COL:750

Foundations of Automatic Verification

Instructor: Priyanka Golia

Course Webpage 

https://priyanka-golia.github.io/teaching/COL-750/index.html

Consider a Kripke structure over propositional variables $\{a, b, c\}$ with states $s_o, \ldots, s_7$ corresponding to the 8 possible valuations of these variables. Start state is s_0.

The transition relation is

$$\mathrm{T}\,(a, b, c, a', b', c') = (a' \leftrightarrow b) \wedge (b' \leftrightarrow c)$$

for each state s_i, the binary representation of i gives the truth values of a,b,c
that is, L(s_i) is the corresponding set of literals.

For example — $L(s_o) = \neg a, \neg b, \neg c$

For example — $L(s_4) = a, \neg b, \neg c$

Check if the Kripke structure satisfies
$\forall \Box \neg a \vee \neg b \vee \neg c$ using BMC for k = 3.

$$\mathrm{T}(a, b, c, a', b', c') = (a' \leftrightarrow b) \wedge (b' \leftrightarrow c) \qquad k = 3, \ \forall \Box \neg a \vee \neg b \vee \neg c$$

$$CheckSAT((\neg a_o \wedge \neg b_o \wedge \neg c_o) \ \wedge (a_1 \leftrightarrow b_o) \wedge (b_1 \leftrightarrow c_o) \wedge (a_2 \leftrightarrow b_1) \wedge (b_2 \leftrightarrow c_1) \ \wedge (a_3 \leftrightarrow b_2) \wedge (b_3 \leftrightarrow c_2)$$

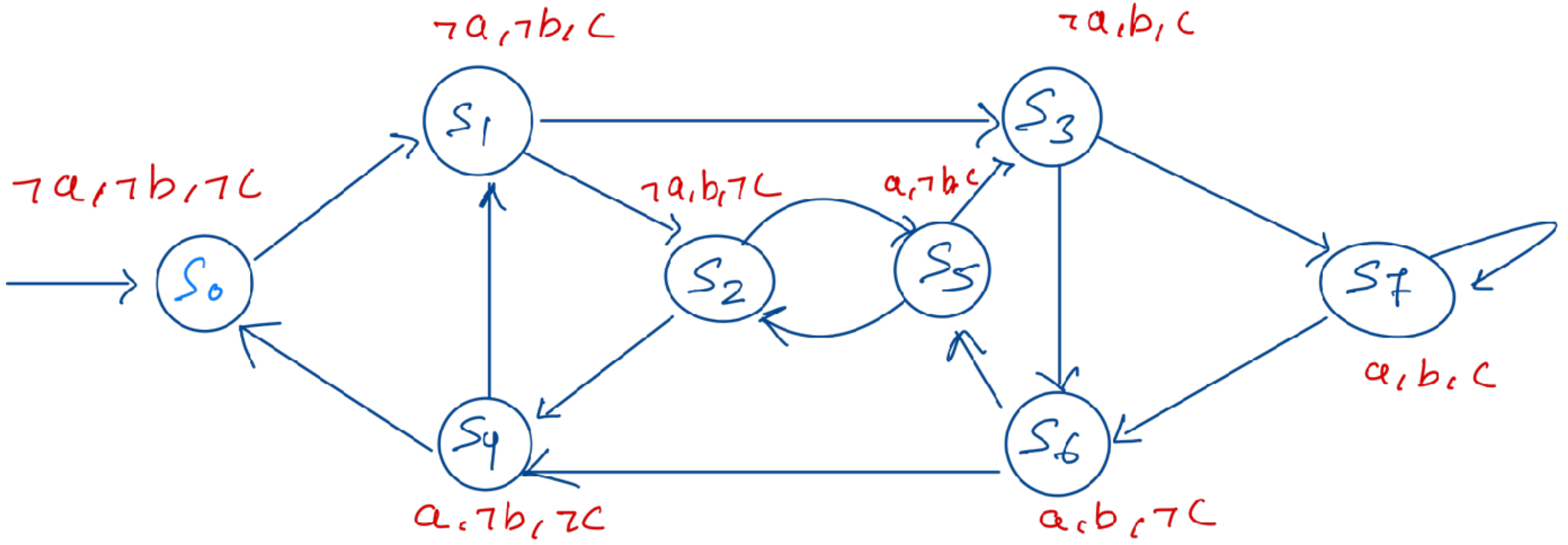
$$\wedge ((a_1 \wedge b_1 \wedge c_1) \vee (a_2 \wedge b_2 \wedge c_2) \vee (a_3 \wedge b_3 \wedge c_3)))$$

