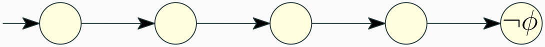
CSCI 740 - Programming Language Theory

Lecture 35
Bounded Model Checking
Instructor: Hossein Hojjat

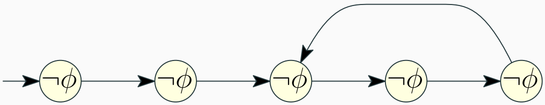
November 29, 2017

Observation

- LTL model checking requires checking **all** paths
- On the other hand: a counterexample to LTL formula ϕ corresponds to the question if there **exists** a witness for $\neg\phi$
- A counterexample for $G\ \phi$ is a finite prefix of a path in which $F\neg\phi$ holds



- A counterexample for $F\ \phi$ is a finite prefix of a path that is a lasso in which $G\neg\phi$ holds



- Finite paths may say something about infinite behaviors

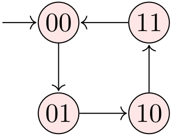
Idea

- Bounded Model Checking (BMC) performs only on the basis of finite **bounded** prefixes of paths of the system
- Unroll the transition relation up to certain fixed bound k and search for violations of the property within that bound
- Transform this search to a Boolean satisfiability problem and solve it using a SAT solver
- Mostly incomplete in practice:

  validity of a formula can often not be proven

Motivating Example: Two-bit Counter

Does the safety property $G\neg(x \wedge y)$ hold in the initial state?

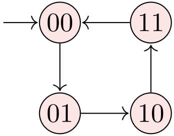


Boolean Variables: x , y

- Represent initial states and the transition relation as Boolean formulas

Motivating Example: Two-bit Counter

Does the safety property $G\neg(x \wedge y)$ hold in the initial state?



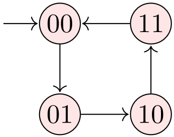
Boolean Variables: x , y

Initial State: $I(x, y) = \neg x \wedge \neg y$

- Represent initial states and the transition relation as Boolean formulas

Motivating Example: Two-bit Counter

Does the safety property $G\neg(x \wedge y)$ hold in the initial state?



Boolean Variables: x , y

Initial State: $I(x, y) = \neg x \wedge \neg y$

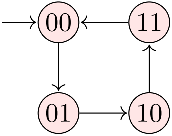
Transition Relation:

$$R(x, y, x', y') = \Big(x' = (x \neq y) \wedge y' = \neg y\Big)$$

- Represent initial states and the transition relation as Boolean formulas

Motivating Example: Two-bit Counter

Does the safety property $G\neg(x \wedge y)$ hold in the initial state?



Boolean Variables: x , y

Initial State: $I(x, y) = \neg x \wedge \neg y$

Transition Relation:

$R(x, y, x', y') = \Big(x' = (x \neq y) \wedge y' = \neg y \Big)$

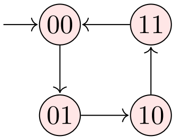
- Represent initial states and the transition relation as Boolean formulas
- Unroll the transition relation up to a bound k starting from the initial states

$$(\neg x_0 \wedge \neg y_0) \quad \wedge \quad \begin{pmatrix} x_1 = (x_0 \neq y_0) \wedge y_1 = \neg y_0 \\ \wedge \\ x_2 = (x_1 \neq y_1) \wedge y_2 = \neg y_1 \\ \wedge \\ x_3 = (x_2 \neq y_2) \wedge y_3 = \neg y_2 \end{pmatrix} \wedge \begin{pmatrix} (x_0 \wedge y_0) \\ \vee \\ (x_1 \wedge y_1) \\ \vee \\ (x_2 \wedge y_2) \\ \vee \\ (x_2 \wedge y_2) \end{pmatrix}$$

$\boxed{\text{UNSAT for } k = 0}$

Motivating Example: Two-bit Counter

Does the safety property $G\neg(x \wedge y)$ hold in the initial state?



Boolean Variables: x , y

Initial State: $I(x, y) = \neg x \wedge \neg y$

Transition Relation:

$$R(x, y, x', y') = \Big(x' = (x \neq y) \wedge y' = \neg y\Big)$$

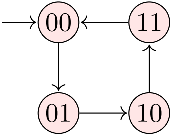
- Represent initial states and the transition relation as Boolean formulas
- Unroll the transition relation up to a bound k starting from the initial states

$$(\neg x_0 \wedge \neg y_0) \quad \wedge \quad \left(\begin{array}{c} x_1 = (x_0 \neq y_0) \wedge y_1 = \neg y_0 \\ \wedge \\ x_2 = (x_1 \neq y_1) \wedge y_2 = \neg y_1 \\ \wedge \\ x_3 = (x_2 \neq y_2) \wedge y_3 = \neg y_2 \end{array}\right) \wedge \left(\begin{array}{c} (x_0 \wedge y_0) \\ \vee \\ (x_1 \wedge y_1) \\ \vee \\ (x_2 \wedge y_2) \\ \vee \\ (x_2 \wedge y_2) \end{array}\right)$$

$\boxed{\text{UNSAT for } k = 1}$

Motivating Example: Two-bit Counter

Does the safety property $G\neg(x \wedge y)$ hold in the initial state?



