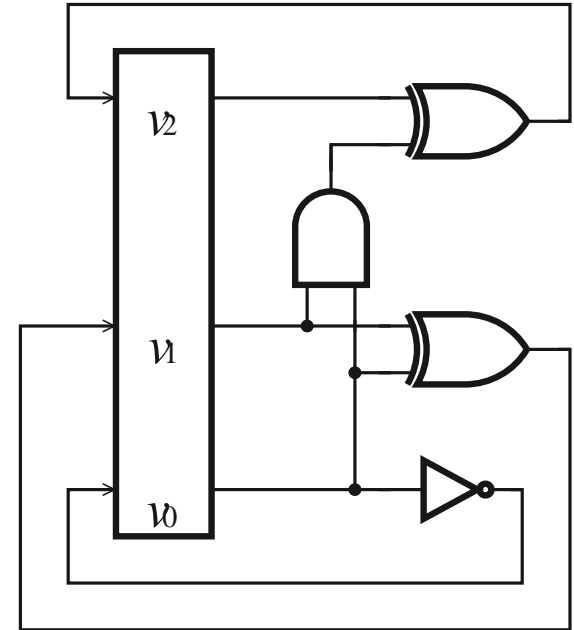
$$R(V, V') = (v'_0 \leftrightarrow \neg v_0) \wedge (v'_1 \leftrightarrow v_0 \oplus v_1) \wedge (v'_2 \leftrightarrow (v_0 \wedge v_1) \oplus v_2).$$

**$k = 1$ :**

$$S_0(v) \quad \neg v_0 \wedge \neg v_1 \wedge \neg v_2 \wedge$$

$$R(v, v') \quad (v'_0 \leftrightarrow \neg v_0) \wedge (v'_1 \leftrightarrow v_0 \oplus v_1) \wedge (v'_2 \leftrightarrow (v_0 \wedge v_1) \oplus v_2) \wedge$$

$$(\underbrace{\neg v_0 \wedge \neg v_1 \wedge v_2}_{\neg p(v)} \vee \underbrace{\neg v'_0 \wedge \neg v'_1 \wedge v'_2}_{\neg p(v')}).$$



# Another Example

Does module-8 counter of Chapter 3.5 ever reach 4?

$$\neg p = \neg v_0 \wedge \neg v_1 \wedge v_2.$$

$$S_0(V) = \neg v_0 \wedge \neg v_1 \wedge \neg v_2.$$

$$R(V, V') = (v'_0 \leftrightarrow \neg v_0) \wedge (v'_1 \leftrightarrow v_0 \oplus v_1) \wedge (v'_2 \leftrightarrow (v_0 \wedge v_1) \oplus v_2).$$

**k = 2:**

$$S_0(v) \quad \neg v_0 \wedge \neg v_1 \wedge \neg v_2 \wedge$$

$$R(v, v') \quad (v'_0 \leftrightarrow \neg v_0) \wedge (v'_1 \leftrightarrow v_0 \oplus v_1) \wedge (v'_2 \leftrightarrow (v_0 \wedge v_1) \oplus v_2) \wedge$$

$$R(v', v'') \quad (v''_0 \leftrightarrow \neg v'_0) \wedge (v''_1 \leftrightarrow v'_0 \oplus v'_1) \wedge (v''_2 \leftrightarrow (v'_0 \wedge v'_1) \oplus v'_2) \wedge$$

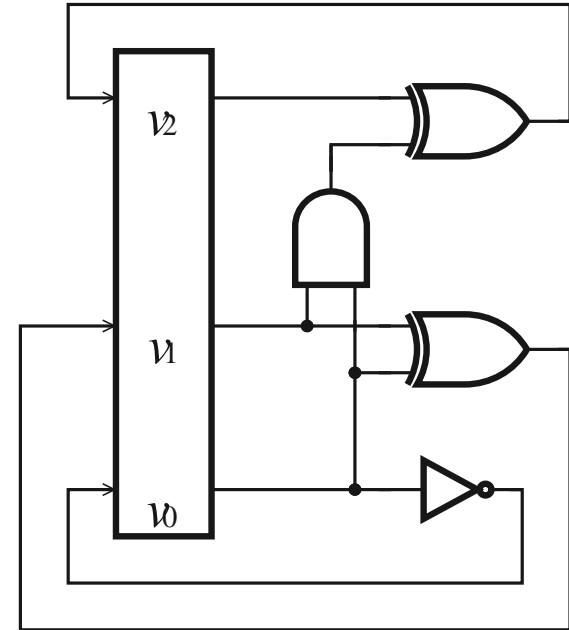
$$(\neg v_0 \wedge \neg v_1 \wedge v_2 \vee \neg v'_0 \wedge \neg v'_1 \wedge v'_2 \vee \neg v''_0 \wedge \neg v''_1 \wedge v''_2.)$$