SAT - <

$$a_o = 0, b_o = 0, c_o = 0, a_1 = 0, b_1 = 0, c_1 = 1, a_2 = 0, b_2 = 1, c_2 = 1, a_3 = 1, b_2 = 1, c_3 = 1 >$$

$$< s_o, s_1, s_3, s_7 >$$

Notice we didn't need to draw the state diagram!
Now, think about the case when we have 100/1000/10,000 variables, number of states $2^{|Vars|}$ — in this storing and working with state diagram is not right!
Hence, SAT solvers will be useful



$$\mathrm{T}(a, b, c, a', b', c') = (a' \leftrightarrow b) \wedge (b' \leftrightarrow c)$$

$$\forall \Box \neg a \vee \neg b \vee \neg c$$

Bounded Model Checking with SAT (BMC)

Diameter of M

Given a model M, the diameter of M is a completeness threshold for any property of the form $\forall \Box p$

Safety properties — something always holding.

Counterexample — ($\exists \Diamond \neg p$) can we find a bad state in k step?

For Safety property, d is a completeness threshold.

Bounded Model Checking with SAT (BMC)

Diameter of M

The diameter of a Kripke structure is the longest shortest path between any two reachable states. Formally:

$$\text{Diameter(M)} = \underset{\forall s,s' \in T}{Max}\ ShortestPathLength(s, s')$$

How to check if $K \overset{?}{\geq} d$?

State s is reachable in j steps:

$$R_j(s) = \exists s_o, \ldots, s_j\ s\ .\ t\ .\ s_j \wedge I(s_o) \wedge \bigwedge_{i=o}^{j-1} T(s_i, s_{i+1})$$

K is greater than or equal to Diameter d if

$$\forall s : R_{k+1}(s) \rightarrow \exists j \leq k\ R_j(s)$$

For all states, does there exists a path of length at most k?

Computationally hard problem — requires QBF calls.

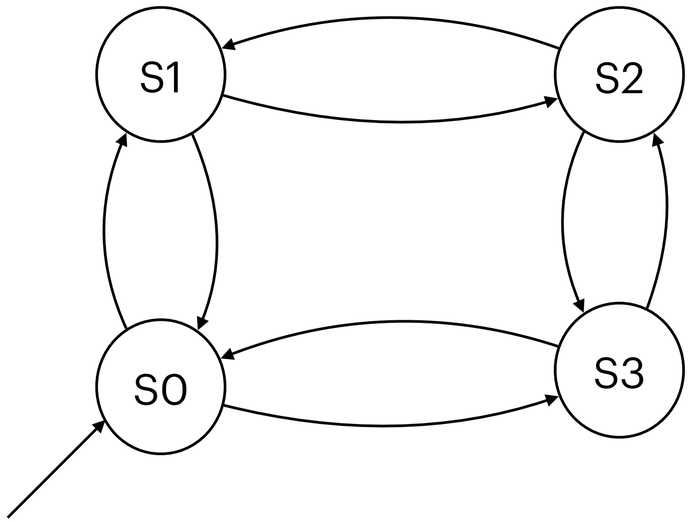
Bounded Model Checking with SAT (BMC)

Recurrence Diameter: the longest length of a path starting from the initial state **without repeating any state.**

rd is the longest loop-free path in M.

