Symbolic CTL Model Checking

Lecture #10 of Advanced Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

E-mail: katoen@cs.rwth-aachen.de

May 27, 2009

Existential normal form (ENF)

The set of CTL formulas in *existential normal form* (ENF) is given by:

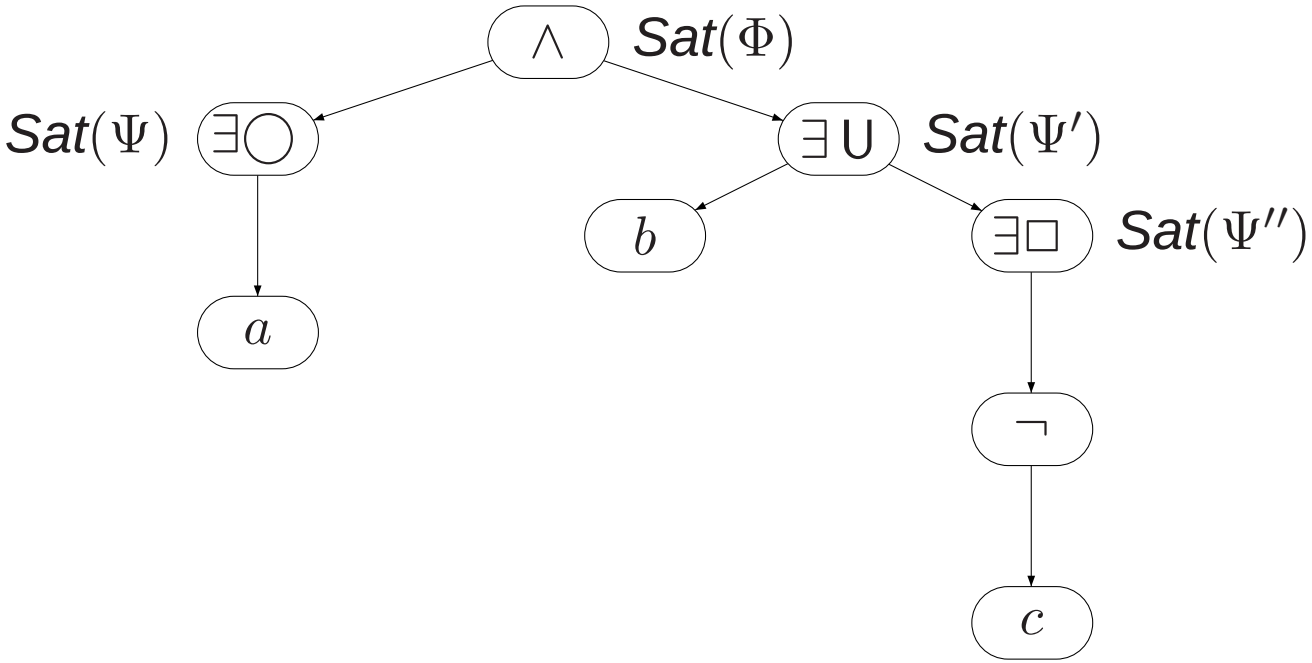
$$\Phi \quad ::= \quad \text{true} \;\Big|\; a \;\Big|\; \Phi_1 \wedge \Phi_2 \;\Big|\; \neg\Phi \;\Big|\; \exists\bigcirc \Phi \;\Big|\; \exists(\Phi_1 \,\mathsf{U}\, \Phi_2) \;\Big|\; \exists\Box\, \Phi$$

For each CTL formula, there exists an equivalent CTL formula in ENF

CTL model checking

- Convert the formula Φ' into an equivalent Φ in ENF
- How to check whether state *TS* satisfies Φ?
  - compute *recursively* the set *Sat*(Φ) of states that satisfy Φ
  - check whether all initial states belong to *Sat*(Φ)
- Recursive bottom-up computation:
  - consider the parse-tree of Φ
  - start to compute *Sat*(a), for all leafs in the tree
  - then go one level up in the tree and check the formula of these nodes
  - then go one level up and check the formula of these nodes
  - and so on..... until the root of the tree (i.e., Φ) is checked

Example



$$\Phi = \underbrace{\exists \bigcirc a}_{\Psi} \wedge \underbrace{\exists (b \,\mathsf{U}\, \underbrace{\exists \Box \neg c}_{\Psi''})}_{\Psi'} \quad .$$

Characterization of *Sat* (1)