$\neg p(v)$

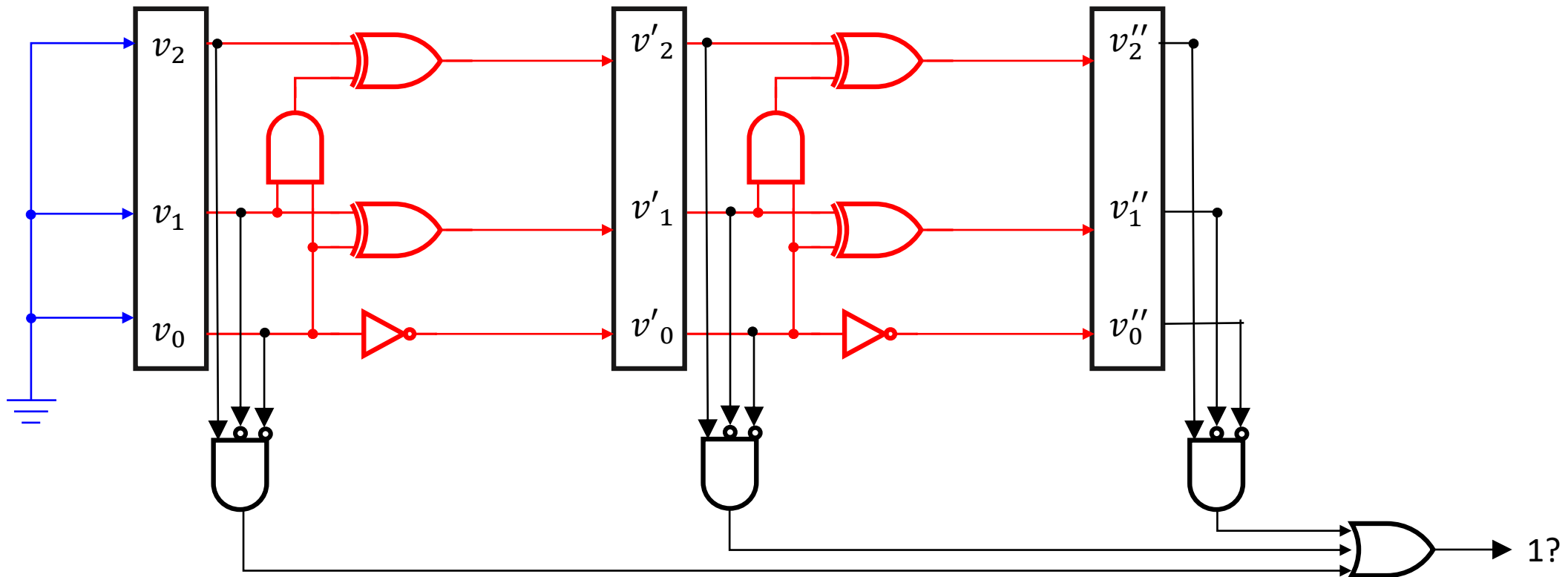
$\neg p(v')$

Model Checking

$\neg p(v'')$



# The Electrical Engineer's Viewpoint



# Complete?

# Completeness

BMC finds bugs of length  $\leq k$ .

Longer counterexamples may exist!

Is there a  $k$  big enough to exclude any counterexample?

**Def.**  $M \models_k \phi$ : all computations of length  $k$  satisfy  $\phi$ .

**Completeness threshold:** Number  $CT$  such that  $M \models_{CT} \phi \Rightarrow M \models \phi$

If completeness threshold known, stop BMC when  $k = CT$

*Ideas for finding  $CT$ ?*

# Completeness Threshold

Finding smallest  $CT$  is as difficult as model checking!