Recurrence Diameter (*rd*) is an upper bound for the diameter d.

$d = 2 \quad rd = 3$



This means that after *rd* steps, either:

All reachable states have been visited.

Any further steps must repeat a previously visited state.

Bounded Model Checking with SAT (BMC)

Recurrence Diameter: the longest length of a path starting from the initial state **without repeating any state**

rd is the longest loop-free path in M.

Recurrence Diameter (rd) is an upper bound for the diameter d.

How to check if $K \overset{?}{\geq} rd$

1. To ensure the path remains acyclic, we need to enforce that no two states along the path are the same:
    1. $$\forall s_0, \ldots, s_k : I(s_o) \wedge \bigwedge_{i=0}^{k-1} T(s_i, s_{i+1}) \rightarrow \bigvee_{l=0}^{k-1} \bigvee_{j=l+1}^{k} s_l = s_j$$

Expensive SAT call.

Smallest k for which this is UNSAT is $\geq rd$

Bounded Model Checking with SAT (BMC)

General idea:
Fix a K

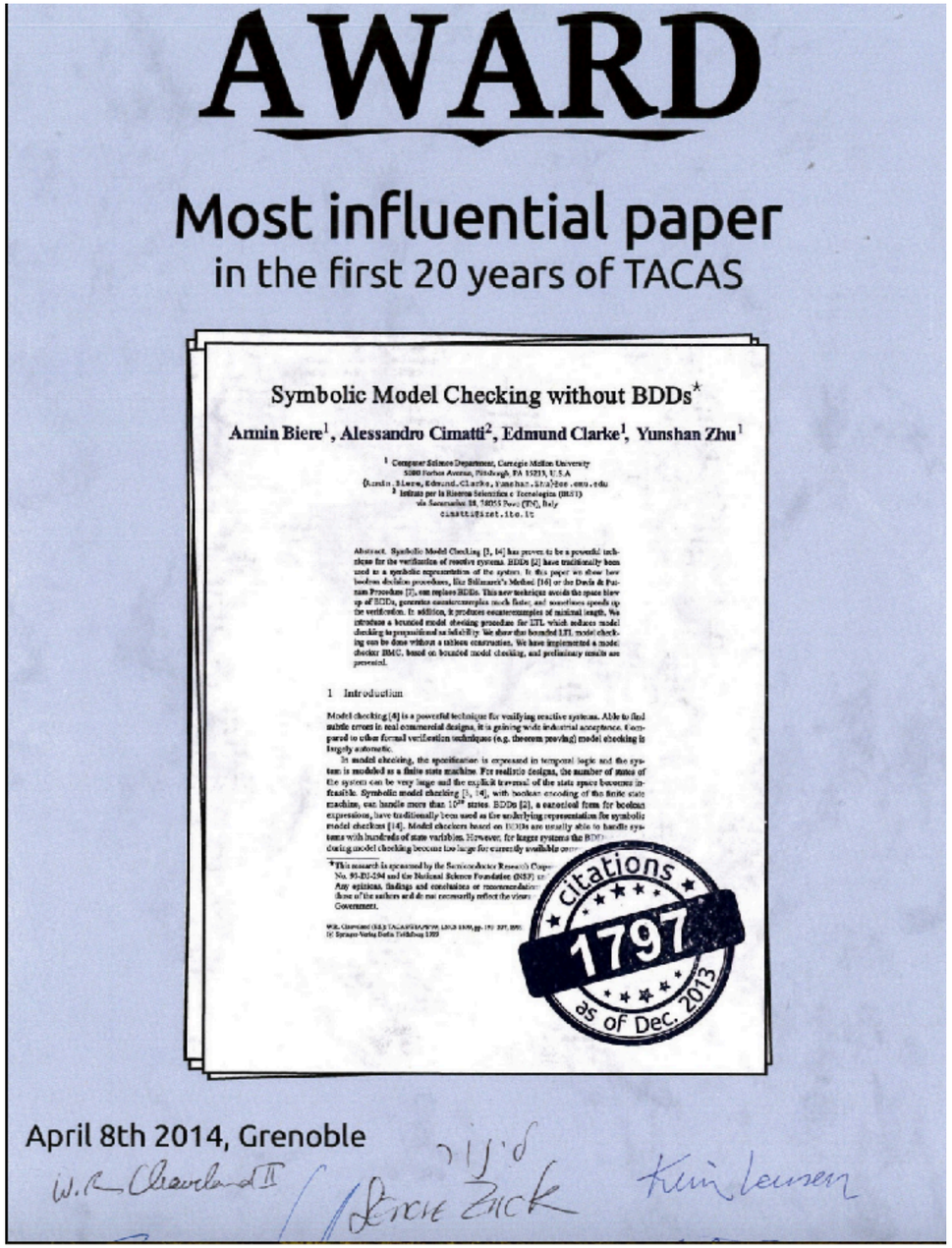
1. Convert transition system to propositional encoding — unroll for path length k
2. Convert temporal formula along the states to propositional encoding for k length
3. Using SAT Solvers look for counterexamples
4. Found a counterexample :