For all CTL formulas Φ, Ψ over *AP* it holds:

$$
\begin{aligned}
\mathit{Sat}(\text{true}) &= S \\
\mathit{Sat}(a) &= \{\, s \in S \mid a \in L(s) \,\}, \text{ for any } a \in \mathit{AP} \\
\mathit{Sat}(\Phi \wedge \Psi) &= \mathit{Sat}(\Phi) \cap \mathit{Sat}(\Psi) \\
\mathit{Sat}(\neg \Phi) &= S \setminus \mathit{Sat}(\Phi) \\
\mathit{Sat}(\exists \bigcirc \Phi) &= \{\, s \in S \mid \mathit{Post}(s) \cap \mathit{Sat}(\Phi) \neq \varnothing \,\}
\end{aligned}
$$

where $\mathit{TS} = (S, \mathit{Act}, \rightarrow, I, \mathit{AP}, L)$ is a transition system without terminal states

Characterization of *Sat* (2)

For all CTL formulas Φ, Ψ over *AP* it holds:

- *Sat*$(\exists(\Phi \,\mathsf{U}\, \Psi))$ is the smallest subset T of S, such that:

  (1) *Sat*$(\Psi) \subseteq T$ and

  (2) $s \in Sat(\Phi)$ and *Post*$(s) \cap T \neq \varnothing$ implies $s \in T$

- *Sat*$(\exists\Box\, \Phi)$ is the largest subset T of S, such that:

  (3) $T \subseteq$ *Sat*(Φ) and

  (4) $s \in T$ implies *Post*$(s) \cap T \neq \varnothing$

where $TS = (S, Act, \rightarrow, I, AP, L)$ is a transition system without terminal states

Computation of *Sat*

switch(Φ):

a	:	**return** $\{ s \in S \mid a \in L(s) \}$;
$\ldots$	:	$\ldots\ldots$
$\exists\bigcirc \Psi$	:	**return** $\{ s \in S \mid \mathit{Post}(s) \cap \mathit{Sat}(\Psi) \neq \varnothing \}$;
$\exists(\Phi_1 \,\mathsf{U}\, \Phi_2)$	:	$T := \mathit{Sat}(\Phi_2)$; (* compute the smallest fixed point *) **while** $\mathit{Sat}(\Phi_1) \setminus T \cap \mathit{Pre}(T) \neq \varnothing$ **do** **let** $s \in \mathit{Sat}(\Phi_1) \setminus T \cap \mathit{Pre}(T)$; $T := T \cup \{ s \}$; **od**; **return** T;
$\exists\Box \Psi$	:	$T := \mathit{Sat}(\Psi)$; (* compute the greatest fixed point *) **while** $\exists s \in T.\, \mathit{Post}(s) \cap T = \varnothing$ **do** **let** $s \in \{ s \in T \mid \mathit{Post}(s) \cap T = \varnothing \}$; $T := T \setminus \{ s \}$; **od**; **return** T;

end switch

Symbolic model checking

- Represent sets of states and of transitions *symbolically*
  - this set-based approach is very natural for CTL
- Prominent symbolic approach:
  - encode states as binary strings, e.g., $s_0 = 0000$
  - and identify subsets of states and the transition relation by switching functions
- How to represent switching functions?
  - DNF, truth tables, binary decision trees, ...
  - CNF — this is used in SAT-based model checking
  - binary decision diagrams (ROBDDs)

Basic approach

- let $TS = (S, \rightarrow, I, AP, L)$ be a "large" finite transition system
  - the set of actions is irrelevant here and has been omitted, i.e., $\rightarrow \subseteq S \times S$
- For $n \geqslant \lceil \log |S| \rceil$, let injective function $enc : S \rightarrow \{0,1\}^n$
  - note: $enc(S) = \{0,1\}^n$ is no restriction, as all elements $\{0,1\}^n \setminus enc(S)$ can be treated as the encoding of pseudo states that are unreachable
- Identify the states $s \in S = enc^{-1}(\{0,1\}^n)$ with $enc(s) \in \{0,1\}^n$
- And $T \subseteq S$ by its characteristic function $\chi_T : \{0,1\}^n \rightarrow \{0,1\}$
  - that is $\chi_T(enc(s)) = 1$ if and only if $s \in T$
- And $\rightarrow \subseteq S \times S$ by the Boolean function $\Delta : \{0,1\}^{2n} \rightarrow \{0,1\}$
  - such that $\Delta\left(enc(s), enc(s')\right) = 1$ if and only if $s \rightarrow s'$

