# CDA 5416: CAV

## Symbolic CTL Model Checking

Hao Zheng

Department of Computer Science and Engineering
University of South Florida
Tampa, FL 33620
Email: zheng@cse.usf.edu
Phone: (813)974-4757
Fax: (813)974-5456

# Explicit Algorithms

- Transition systems are stored as graphs using hash tables.
- States are labeled with appropriate AP/subformlas.
- Complexity of model checking algorithms is linear in the structure sizes.
  - Structure size can be exponential!
- Problems
  - Demand of large amount of memory.
  - Low performance.

# Symbolic CTL Model Checking

- Idea: reformulate model-checking in a symbolic way.
- Concept: represent sets of states and transitions symbolically.
- Approach: binary encoding of states + switching functions for sets.
- Compact representation of switching functions is possible using binary decision diagrams (BDDs).
- Alternative representation is the conjunctive normal form which is the basis for SAT-based model checking.

# Contents

# Switching Functions

- Let $Var = \{z_1, \ldots, z_m\}$ be a finite set of Boolean variables, $m \geq 0$.
- An evaluation is a function $\eta : Var \to \{0, 1\}$.
  - Let $Eval(z_1, \ldots, z_m)$ denote the set of evaluations for $z_1, \ldots, z_m$.
  - Shorthand $[z_1 = \mathfrak{b}_1, \ldots, z_m = \mathfrak{b}_m]$ for $\eta(z_1) = \mathfrak{b}_1, \ldots, \eta(z_m) = \mathfrak{b}_m$.
- $f : Eval(Var) \to \{0, 1\}$ is a *switching function* for $Var = \{z_1, \ldots, z_m\}$.
  - Can be defined by Boolean expressions, i.e. $(z_1 \vee \neg z_2) \wedge z_3$

# Switching Functions: Definitions

- $f_1 \ \wedge \ f_2 \ = \ \min\{ f_1, f_2 \}$
- $f_1 \ \vee \ f_2 \ = \ \max\{ f_1, f_2 \}$
- $f|_{z_i=b_i}(z_1, \ldots, z_i, \ldots, z_m) = f(z_1, \ldots, b_i, \ldots, z_m)$ *(cofactor)*.

  e.g. $((a \wedge b) \vee c)|_{b=1} \ = \ a \vee c$
- $f|_{z_i=b_i,\ldots,z_k=b_k} = ((f|_{z_i=b_i}) \ldots)|_{z_k=b_k}$ *(iterated cofactor)*.
- If $f|_{z_i=0} \neq f|_{z_i=1}$ then $z_i$ is an *essential variable*.

# Switching Functions: Definitions (1)

- $f = (\neg z \;\wedge\; f|_{z=0}) \;\vee\; (z \;\wedge\; f|_{z=1})$ (*Shannon expansion*).
- $\exists z.\, f \;=\; f|_{z=0} \;\vee\; f|_{z=1}$ (*existential quantification*).

  e.g. $\exists \boldsymbol{b}.((\boldsymbol{a} \wedge \boldsymbol{b}) \vee \boldsymbol{c}) \;=\; (\boldsymbol{c}) \vee (\boldsymbol{a} \vee \boldsymbol{c}) \;=\; \boldsymbol{a} \vee \boldsymbol{c}$

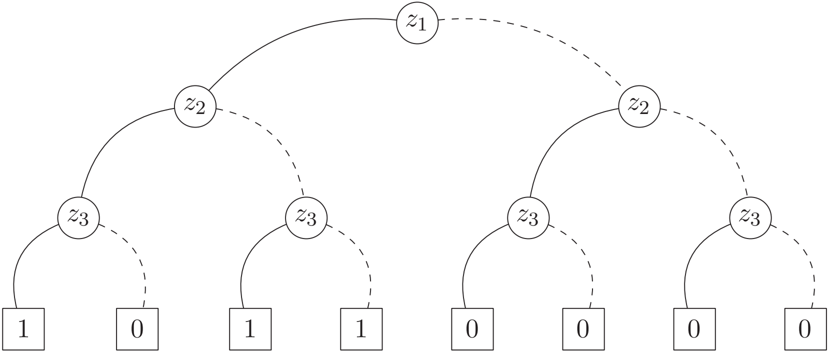
- $\forall z.\, f \;=\; f|_{z=0} \;\wedge\; f|_{z=1}$ (*universal quantification*).