- Smallest  $CT$  is size of shortest counterexample, or 0 if the property is satisfied

$$New(V_0, \dots, V_k) = S_0(V_0) \wedge \bigwedge_{i=0}^{k-1} (R(V_i, V_{i+1})) \wedge \bigwedge_{j<i} V_i \neq V_j$$

Simple values for  $CT$

- Number of state of  $M$  is bound on  $CT$
- Diameter (longest simple path between two states) is bound on  $CT$

*These are typically very large numbers*



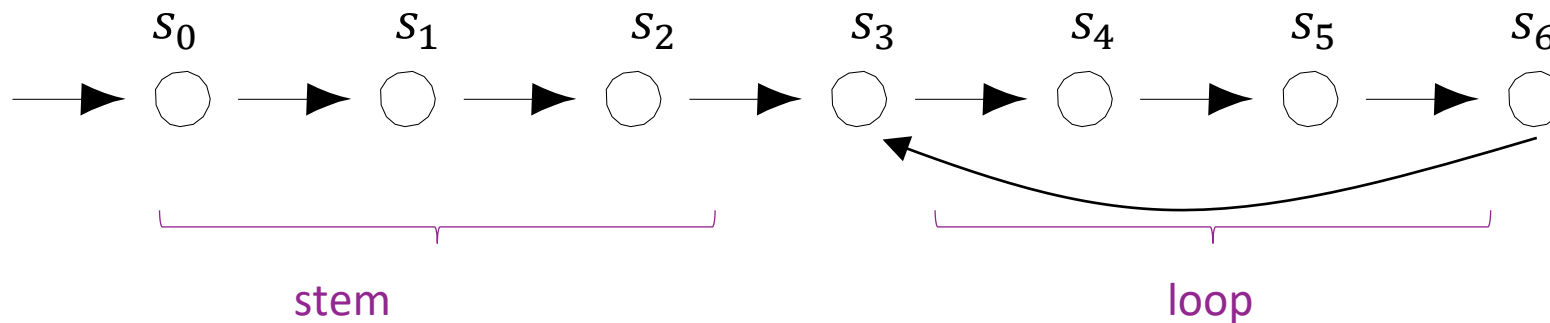
# Eventuality Properties

## Property

Suppose  $\phi = \mathbf{AF} p$

Counterexamples fulfill  $\mathbf{EG} \neg p$

Counterexamples have *Lasso Shape* (See Chapter 4.)



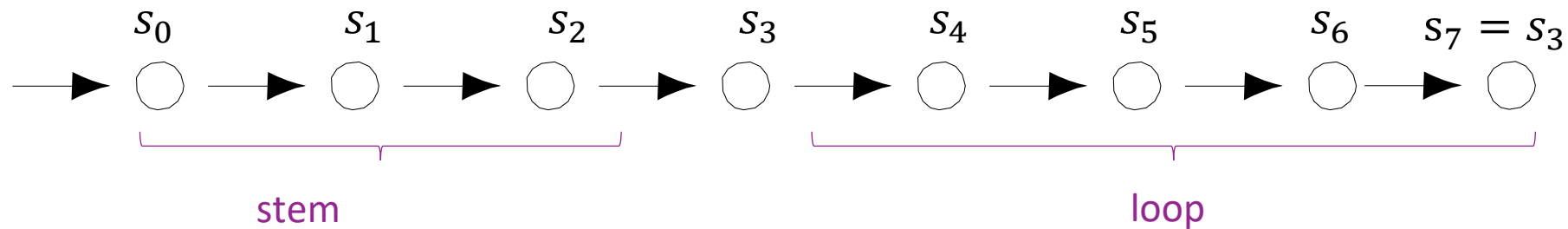
# Eventuality Properties

## Property

Suppose  $\phi = \mathbf{AF} p$  (all paths eventually reach  $p$ )

Counterexamples fulfill  $\mathbf{EG} \neg p$  (for some path  $\neg p$  always holds)

Counterexamples have *Lasso Shape* (See Chapter 4.)



$$lasso_k(s_0, \dots, s_k) = path_k(s_0, \dots, s_k) \wedge \bigvee_{i=0}^{k-1} s_i = s_k$$