   Return counterexample

5. Else:

   K= K+1

6. At some point, check if $K \overset{?}{\geq} rd$ Return True, Else: K = K+1 For safety property.



extensions to completeness

diameter checking,

k-induction,

interpolation –

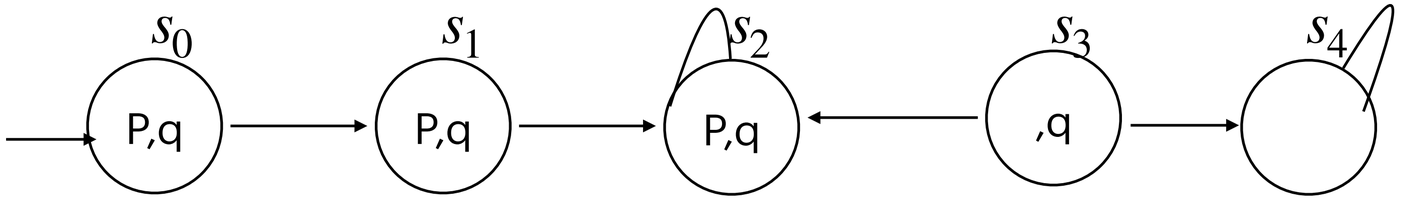
SAT based model checking without unrolling: IC3

Induction

For verifying safety property/ verifying reachability properties.

Often the completeness threshold is very large.

Exploring techniques that requires fewer unwinding.



$M \models \forall \Box p \qquad M \models \forall \Box q$

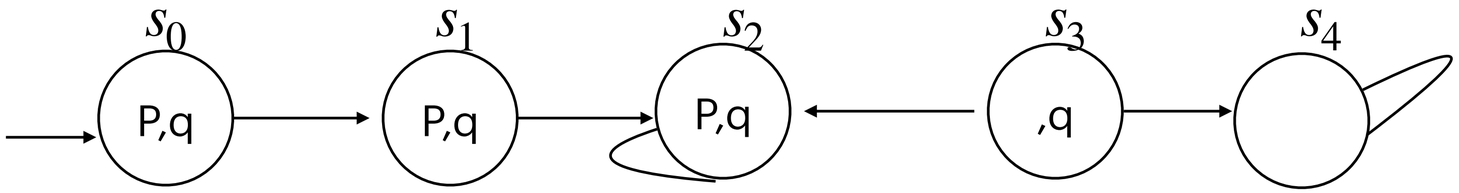
Induction principles —

To prove the claim $Q(n)$ for all values of some parameter n.

We need to show the validity of these cases	Base case — Q(0) Inductive step case — Q(n-1) -> Q(n)

Induction

For verifying safety property/ verifying reachability properties.