  e.g. $\forall \boldsymbol{b}.((\boldsymbol{a} \wedge \boldsymbol{b}) \vee \boldsymbol{c}) \;=\; (\boldsymbol{c}) \wedge (\boldsymbol{a} \vee \boldsymbol{c}) \;=\; \boldsymbol{c}$

- $f\{z \leftarrow y\}(s) = f(s\{y \leftarrow z\})$ (*rename operator*).

# Switching Functions – Shannon Expansion

$$f = (\neg z_1 \wedge f|_{z_1=0}) \vee (z_1 \wedge f|_{z_1=1})$$



# Contents

# Symbolic Representation of TS

- Let $TS = (S, \to, I, AP, L)$ be a "large" finite transition system.
  Note: the set of actions is irrelevant and has been omitted, i.e., $\to \subseteq S \times S$.

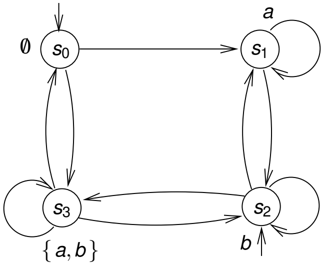
- For $n \geq \lceil \log |S| \rceil$, let injective function

$$enc : S \to \{ 0, 1 \}^n$$

  be the encoding of the states by bit vectors of length $n$.

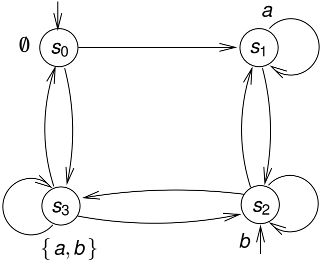
- Identify:
  - Each states $s \in S$ has an unique $enc(s) \in \{0, 1\}^n$.
  - $B \subseteq S$ by its characteristic function $\chi_B : \{ 0, 1 \}^n \to \{ 0, 1 \}$, that is $\chi_B(enc(s)) = 1$ if and only if $s \in B$.
  - $\to \subseteq S \times S$ by the Boolean function $\Delta : \{ 0, 1 \}^{2n} \to \{ 0, 1 \}$, such that $\Delta\,(enc(s), enc(s')) = 1$ if and only if $s \to s'$.

# Symbolic Representation of TS: Example




- Four states: two Boolean variables needed for encoding, i.e. $x_1, x_2$.

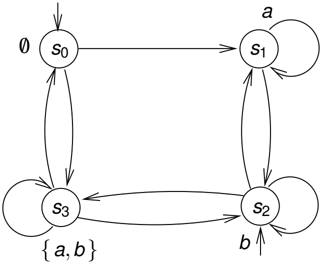
# Symbolic Representation of TS: Example




- State encoding on variables $x_1, x_2$:

$$f_S = 1$$

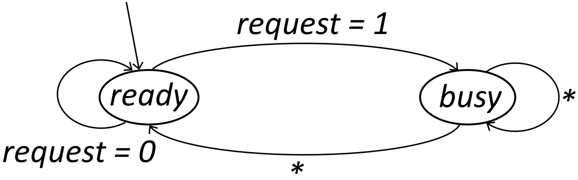
# Symbolic Representation of TS: Example




- Switching function: $\Delta(\underbrace{x_1, x_2}_{s}, \underbrace{x_1', x_2'}_{s'}) = 1$ if and only if $s \rightarrow s'$