Switching functions

- Let *Var* $= \{z_1, \ldots, z_m\}$ be a finite set of Boolean variables
- An evaluation is a function η : *Var* $\rightarrow \{0, 1\}$
  - let *Eval*$(z_1, \ldots, z_m)$ denote the set of evaluations for $z_1, \ldots, z_m$
  - shorthand $[z_1 = b_1, \ldots, z_m = b_m]$ for $\eta(z_1) = b_1, \ldots, \eta(z_m) = b_m$
- Notations:
  - $\overline{z}$ denotes the variable tuple $(z_1, \ldots, z_m)$
  - $\overline{b}$ denotes the bit tuple $(b_1, \ldots, b_m) \in \{0, 1\}^m$
  - $[\overline{z} = \overline{b}]$ denotes $[z_1 = b_1, \ldots, z_m = b_m]$
- f : *Eval*(*Var*) $\rightarrow \{0, 1\}$ is a *switching function* for *Var* $= \{z_1, \ldots, z_m\}$
  - the switching functions for the empty variable set are just constants 0 or 1

Logical operations on switching functions

- Boolean connectives for switching functions are straightforward
  - e.g., let switching function f_1 for $\{ z_1, \ldots, z_n, \ldots, z_m \}$ and
  - switching function f_2 for $\{ z_n, \ldots, z_m, \ldots, z_k \}$ with $0 \leqslant n \leqslant m \leqslant k$
  - then the switching function $f_1 \vee f_2$ and $f_1 \wedge f_2$ for $\{z_1, \ldots, z_k\}$ are defined by:

$$(f_1 \vee f_2)([z_1 = b_1, \ldots, z_k = b_k])$$
$$= \max\Big\{ f_1([z_1 = b_1, \ldots, z_m = b_m]),\ f_2([z_n = b_n, \ldots, z_k = b_k]) \Big\} \quad \text{and}$$
$$(f_1 \wedge f_2)([z_1 = b_1, \ldots, z_k = b_k])$$
$$= \min\Big\{ f_1([z_1 = b_1, \ldots, z_m = b_m]),\ f_2([z_n = b_n, \ldots, z_k = b_k]) \Big\}$$

- Let z_i denote the *projection function* $pr_{z_i} : \mathit{Eval}(\overline{z}) \to \{ 0, 1 \}$ with

$$pr_{z_i}([\overline{z} = \overline{b}]) = b_i$$ and 0 or 1 for constant switching functions

Cofactors

Let $f : \mathit{Eval}(z, y_1, \ldots, y_m) \rightarrow \{ 0, 1 \}$ be a switching function

- The *positive cofactor* of f for variable z is the switching function $f|_{z=1}$:

$$f|_{z=1}(\mathfrak{c}, b_1, \ldots, b_m) \;=\; f(1, b_1, \ldots, b_m)$$

- The *negative cofactor* of f for z is the switching function $f|_{z=0}$:

$$f|_{z=0}(\mathfrak{c}, b_1, \ldots, b_m) \;=\; f(0, b_1, \ldots, b_m)$$

- For switching function f for $\{ z_1, \ldots, z_k, y_1, \ldots, y_m \}$, the *iterated* cofactor of f is

$$f|_{z_1=b_1,\ldots,z_k=b_k} \;=\; (\ldots(f|_{z_1=b_1})|_{z_2=b_2}\ldots)\,|_{z_k=b_k}$$

Example

Shannon expansion

- If f is a switching function for *Var*, then for each variable $z \in$ *Var*:

$$f = \underbrace{\left(\neg z \wedge f|_{z=0}\right) \vee \left(z \wedge f|_{z=1}\right)}_{\text{Shannon expansion}}$$

- Variable z is *essential* for f if $f|_{z=0} \neq f|_{z=1}$; else it is not essential

