Verifying Reachability Properties with k-induction



Mary Sheeran, Koen Claessen, Per Bjesse, 2000

Make BMC Complete

New_{k}: There is a simple path of length k:

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

Increase k until following is unsatisfiable

Drawback: k can be very large

How do you prove $i < n + 1$ for the following program?

```
BigInt i;
i = 0;
while(true)
      if(i == n) i = 0;
      else i++;
```

Motivation

- Can we **prove** a property with fewer unrollings?
- Idea: Use induction.

Base: Prove $Q(0)$

Induction: Prove $Q(\mathrm{t}-1) \Rightarrow Q(\mathrm{t})$

Conclusion: $\forall \mathrm{t}.\, Q(\mathrm{t})$

Caveat: Property may be true, but not inductive

We will go through a series of algorithms until we find a nice one

Induction

Let's prove $\mathbf{AG}\ p$ on the following structure.

Take arbitrary path π

- **Base case:** $\pi(0) \vDash p$ — true: $q_1 \vDash p$
- **Induction:** if $\pi(n-1) \vDash p$ then $\pi(n) \vDash p$ — true: any successor of a p-state is a p-state
- **Conclusion**: for any path π we have $\forall n.\, \pi(n) \vDash p$

Induction

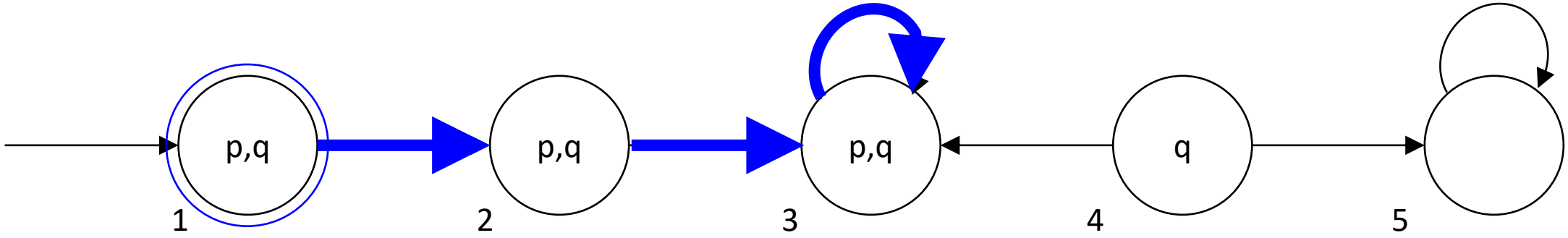
Let's prove **AG** p on the following structure.

Take arbitrary path π

- **Base case:** $\pi(0) \vDash p$
- **Induction:** if $\pi(n-1) \vDash p$ then $\pi(n) \vDash p$
- **Conclusion**: for any path π we have $\forall n.\, \pi(n) \vDash p$

true: $q_1 \vDash p$

true: any successor of a p-state is a p-state



Satisfiability

Let's prove $\mathrm{AG}\, p$ on the following structure.

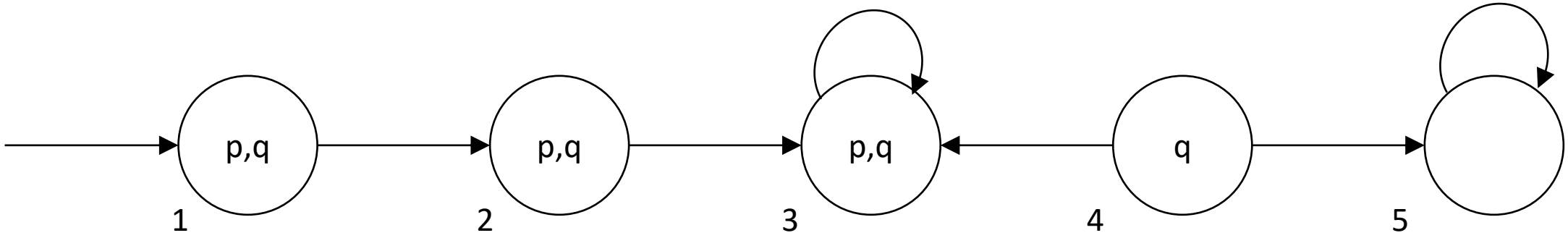
Take arbitrary path π

- **Base case:** $\pi(0) \vDash p$
- **Induction:** if $\pi(n-1) \vDash p$ then $\pi(n) \vDash p$
- **Conclusion**: for any path π we have $\forall n.\, \pi(n) \vDash p$

How can these properties be violated?