$M \models \forall \Box p$

Induction principles

To prove $\forall \Box p$, we prove that $p(\pi(n))$ holds for $\forall n$

Given M, π denotes a path in M.

i^{th} state in π is $\pi(i)$.

$p(\pi(i))$ is to denote that property p holds in state $\pi(i)$

Idea — base case (initial states). $p(s_0)$ holds.

Inductive step. Assuming $p(\pi(n-1))$ holds, $p(\pi(n))$ must hold.

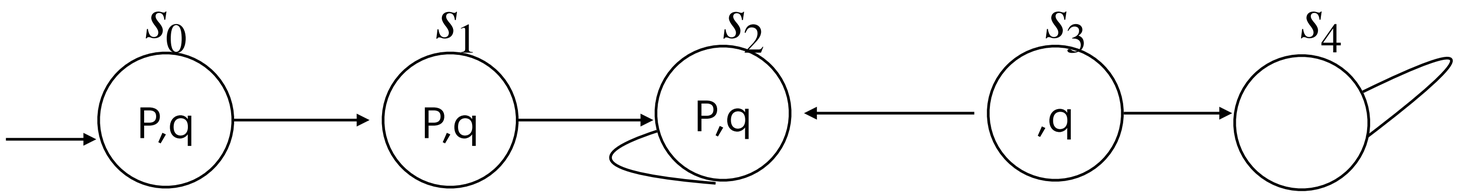
all the states labelled with p, that is, $\{0,1,2,\}$

All the states where $p(\pi(n-1))$ holds!

$p(\pi(n))$ must hold true, which will be successor of $\{0,1,2\} - \{1,2\}$

Induction

For verifying safety property/ verifying reachability properties.



$M \models \forall \square p$

To prove $\forall \square p$, we prove that $p(\pi(n))$ holds for $\forall n$

Idea — base case (initial states). $p(s_0)$ holds.

Inductive step. $p(\pi(n-1))$ holds, states labelled with p, that is, $\{0,1,2\}$

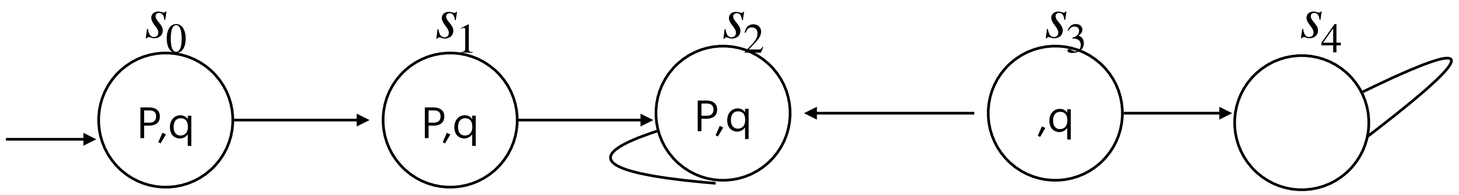
$p(\pi(n))$ must hold true, which will be successor of $\{0,1,2\} - \{1,2\}$

Validity of the base case and inductive step?

Validity of F $\equiv \neg F$ being UNSAT.

Induction

For verifying safety property/ verifying reachability properties.



$M \models \forall \square p$

To prove $\forall \square p$, we prove that $p(\pi(n))$ holds for $\forall n$

For base case, check satisfiability of

$$s_o \wedge \neg p(s_o)\ \forall s_o \in I \qquad (p_o \wedge q_o) \wedge \neg p_o$$

If this is UNSAT, then all initial state satisfy p.