Boolean Variables: x , y

Initial State: $I(x, y) = \neg x \wedge \neg y$

Transition Relation:

$R(x, y, x', y') = \Big(x' = (x \neq y) \wedge y' = \neg y\Big)$

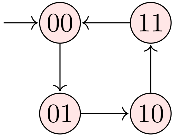
- Represent initial states and the transition relation as Boolean formulas
- Unroll the transition relation up to a bound k starting from the initial states

$$(\neg x_0 \wedge \neg y_0) \quad \wedge \quad \left(\begin{array}{c} x_1 = (x_0 \neq y_0) \wedge y_1 = \neg y_0 \\ \wedge \\ x_2 = (x_1 \neq y_1) \wedge y_2 = \neg y_1 \\ \wedge \\ x_3 = (x_2 \neq y_2) \wedge y_3 = \neg y_2 \end{array} \right) \wedge \left(\begin{array}{c} (x_0 \wedge y_0) \\ \vee \\ (x_1 \wedge y_1) \\ \vee \\ (x_2 \wedge y_2) \\ \vee \\ (x_2 \wedge y_2) \end{array} \right)$$

UNSAT for $k = 2$

Motivating Example: Two-bit Counter

Does the safety property $G\neg(x \wedge y)$ hold in the initial state?



Boolean Variables: x , y
Initial State: $I(x,y) = \neg x \wedge \neg y$
Transition Relation:
$R(x,y,x',y') = \Big(x' = (x \neq y) \wedge y' = \neg y\Big)$

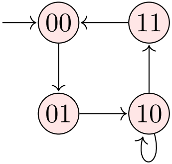
- Represent initial states and the transition relation as Boolean formulas
- Unroll the transition relation up to a bound k starting from the initial states

$$(\neg x_0 \wedge \neg y_0) \quad \wedge \quad \begin{pmatrix} x_1 = (x_0 \neq y_0) \wedge y_1 = \neg y_0 \\ \wedge \\ x_2 = (x_1 \neq y_1) \wedge y_2 = \neg y_1 \\ \wedge \\ x_3 = (x_2 \neq y_2) \wedge y_3 = \neg y_2 \end{pmatrix} \wedge \begin{pmatrix} (x_0 \wedge y_0) \\ \vee \\ (x_1 \wedge y_1) \\ \vee \\ (x_2 \wedge y_2) \\ \vee \\ (x_2 \wedge y_2) \end{pmatrix}$$

$\boxed{\text{SAT for } k = 3}$

Motivating Example: Two-bit Counter

Does the safety property $F(x \wedge y)$ hold in the initial state?



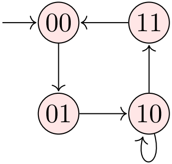
Boolean Variables: x , y

Initial State: $I(x, y) = \neg x \wedge \neg y$

$$R(x, y, x', y') = \Big(x' = (x \neq y) \wedge y' = \neg y\Big) \vee$$
$$(x' = x \wedge y' = y \wedge x \wedge \neg y)$$

Motivating Example: Two-bit Counter

Does the safety property $F(x \wedge y)$ hold in the initial state?



Boolean Variables: x , y

Initial State: $I(x,y) = \neg x \wedge \neg y$

$$R(x,y,x',y') = \Big(x' = (x \neq y) \wedge y' = \neg y\Big) \vee$$
$$(x' = x \wedge y' = y \wedge x \wedge \neg y)$$

$$I(x_0,y_0) \wedge R(x_0,y_0,x_1,y_1) \wedge R(x_1,y_1,x_2,y_2) \wedge \bigvee_{i=0}^{2} \neg(x_i \wedge y_i) \wedge \textit{loop}$$

where

$$\textit{loop} = R(x_2,y_2,x_3,y_3) \wedge$$
$$\big((x_3 = x_0 \wedge y_3 = y_0) \vee (x_3 = x_1 \wedge y_3 = y_1) \vee (x_3 = x_2 \wedge y_3 = y_2)\big)$$

SAT: satisfying assignment gives counterexample to the liveness property

BMC: Safety Property

- Given: transition system M, temporal logic formula ϕ and user-supplied bound k
- Construct propositional formula that is satisfiable iff ϕ is valid along a path of length k
- Initialized Paths of length k:

$$[\![M]\!]_k = I(s_0) \wedge \bigwedge_{i=0}^{k-1} R(s_i, s_{i+1})$$

- $G\ \phi$ means ϕ must hold in every state along any path of length k

$$[\![M]\!]_k \wedge \bigvee_{i=0}^{k} \neg\phi_i$$

- If satisfiable, satisfying assignment gives:
  - Witness for $F\neg\phi$
  - Counterexample to the safety property $G\phi$

How big should k be?

- For every model M and LTL property ϕ there exists k s.t.

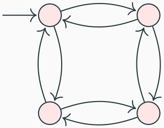
$$M \models_k \phi \rightarrow M \models \phi$$

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

How big should k be?

- Diameter d = longest shortest path from an initial state to any other reachable state
- Recurrence Diameter rd = longest loop-free path
- $rd \geq d$

$d = 2$
$rd = 3$

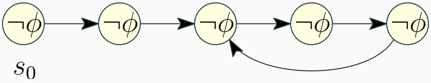


How big should k be?

- Theorem: for $G\phi$ properties CT = d

How big should k be?

- Theorem: for $F\phi$ properties CT = rd



Open Problem: The value of CT for general Linear Temporal Logic properties is unknown

See e.g.
"Linear Completeness Thresholds for Bounded Model Checking" (CAV'11)