$S_o(s) \wedge \neg \mathrm{p}(s)$ Unsatisfiable

$p(s) \wedge R(s, s') \wedge \neg p(s')$ Unsatisfiable



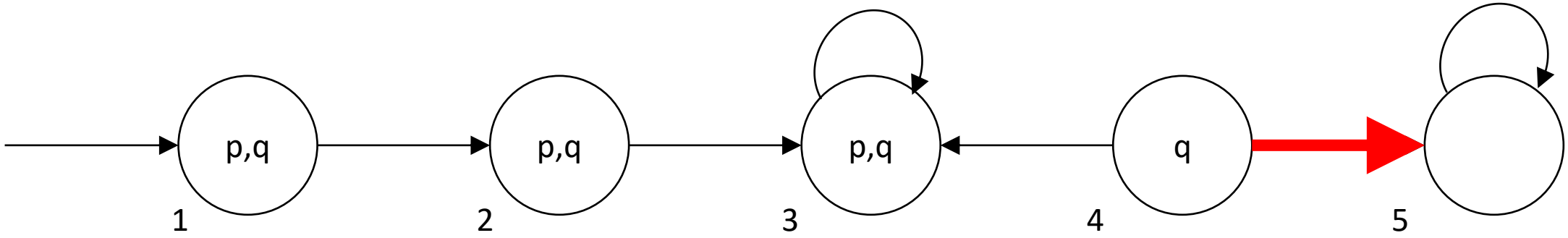
A Problem

Let's prove $\mathbf{AG}\ q$ on the following structure.

Take arbitrary path π

- **Base case:** $\pi(0) \vDash \mathrm{q}$
- ~~**Induction:** if $\pi(n-1) \vDash q$ then $\pi(n) \vDash q$~~ **not true!**
- **Conclusion**: for any path π we have $\forall n.\, \pi(n) \vDash q$

not all true properties are inductive



k-induction

Base: Prove $Q(0) \wedge \cdots \wedge \mathrm{Q}(k)$

Induction: Prove $\mathrm{Q}(\mathrm{n}-\mathrm{k}) \wedge \cdots \wedge Q(n) \Rightarrow Q(n+1)$

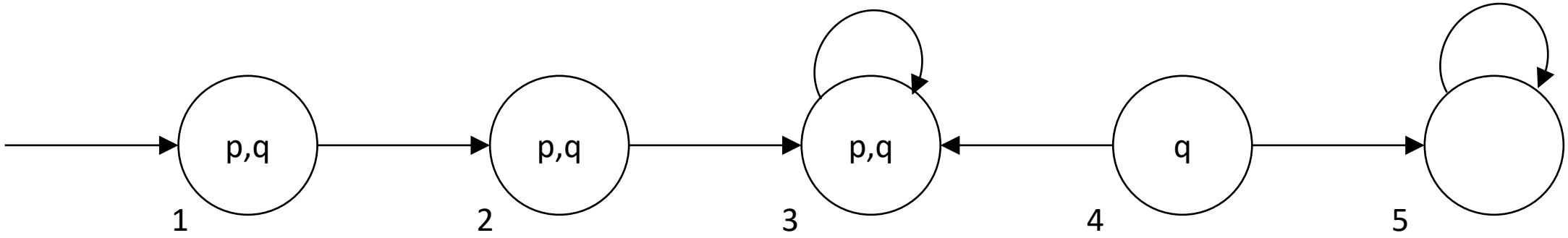
Conclusion: $\forall n.\, Q(n)$

In our setting:

Base. all paths from S_0 with k or fewer edges are labeled q

Induction. all paths of length k labeled with all qs are followed by a q

Conclusion. All paths from S_0 are labeled q

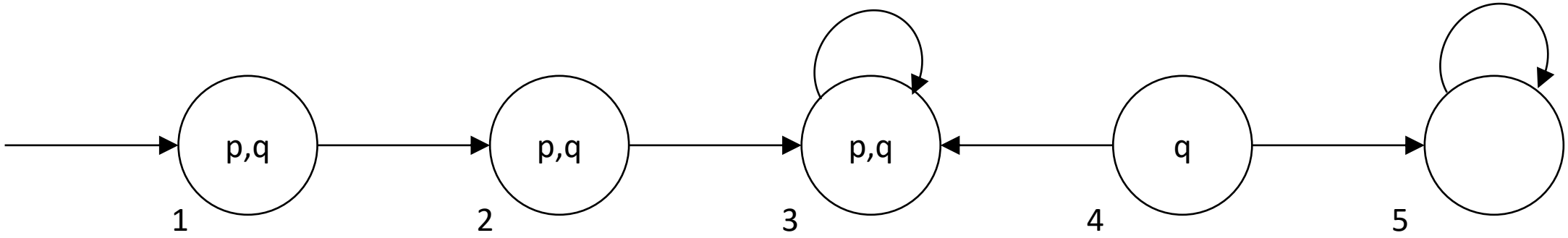


Prove AG q using 1-induction

Base: Consider all paths of length 1 from q_1: $q_1 \vDash q$ and $q_2 \vDash q$.