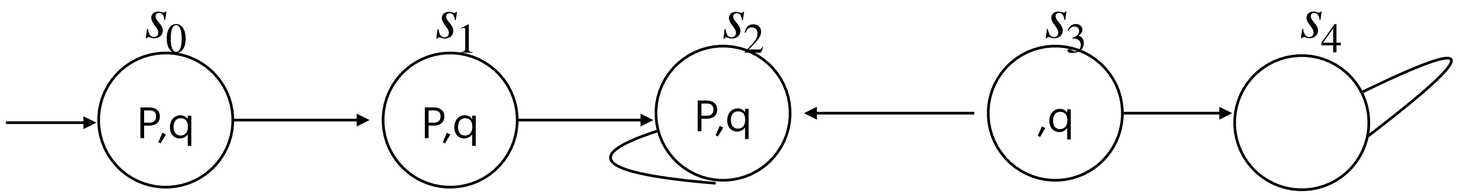
Inductive case — observation $T(\pi(n-1), \pi(n))$ holds.

Let s be the states in $\pi(n-1)$ Let s' be the states in $\pi(n)$

Validity of $p(s) \wedge T(s, s') \rightarrow p(s')$ CheckSAT($p(s) \wedge T(s, s') \wedge \neg p(s')$)

Induction

For verifying safety property/ verifying reachability properties.



$M \models \forall \square p$

To prove $\forall \square p$, we prove that $p(\pi(n))$ holds for $\forall n$

Inductive case — observation $T(\pi(n-1), \pi(n))$ holds.

Let s be the states in $\pi(n-1)$ Let s' be the states in $\pi(n)$

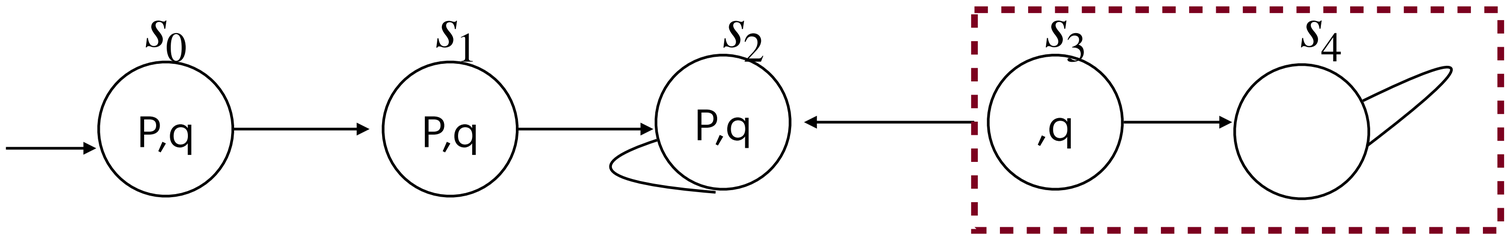
Validity of $p(s) \wedge T(s, s') \rightarrow p(s')$ CheckSAT($p(s) \wedge T(s, s') \wedge \neg p(s'))$ $\forall s \in S$, s.t. $P(s)$

Inductive steps — any reachable state in model M

CheckSAT$((p_o \wedge \neg p_1') \vee (p_1 \wedge \neg p_2') \vee (p_2 \vee \neg p_3') \wedge T)$

Induction

For verifying safety property/ verifying reachability properties.



$M \models \forall \square q$

To prove $\forall \square q$, we prove that $q(\pi(n))$ holds for $\forall n$

Base case: $q_o \wedge \neg q_o$ UNSAT

Inductive case: $((q_o \wedge \neg q_1') \vee (q_1 \wedge \neg q_2') \vee (q_2 \wedge \neg q_2') \vee (q_3 \wedge \neg q_4') \vee (q_3 \wedge \neg q_2')) \wedge T$ SAT

$M \stackrel{?}{\models} \forall \square q$

Inductive step case $-q(\pi(n-1)) \rightarrow q(\pi(n))$

Inductive case also deals with unreachable states

K-Induction For verifying safety property/ verifying reachability properties.