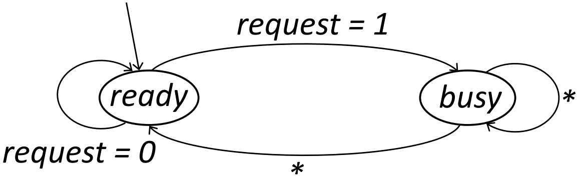
$$\begin{array}{rcl} \Delta(x_1, x_2, x_1', x_2') = & & (\neg x_1 \ \wedge \ \neg x_2 \ \wedge \ \neg x_1' \ \wedge \ x_2') \\ & \vee & (\neg x_1 \ \wedge \ \neg x_2 \ \wedge \ x_1' \ \wedge \ x_2') \\ & \vee & (\neg x_1 \ \wedge \ x_2 \ \wedge \ x_1' \ \wedge \ \neg x_2') \\ & \vee & \ldots \\ & \vee & (x_1 \ \wedge \ x_2 \ \wedge \ x_1' \ \wedge \ x_2') \end{array}$$

# Another Encoding Example



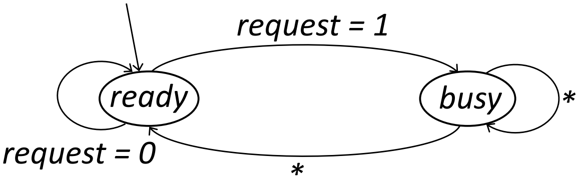

- Boolean variables, $x_1, x_2$.
  - $x_1 \leftrightarrow (\texttt{request} = 1)$, $\neg x_1 \leftrightarrow (\texttt{request} = 0)$,
    $x_2 \leftrightarrow (\texttt{state} = \mathit{ready})$, $\neg x_2 \leftrightarrow (\texttt{state} = \mathit{busy})$

# Another Encoding Example




- Initial state: $state = ready \longrightarrow x_2$

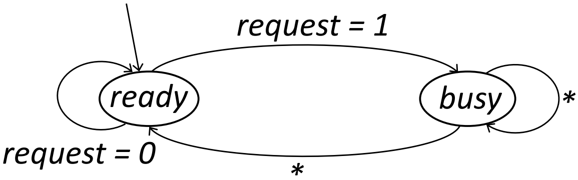
# Another Encoding Example




- Transition relation:

$$\begin{aligned}\Delta(\vec{x}, \vec{x}') \;=\; & (state = ready \wedge request = 1 \wedge state' = busy) \vee \\ & \big(\neg(state = ready \wedge request = 1) \wedge \\ & \quad ((state' = ready) \vee (state' = busy)\big)\end{aligned}$$

# Another Encoding Example




- Transition relation:

$$\begin{aligned}\Delta(\vec{x}, \vec{x}') &= (x_2 \wedge x_1 \wedge \neg x_2') \;\vee\; (\neg(x_2 \wedge x_1) \wedge (x_2' \vee \neg x_2')) \\ &= (x_2 \wedge x_1 \wedge \neg x_2') \;\vee\; (\neg(x_2 \wedge x_1)) \\ &= \neg x_2' \vee \neg(x_2 \wedge x_1)\end{aligned}$$

# Contents

# Computation of *Sat* - Review

**switch**($\Phi$):

$\text{EX}\,\Psi$ : **return** $\{ s \in S \mid Post(s) \cap Sat(\Psi) \neq \emptyset \}$;

$\exists(\Phi_1 \,\text{U}\, \Phi_2)$ : $T := Sat(\Phi_2)$; *compute the smallest fixed point*

**while** $\{ s \in Sat(\Phi_1) \setminus T \mid Post(s) \cap T \neq \emptyset \} \neq \emptyset$ **do**

  **let** $s \in \{ s \in Sat(\Phi_1) \setminus T \mid Post(s) \cap T \neq \emptyset \}$;

  $T := T \cup \{ s \}$;

**od**;

**return** $T$;

$\text{EG}\,\Phi$ : $T := Sat(\Phi)$; *compute the greatest fixed point*

**while** $\{ s \in T \mid Post(s) \cap T = \emptyset \} \neq \emptyset$ **do**

  **let** $s \in \{ s \in T \mid Post(s) \cap T = \emptyset \}$;

  $T := T \setminus \{ s \}$;

**od**;

**return** $T$;

**end switch**

# Symbolic Model Checking

- Preimage of state set $B$: $\quad Pre(B) = Sat(\mathsf{EX}\, B)$.

