## COL:750

## **Foundations of Automatic Verification**

Course Webpage

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

#### Instructor: Priyanka Golia



#### Model Checking Algorithm — so far

 $M, s \models F?$ 



#### $S' \quad s \, . \, t \, . \, S' \subseteq S$ , and $M, s \models F \forall s \in S'$ All states s of the model M that satisfy F

Note that not necessarily  $I \subseteq S'$ 



Output: UNSAT, if M unrolled upto k satisfies F A counterexample if M unrolled upto k don't satisfy F

#### $\rightarrow$ Yes, if $M, s \models F, \forall s \in I$ No, if $\exists s \in I s \cdot t \cdot M, s \not\models F$ , A path of the system M demonstrating that M can't satisfy F

Given: Transition system M, Temporal logic formula F, and a user-supplied time bound k



Output: UNSAT, if M unrolled upto k satisfies F A counterexample if if M unrolled upto k don't satisfy F

 $\rightarrow$  Yes, there is a counterexample  $\sigma \models M_k \land \neg F_k$  $\rightarrow$  No, if  $M_k \models F_K, \forall s \in I$ 

Given: Transition system M, Temporal logic formula F, and a user-supplied time bound k



General idea:

- Convert transition system to propositional encoding unroll for path 1. length k
- Convert temporal formula along the states to propositional encoding for 2. k length.
- 3. Using SAT Solvers look for counterexamples

Given two processes P and Q which share a resource R.

- If R is accessed by P, then property p is True. 1.
- If R is accessed by Q, then property q is True. 2.







 $M_k = (\neg p_0 \land \neg q_0) \land ((\neg p_0 \land \neg q_0 \land p_1 \land \neg q_1) \lor (\neg q_0 \land p_1 \land q_1) \lor (\neg q_0 \land p_1 \land q_1) \lor (\neg q_0 \land p_1 \land q_1) \lor (\neg q_1 \land q_1) \lor (\neg q_1) \lor (\neg q_1 \land q_1) \lor (\neg q_$ 

 $\neg F_k = p_1 \land q_1$ 

 $SAT\{M_k \land \neg F_k\} \longrightarrow UNSAT, M_{k=1} \models F_{k=1}$ 

bes 
$$\forall \Box \neg (p \land q)$$
 K =

י
$$p_0 \wedge \neg q_0 \wedge \neg p_1 \wedge q_1))$$



 $M_k = (\neg p_0 \land \neg q_0) \land ((\neg p_0 \land \neg q_0 \land p_1 \land \neg q_1) \lor (\neg q_0 \land p_1 \land (\neg q_1 \land q_1) \lor (\neg q_1 \land q_1) \lor (\neg q_0 \land p_1 \land q_1) \lor (\neg q_1 \land q_1) \lor (\neg q_1) \lor (\neg q_1 \land q_1) \lor (\neg q_1$  $M_k = (\neg p_0 \land \neg q_0) \land ((\neg p_0 \land \neg q_0 \land p_1 \land \neg q_1) \lor (\neg p_0 \land \neg q_0 \land \neg p_1 \land q_1))$ 

bes 
$$\forall \Box \neg (p \land q)$$
 K = 2

$$\neg p_0 \land \neg q_0 \land \neg p_1 \land q_1)) \qquad \mathsf{K} = \mathsf{1}$$

K = 2  $\wedge \left( \left( (p_1 \land \neg q_1 \land p_2 \land q_2) \lor (p_1 \land \neg q_1 \land \neg p_2 \land \neg q_2) \right) \lor \left( \neg p_1 \land q_1 \land \neg p_2 \land \neg q_2 \right) \right)$ 





$$\begin{split} M_k &= (\neg p_0 \land \neg q_0) \land ((\neg p_0 \land \neg q_0 \land p_1 \land \neg q_1) \lor (\neg q_1 \land (((p_1 \land \neg q_1 \land p_2 \land q_2) \lor (p_1 \land \neg q_1 \land q_1) \land q_2))) \end{split}$$