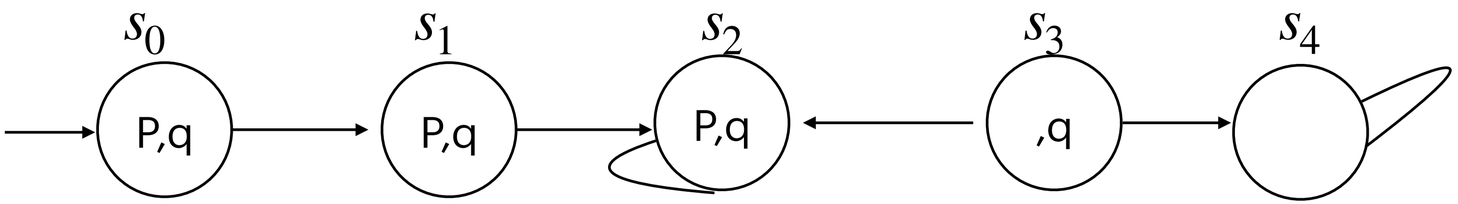
we strengthen the criterion for base case, and weaken the criterion for step case.

Input K, M, F/P

Base case — $p(0) \wedge \ldots \wedge p(K-1)$

Step case — $p(n-K) \wedge \ldots \wedge p(n-1) \rightarrow p(n)$

Any path with k states labelled with p,
is followed by a state labelled with p.



$M \models \forall \Box q$ K = 2

K-Induction For verifying safety property/ verifying reachability properties.

Base case — $p(0) \wedge \ldots \wedge p(K-1)$ Step case — $p(n-K) \wedge \ldots \wedge p(n-1) \rightarrow p(n)$

Base case — property holds in K states starting from initial state Same as BMC

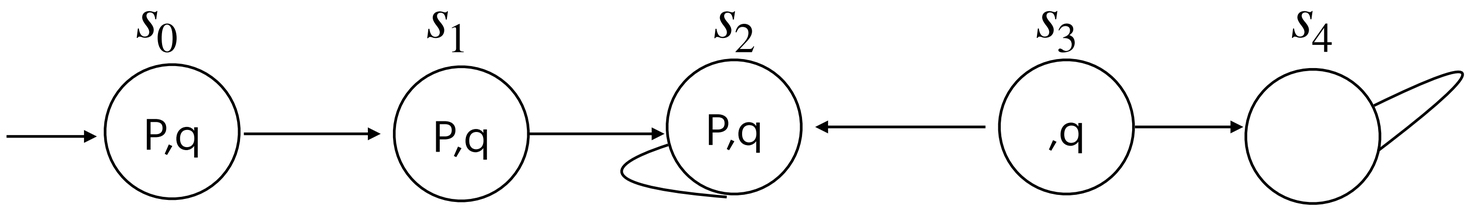
Property q holds true in {0,1}

Inductive step — we need to consider all paths with two states.

Property q holds true in {0,1}, {1,2},{2,2},{3,2}

For each of these paths, q holds in their successor

Property q holds true in {2}



$M \models \forall \Box q$ K = 2

K-Induction

For verifying safety property/ verifying reachability properties.

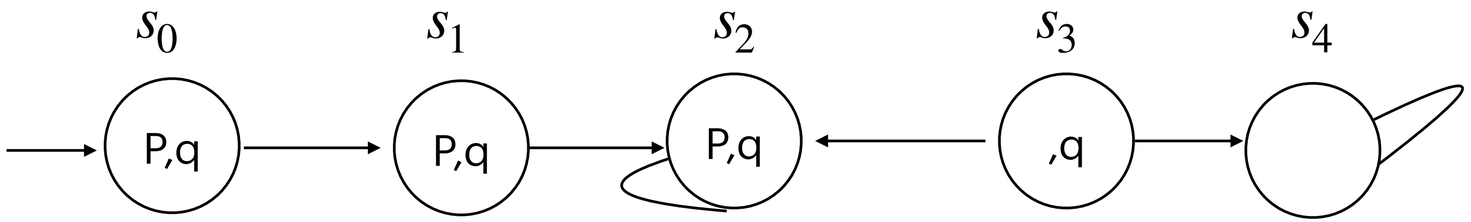
Base case — $p(0) \wedge \ldots \wedge p(K-1)$ Step case — $p(n-K) \wedge \ldots \wedge p(n-1) \rightarrow p(n)$

Base case — property holds in K states starting from initial state Same as BMC

$$M_k \wedge \neg p_k$$

Inductive step — we need to consider all paths with K states.