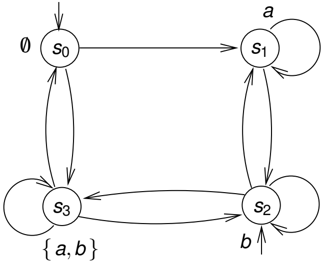
$$Pre(B) = \{s \in S \mid Post(s) \cap B \neq \emptyset\}$$

- Take a symbolic representation of a transition system ($\Delta$ and $\chi_B$).
- $Pre(B)$ can be symbolically computed as

$$\chi_{\mathsf{EX}\, B}(\overline{x}) = \exists \overline{x}'. ( \underbrace{\Delta(\overline{x}, \overline{x}')}_{s' \in Post(s)} \wedge \underbrace{\chi_B(\overline{x}')}_{s' \in B} ).$$

- $\chi_B(\overline{x}')$ is $\chi_B$ after renaming the variables $x_i$ to their primed copies $x_i'$.

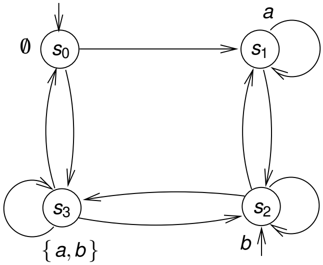
# Preimage Computatioin: Example




$$
\begin{array}{rcl}
\Delta(x_1, x_2, x_1', x_2') = & & (\neg x_1 \ \wedge \ \neg x_2 \ \wedge \ \neg x_1' \ \wedge \ x_2') \\
& \vee & \ldots \\
& \vee & (\neg x_1 \ \wedge \ x_2 \ \wedge \ x_1' \ \wedge \ \neg x_2') \\
& \vee & (x_1 \ \wedge \ \neg x_2 \ \wedge \ x_1' \ \wedge \ \neg x_2') \\
& \vee & (x_1 \ \wedge \ x_2 \ \wedge \ x_1' \ \wedge \ \neg x_2')
\end{array}
$$

Compute Preimage of $s_2$ $(x_1 \wedge \neg x_2)$

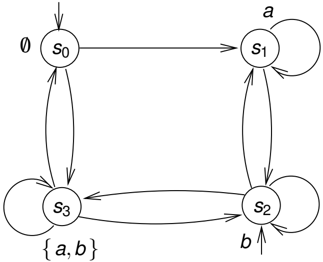
# Preimage Computatioin: Example




$$\exists x_1', x_2',\ \Delta(x_1, x_2, x_1', x_2') \wedge x_1' \wedge \neg x_2' =$$

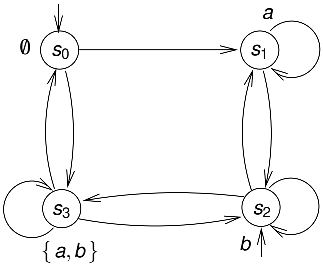
$$\exists x_1', x_2',\ \left( \begin{array}{cl} & (\neg x_1 \wedge \neg x_2 \wedge \neg x_1' \wedge x_2') \\ \vee & \ldots \\ \vee & (\neg x_1 \wedge x_2 \wedge x_1' \wedge \neg x_2') \\ \vee & (x_1 \wedge \neg x_2 \wedge x_1' \wedge \neg x_2') \\ \vee & (x_1 \wedge x_2 \wedge x_1' \wedge \neg x_2') \end{array} \right) \wedge (x_1' \wedge \neg x_2')$$

# Preimage Computatioin: Example




$$\exists x_1', x_2', \quad \left( \begin{array}{ll} \vee & (\neg x_1 \wedge x_2 \wedge x_1' \wedge \neg x_2') \\ \vee & (x_1 \wedge \neg x_2 \wedge x_1' \wedge \neg x_2') \\ \vee & (x_1 \wedge x_2 \wedge x_1' \wedge \neg x_2') \end{array} \right)$$