 $\neg F = \exists \Diamond (p \land q) \quad \neg F_k = p_2 \land q_2 \qquad \text{SAT}\{$ 

bes 
$$\forall \Box \neg (p \land q)$$
 K = 2

$$\begin{array}{l} \mathsf{K} = \mathbf{2} \\ \mathsf{K} = \mathbf{2} \\ \mathsf{K} = q_0 \land \neg q_0 \land \neg p_1 \land q_1 )) \\ \mathsf{K} = q_1 \land \neg p_2 \land \neg q_2 )) \lor (\neg p_1 \land q_1 \land \neg p_2 \land \neg q_2 ))) \end{array}$$





$$\begin{split} M_k &= (\neg p_0 \land \neg q_0) \land ((\neg p_0 \land \neg q_0 \land p_1 \land \neg q_1) \lor (\neg q_1 \land ((p_1 \land \neg q_1 \land p_2 \land q_2) \lor (p_1 \land \neg q_1 \land q_1))) \end{split}$$

 $\neg F_k = p_2 \land q_2 \qquad SAT\{M_k \land \neg F_k\}$ 

 $\sigma = \langle p_0 = 0, q_0 = 0, p_1 = 1, q_1 = 0, p_2 = 1, q_2 = 1 \rangle$ 

bes 
$$\forall \Box \neg (p \land q)$$
 K = 2

$$\begin{array}{l} (\mathbf{x} = \mathbf{z}) \\ (\mathbf{y}_{1} \wedge \mathbf{y}_{0} \wedge \mathbf{y}_{1} \wedge q_{1})) \\ (\mathbf{y}_{1} \wedge \mathbf{y}_{2} \wedge \mathbf{y}_{2})) \vee (\mathbf{y}_{1} \wedge q_{1} \wedge \mathbf{y}_{2} \wedge \mathbf{y}_{2}))) \end{array}$$



Two-bit counter



 $\neg F_k = (\neg p_0 \lor \neg q_0) \land (\neg p_1 \lor \neg q_1) \land (\neg p_2 \lor \neg q_2) \land (\neg p_3 \lor \neg q_3)$ 

 $M_k \wedge \neg F_k \qquad SAT\{M_k \wedge \neg F_k\}$ 

 $\sigma = \langle p_0 = 0, q_0 = 0, p_1 = 0, q_1 = 1, p_2 = 1, q_2 = 0, p_3 = 1, q_3 = 0 \rangle$ 

## $F = \forall \Diamond (p \land q) \quad \neg F = \exists \Box \neg p \lor \neg q$

 $M_{k} = (\neg p_{o} \land \neg q_{o}) \land (\neg p_{o} \land \neg q_{o} \land \neg p_{1} \land q_{1}) \land (\neg p_{1} \land q_{1} \land p_{2} \land \neg q_{2}) \land ((p_{2} \land \neg q_{2} \land p_{3} \land \neg q_{3}) \lor (p_{2} \land \neg q_{2} \land p_{3} \land q_{3}))$ 

What happens for K= 2?

$$M_k \nvDash F_k$$





Property - 
$$\forall \diamondsuit p$$
Every path in M includes a s $\exists \Box \neg p$ An infinite path in which  $d$ 

Lasso: A lasso is a finite path that consists of: A prefix: a finite sequence of transitions from the initial state. A loop: a back edge that loops from the last state back to some earlier state.

$$Lasso_k(s_o, \dots, s_k) := \bigvee$$

 $M_k \models (\exists \Box \neg p)_k := M_k \land (Las)$ 

state in which p is True.

all states satisfy  $\neg p$ . A loop is needed!

 $T(s_k, s_i)$ 

l=0

$$SO_i(S_o, \dots, S_k) \wedge \bigwedge_{i=o}^k \neg p(S_i)$$

## **Bounded Model Checking with SAT (BMC)** $Lasso_k(s_o, \dots, s_k) := \bigvee^k T(s_k, s_i) \quad M_k \models (\exists \Box \neg p)_k := M_k \land (Lasso_i(s_o, \dots, s_k) \land \bigwedge^{\kappa} \neg p(s_i)$



 $M_2 \models (\exists \Box \neg p \lor \neg q)_2 :=$ 

#### $M_k \wedge (T(s_2, s_0) \vee T(s_2, s_1) \vee T(s_2, s_2)) \wedge (\neg p_0 \vee \neg q_0) \wedge (\neg p_1 \vee \neg q_1) \wedge (\neg p_2 \vee \neg q_2))$





How big should be K?