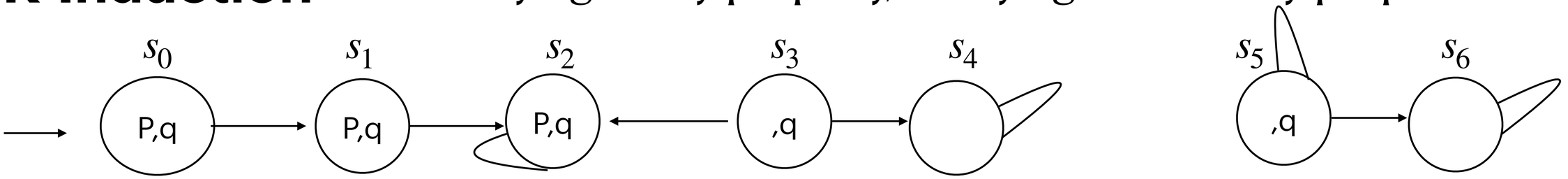
$$\bigwedge_{i=0}^{k-1} T(s_i, s_{i+1}) \wedge \bigwedge_{i=0}^{k-1} ((p(s_i)) \wedge \neg p(s_k)$$



$M \models \forall \Box q$ K = 2

K-Induction

For verifying safety property/ verifying reachability properties.



$$M \overset{?}{\models} \forall \Box q$$

Neither of the two states are initial states
Nor are they reachable from an initial state.

$$\bigwedge_{i=0}^{k-1} T(s_i, s_{i+1}) \wedge \bigwedge_{i=0}^{k-1} ((p(s_i)) \wedge \neg p(s_k)$$

This should be UNSAT!

No, irrespective of K, this is SAT

$$s_o \mapsto 5, \ldots, s_{k+1} \mapsto 5, s_k \mapsto 6$$

Therefore, to obtain completeness — we need to add $\bigwedge_{i=0}^{k-1} \bigwedge_{j=i+1}^{k} s_i \neq s_j$

Ensuring simple path

K-Induction

For verifying safety property/ verifying reachability properties.

Base case — property holds in K states starting from initial state

$$M_k \wedge \neg p_k \equiv I(s_o) \wedge \bigwedge_{i=0}^{k-1} T(s_i, s_{i+1}) \wedge \neg p_k$$

Same as BMC

Inductive step — we need to consider all paths with K states.

$$\bigwedge_{i=0}^{k-1} T(s_i, s_{i+1}) \wedge \bigwedge_{i=0}^{k-1} ((p(s_i)) \wedge \neg p(s_k) \wedge \bigwedge_{i=0}^{k-1} \bigwedge_{j=i+1}^{k} s_i \neq s_j$$

- If Base case is SAT, return counterexample.
- If Inductive case is UNSAT, return True.
- Otherwise, increase K and continue.

K-Induction For verifying safety property/ verifying reachability properties.

k-induction extends the capabilities of BMC by not only detecting counterexamples within a bounded number of steps but also proving the absence of such counterexamples, thereby establishing the validity of properties over unbounded executions

Observations

1. We do not need to know the exact reachable states, as long as we can guarantee they meet the property .
2. Beginning of “Property directed” techniques — which is associated with a family of techniques that build inductive invariants automatically

Property Directed Reachability (PDR) is another name for IC3 (Incremental Construction of Inductive Clauses for Indubitable Correctness).
The phrase “property directed” refers to how IC3 works — it constructs inductive invariants incrementally, guided by the property being verified.