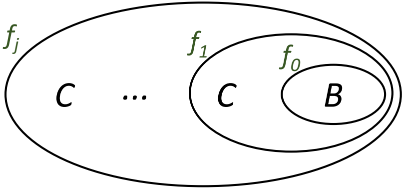
# Preimage Computatioin: Example




$$(\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg x_2) \vee (x_1 \wedge x_2)$$
$$= x_1 \vee x_2$$

# Symbolic Computation of $Sat(\exists(C \cup B))$

$f_0(\overline{x}) := \chi_B(\overline{x});$
$j := 0;$
**repeat**
  $f_{j+1}(\overline{x}) := f_j(\overline{x}) \vee \left( \chi_C(\overline{x}) \wedge \exists \overline{x}'.\,(\,\Delta(\overline{x},\overline{x}') \wedge f_j(\overline{x}')\,)\right);$
  $j := j + 1$
**until** $f_j(\overline{x}) = f_{j-1}(\overline{x});$
**return** $f_j(\overline{x})$.



# Symbolic Computation of $Sat(\mathsf{EG}\, B)$

Compute the largest set $T \subseteq B$ with $Post(t) \cap T \neq \emptyset$ for all $t \in T$

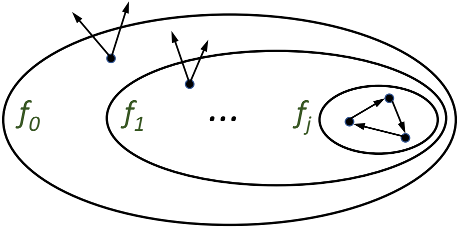
Take $T_0 = B$, repeat

$$T_j \;=\; T_{j-1} \cap \{ s \in S \mid \exists s' \in S.\, s' \in Post(s) \,\wedge\, s' \in T_{j-1} \,\}$$

until $T_j = T_{j-1}$

# Symbolic Computation of $Sat(\text{EG}\, B)$

$f_0(\overline{x}) := \chi_B(\overline{x})$;
$j := 0$;
**repeat**
  $f_{j+1}(\overline{x}) := f_j(\overline{x}) \wedge \exists \overline{x}'.\,(\,\Delta(\overline{x}, \overline{x}') \wedge f_j(\overline{x}')\,)$;
  $j := j + 1$
**until** $f_j(\overline{x}) = f_{j-1}(\overline{x})$;
**return** $f_j(\overline{x})$.



# Symbolic Composition

- How to compose $TS_i = (\Delta_i(\vec{x_i}, \vec{x_i}'), \chi_{I_i}(\vec{x_i})), \quad 0 \leq i \leq n$?
- Synchronous systems

$$\chi_I = \bigwedge_{0 \leq i \leq n} \chi_{I_i}(\vec{x_i}) \tag{1}$$

$$\Delta = \bigwedge_{0 \leq i \leq n} \Delta_i(\vec{x_i}, \vec{x_i}') \tag{2}$$

- Asynchronous systems

$$\chi_I = \bigwedge_{0 \leq i \leq n} \chi_{I_i}(\vec{x_i}) \tag{3}$$

$$\Delta = \bigvee_{0 \leq i \leq n} \left( \Delta_i(\vec{x_i}, \vec{x_i}') \bigwedge_{0 \leq j \leq n, j \neq i} \vec{x_j} = \vec{x_j}' \right) \tag{4}$$

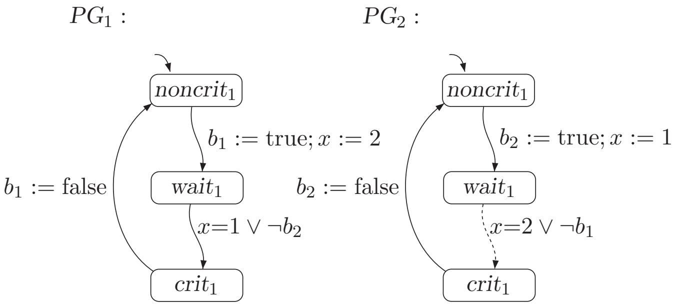
# Synchronous Counter

- Consider a 3-bit synchronous counter $(x_0, x_1, x_2)$
  - $\chi_{I_0} = \neg x_0,\ \chi_{I_1} = \neg x_1,\ \chi_{I_2} = \neg x_2.$
  - $\Delta_0 = x_0' \Leftrightarrow \neg x_0$
  - $\Delta_1 = x_1' \Leftrightarrow x_0 \oplus x_1$
  - $\Delta_2 = x_2' \Leftrightarrow \big(x_2 \wedge (x_0' = \neg x_0)\big) \vee \big(x_1 \wedge (x_2 \oplus x_0)\big)$
- The system