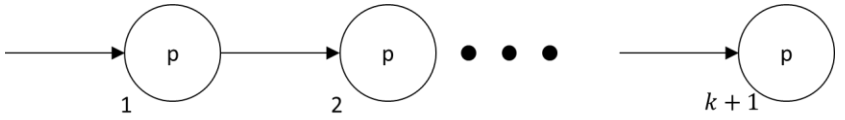
Induction: Do all successors of paths of length 1 labeled (q, q) fulfill q?

- (q_1, q_2)
- (q_2, q_3)
- (q_3, q_3)
- (q_4, q_3)



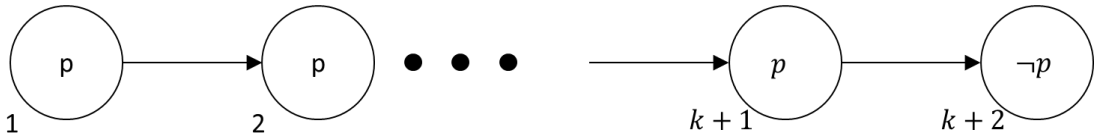
k-induction as Satisfiability

Base. There is a path of length *k* from S_0 not fully labeled *p*



$$S_0(s_1) \wedge \bigwedge_{i=1}^{k} R(s_i, s_{i+1}) \wedge \bigvee_{i=1}^{k+1} \neg p(s_i)$$

Induction. There is a path of length *k* labeled with all *p*s is followed by $\neg p$



$$\bigwedge_{i=1}^{k+1} R(s_i, s_{i+1}) \wedge \bigwedge_{i=1}^{k+1} p(s_i) \wedge \neg p(s_{k+2})$$

Formula satisfiable iff there is a counterexample

k-induction

while(k=0; ; k++){

build BMC formula ϕ_k

if ϕ SAT return “bug!”

build induction formula ψ_k

if ϕ UNSAT return “correct!”

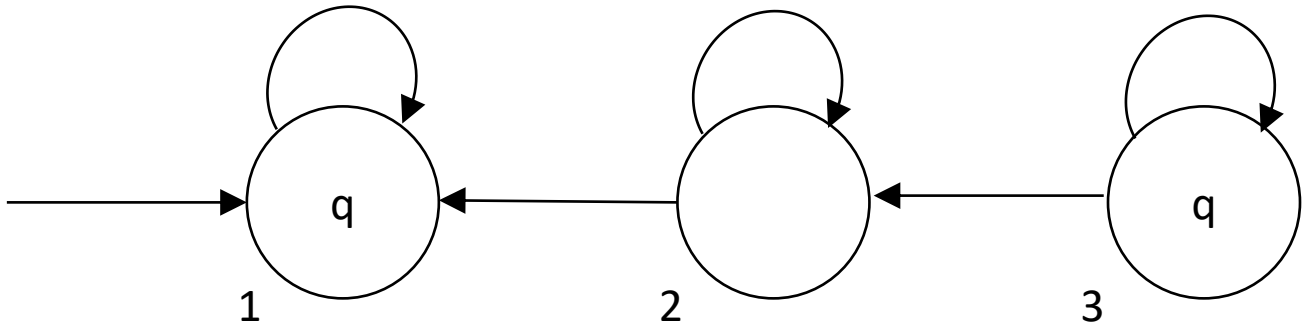
}

This Version of k-induction is not Complete

System satisfies $\mathbf{AG}\ q$, but induction step fails for any *k*

Base. all paths of length *k* from S_0 are labeled *q*

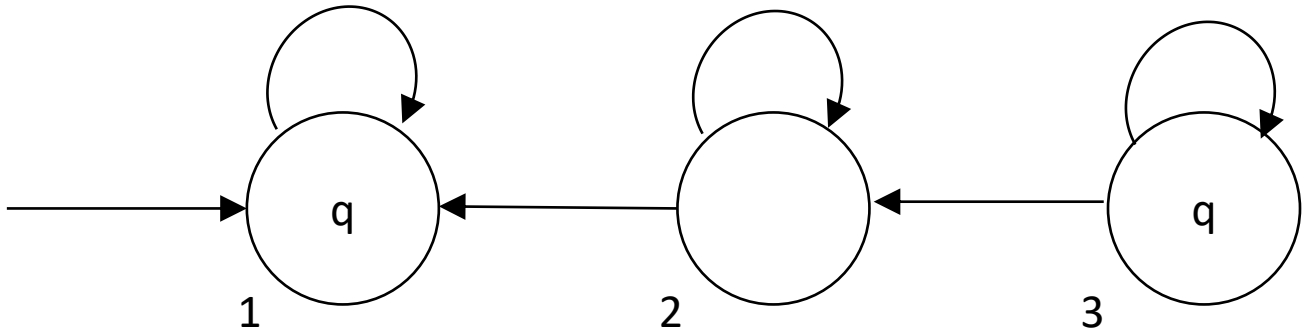
Induction. all paths of length *k* labeled with all *q*s are followed by a *q*. *FALSE*