For every model M and formula (LTL/CTL) F, there exists k, such that  $M \models_K F \to M \models F$ 

The minimal such k is the Completeness Threshold (CT).

How big should be K?

 $M \models_K F \to M \models F$ 

The minimal such k is the Completeness Threshold (CT).

Diameter of M

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

#### For every model M and formula (LTL/CTL) F, there exists k, such that

Diameter(M) = Max ShortestPathLength(s, s')  $\forall s, s' \in T$ 

Diameter of M

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

> Diameter(M) = Max ShortestPathLength(s, s')  $\forall s, s' \in T$

It measures how far apart any two states can be. states.

- It gives a worst case bound on how many steps are required to reach any states from another

Diameter of M What is the smallest k such that every state is reachable within k transitions?

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

Diameter(M) = Max ShortestPathLength(s, s') $\forall s, s' \in T$ 



Diameter is 2.

Diameter of M What is the smallest k such that every state is reachable within k transitions?

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



- Diameter(M) = Max ShortestPathLength(s, s')  $\forall s, s' \in T$

Diameter is 2.

Observe that Diameter is not a completeness threshold for arbitrary properties.

Minimum k required for finding a counterexample for  $\forall \Diamond p$ ?



$$p)_k := M_k \wedge (Lasso_i(s_o, \dots, s_k) \wedge \bigwedge_{i=o}^k \neg p(s_i)$$

Shortest counterexample requires K=5

If F is a liveness property (something that must eventually hold, or hold infinitely, the Diameter is not a completeness threshold

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

For Safety property, d is a completeness threshold.

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

How to check if  $K \ge d$ ?

State s is reachable in j steps:

$$R_{j}(s) = \exists s_{o}, \dots, s_{j} \ s \ t \ s_{j} \land I(s_{o}) \land \bigwedge_{i=o}^{j-1} T(s_{i}, s_{i+1})$$
  
K is greater than or equal to Diameter d if

$$\forall s : R_{k+1}(s) \to \exists j \le k \; R_j(s)$$
 For

Computationally hard problem — requires QBF calls.

Diameter(M) = Max ShortestPathLength(s, s')  $\forall s, s' \in T$ 

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

Recurrence Diameter: Least number of steps n such that all valid paths of length n have at least one cycle.

rd is the longest loop-free path in M.

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



- This means that after *rd* steps, either:
- All reachable states have been visited.
- Any further steps must repeat a previously visited state.

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  $K \ge rd$  Return True, Else: K = K+1 For safety property.



#### Most influential paper in the first 20 years of TACAS

#### Symbolic Model Checking without BDDs\*

Armin Biere<sup>1</sup>, Alessandro Cimatti<sup>2</sup>, Edmund Clarke<sup>1</sup>, Yunshan Zhu<sup>1</sup>

Computer Science Department, Carnegie Mellon Unive 1000 Forbes Avenue, Pittsburgh, PA 15213, U.S.A .Biere, Edmund. Clarke, Yunshan. Zhu }gcs.cmu.edu arive 18, 38055 Povo (TN). Its

Abstract. Symbolic Model Checking [3, 14] has proven to be a powerful tech nique for the verification of reactive systems. BDDs [2] have traditionally been entation of the system. In this paper we show ho dures, like Stålmarck's Method [16] or the Davis & Putre [7], can replace BDDs. This new tech amples much faster, and som onal satisfiability. We show that bounded LTL model chee can be done without a tableau co

#### Introducti

Model checking [4] is a powerful technique for verifying reactive systems. Able to find subtle errors in real commercial designs, it is gaining wide industrial acceptance. Com-pared to other formal verification techniques (e.g. theorem proving) model checking is

In model checking, the specification is expressed in tem is modeled as a finite state machine. For realistic designs, the number of states of the system can be very large and the explicit traversal of the state space becomes in asible. Symbolic model checking [3, 14], with boolean encoding of the finite sta machine, can handle more than 10<sup>2</sup> states, BDDs [2], a onally been used as the underly odel checkers based on BDDs are usually able to handle is of state variables. However, for larger systems the

April 8th 2014, Grenoble i J'd W. Chauchend J. Atrone Zick Kim Leinen

#### extensions to completeness

- diameter checking,
- k-induction,
- interpolation
  - SAT based model checking without unrolling: IC<sub>3</sub>