$$\chi_I = \bigwedge_{0 \leq i \leq 2} \chi_{I_i}(\vec{x_i}) = \neg x_0 \wedge \neg x_1 \wedge \neg x_2 \tag{5}$$

$$\Delta = \bigwedge_{0 \leq i \leq 2} \Delta_i(\vec{x_i}, \vec{x_i}') \tag{6}$$

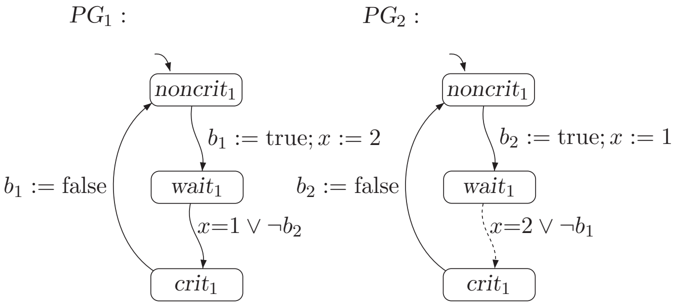
# Peterson's Mutual Exclusion Algorithm




- Encode program locations and propositions

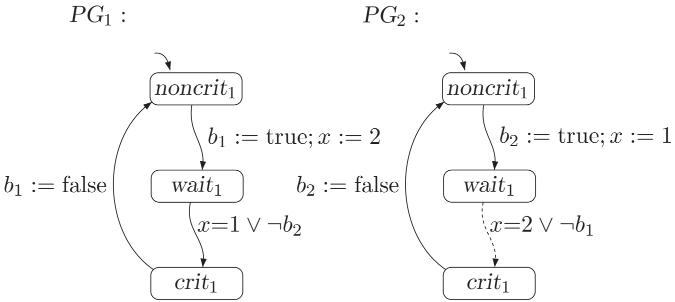
$$
\begin{aligned}
noncrit_1 &: \neg v_1 \wedge \neg v_0 \\
wait_1 &: \neg v_1 \wedge v_0 \\
crit_1 &: v_1 \wedge \neg v_0 \\
x = 1 &: \neg w_1 \wedge w_0 \\
x = 2 &: w_1 \wedge \neg w_0
\end{aligned}
\qquad
\begin{aligned}
noncrit_2 &: \neg u_1 \wedge \neg u_0 \\
wait_2 &: \neg u_1 \wedge u_0 \\
crit_2 &: u_1 \wedge \neg u_0 \\
x = 0 &: \neg w_1 \wedge \neg w_0
\end{aligned}
$$

# Peterson's Mutual Exclusion Algorithm




- Initial state:
  - Global variable: $\neg w_1 \wedge \neg w_0 \;\; (x = 0)$
  - Local variables of $PG_1$: $\neg v_1 \wedge \neg v_0 \wedge \neg b_1$
  - Local variables of $PG_2$: $\neg u_1 \wedge \neg u_0 \wedge \neg b_2$

# Peterson's Mutual Exclusion Algorithm




- Transition relation of $PG_1$:
  - $noncrit_1 \hookrightarrow wait_1$: $\neg v_1 \wedge \neg v_0 \wedge \neg {v_1}' \wedge {v_0}' \wedge {b_1}' \wedge w_1' \wedge \neg {w_0}'$
  - $wait_1 \hookrightarrow crit_1$:
  - $crit_1 \hookrightarrow wait_1$:
  - $\Delta_{PG_1} = \Delta_{noncrit_1 \hookrightarrow wait_1} \vee \Delta_{wait_1 \hookrightarrow crit_1} \vee \Delta_{crit_1 \hookrightarrow wait_1}$