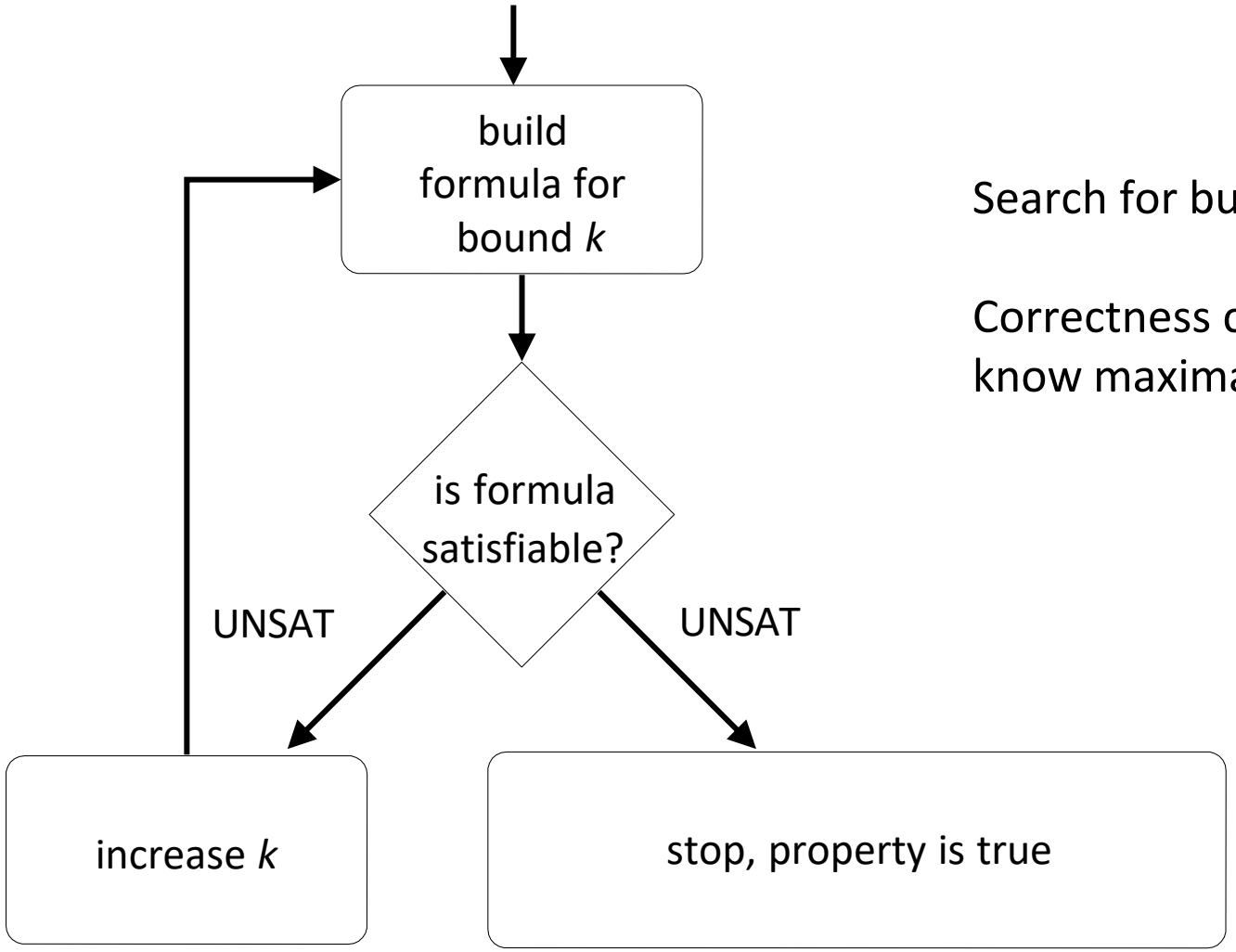
k-induction, the Final Version

Induction. all noncyclic paths of length k labeled with all ps are followed by a p

$$\bigwedge_{i=1}^{k+1} R(s_i, s_{i+1}) \wedge \bigwedge_{i=1}^{k+1} p(s_i) \wedge \neg p(s_{k+2}) \wedge \bigwedge_{i=1}^{k+1} \bigwedge_{j=i+1}^{k+1} s_i \neq s_j$$



k-induction



Search for bugs within k steps.

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

Problems with *k*-induction

Problem: Sometimes *k* is very large

In the following machine, you need $k = n + 1$ to prove $\mathbf{AG}\, p$.

Idea: Automatically find better inductive invariants.