# Eventuality Properties

## Property

Suppose  $\phi = \mathbf{AF} p$

Counterexamples fulfill  $\mathbf{EG} \neg p$

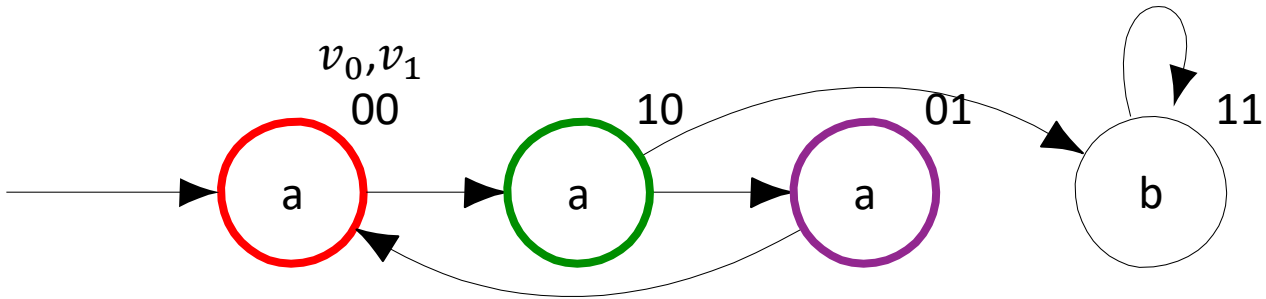
Counterexamples have *Lasso Shape* (See Chapter 4.)

$$\mathit{lasso}_k(s_0, \dots, s_k) = \mathit{path}_k(s_0, \dots, s_k) \wedge \bigvee_{i=0}^{k-1} s_i = s_k$$

Counterexample lasso:

$$\mathit{lasso}_k(s_0, \dots, s_k) \wedge \bigwedge_{i=0}^{k-1} \neg p$$

# Example



**AF  $b$**

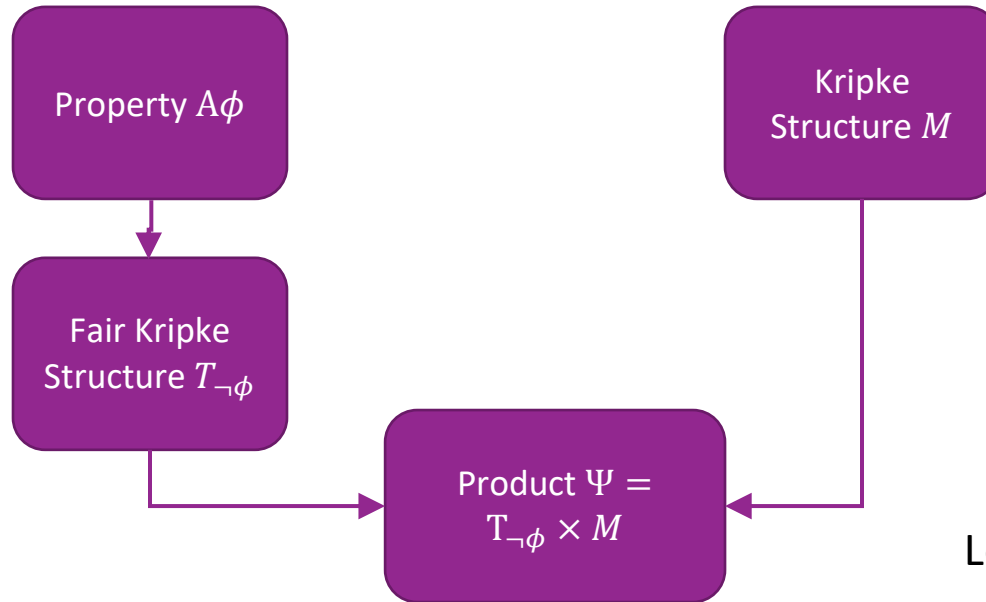
$$\begin{aligned}
 & \text{lasso}_3(s_0, \dots, s_3) = \\
 & \text{path}_3(s_0, \dots, s_3) \wedge \bigvee_{i=0}^2 s_i = s_3 \wedge \bigwedge_{i=0}^2 \neg b
 \end{aligned}$$

Satisfying assignment

i	0	1	2	3
$v_{0i}$	0	1	1	0
$v_{1i}$	0	0	1	0

# Full LTL

See Chapter 6



Let  $\Psi = (S^\Psi, S_0^\Psi, R^\Psi, AP^\Psi, L^\Psi, F^\Psi)$

$$S_0^\Psi(s_0) \wedge \bigwedge_{i=0}^{k-1} R^\Psi(s_i, s_{i+1}) \wedge \bigvee_{i=0}^{k-1} (s_i = s_k \wedge Fair_i^P)$$

**Satisfiable iff there is a fair lasso of length  $k$**