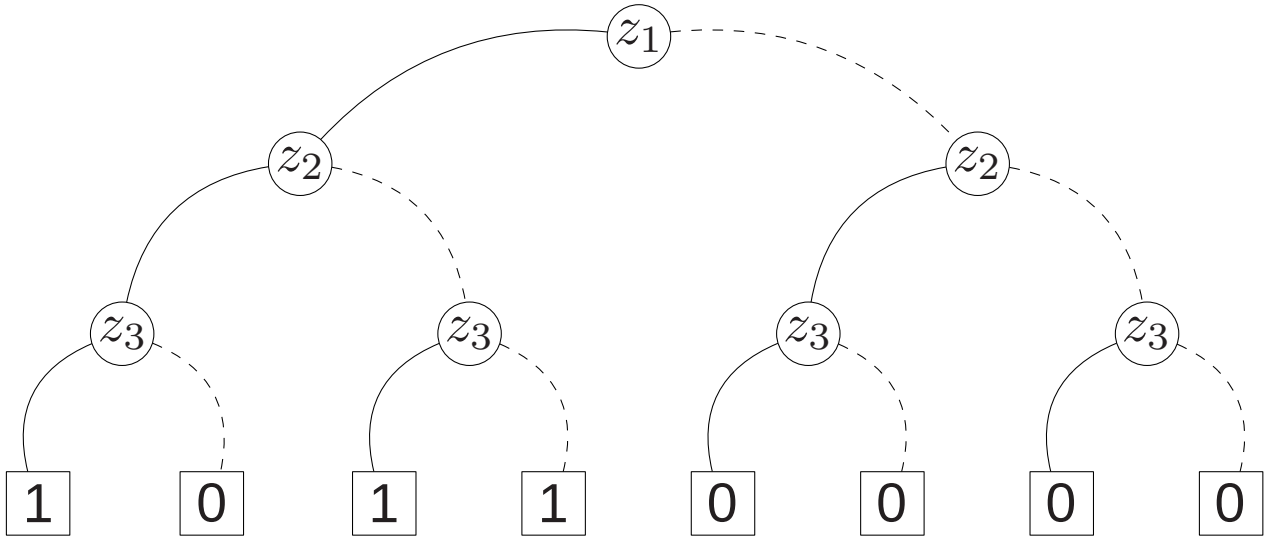
Binary decision tree

- Let *Var* be a set of Boolean variables and $<$ a total order on *Var*
- Binary decision tree (BDT) is a complete binary tree over $\langle \mathit{Var}, < \rangle$
  - each leaf v is labeled with a boolean value $\mathit{val}(v) \in \{ 0, 1 \}$
  - non-leaf v is labeled by a boolean variable $\mathit{Var}(v) \in \mathit{Var}$
  - such that for each non-leaf v and vertex w:

$$w \in \{ \mathit{left}(v), \mathit{right}(v) \} \Rightarrow (\mathit{Var}(v) < \mathit{Var}(w) \vee w \text{ is a leaf})$$

$\Rightarrow$ On each path from root to leaf, variables occur in the same order

Binary decision tree



binary decision tree (BDT) for $z_1 \wedge (\neg z_2 \vee z_3)$

satisfying assignments $[z_1 = 1, z_2 = 0, z_3 = 0]$, $[z_1 = 1, z_2 = 0, z_3 = 1]$ and $[z_1 = 1, z_2 = 1, z_3 = 1]$

Binary decision tree

- The BDT for function f on *Var* $= \{ z_1, \ldots, z_m \}$ has depth m
  - outgoing edges for node at level i stand for $z_i = 0$ (dashed) and $z_i = 1$ (solid)
- For evaluation $s = [z_1 = b_1, \ldots, z_m = b_m]$, $f(s)$ is the value of the leaf
  - reached by traversing the BDT from the root using branch $z_i = b_i$ for at level i
- The subtree of node v at level i for variable ordering $z_1 < \ldots < z_m$ represents

  the iterated cofactor $f\big|_{z_1=b_1,\ldots,z_{i-1}=b_{i-1}}$

  - which is a switching function over $\{ z_i, \ldots, z_m \}$ and
  - where $z_1 = b_1, \ldots, z_{i-1} = b_{i-1}$ is the sequence of decisions made along the path from the root to node v

Existential and universal quantification

Let f be a switching function for *Var* and $z \in$ *Var*

- $\exists z.f$ is the switching function given by $\exists z.f \;=\; f|_{z=0} \vee f|_{z=1}$
  - if $\overline{z} = (z_1, \ldots, z_k)$, then $\exists \overline{z}.f$ abbreviates $\exists z_1.\exists z_2. \ldots \exists z_k.f$
- $\forall z.f$ is the switching function given by $\forall z.f \;=\; f|_{z=0} \wedge f|_{z=1}$
  - if $\overline{z} = (z_1, \ldots, z_k)$, then $\forall \overline{z}.f$ abbreviates $\forall z_1.\forall z_2. \ldots \forall z_k.f$
- Let $f(z, y_1, y_2) \;=\; (z \vee y_1) \wedge (\neg z \vee y_2)$. Then:
  - $\exists z.f \;=\; f|_{z=0} \vee f|_{z=1} \;=\; y_1 \vee y_2$, and
  - $\forall z.f \;=\; f|_{z=0} \wedge f|_{z=1} \;=\; y_1 \wedge y_2$

Renaming

Let $\overline{z} = (z_1, \ldots, z_m)$, $\overline{y} = (y_1, \ldots, y_m)$ and $\overline{x} = (x_1, \ldots, x_k)$ such that no z_i and y_i occur in $\overline{x}$

- For evaluation $s = [\overline{y} = \overline{b}] \in \mathit{Eval}(\overline{y}, \overline{x})$, evaluation $s\{\overline{z} \leftarrow \overline{y}\}$
  - agrees with s for the variables in $\overline{x}$
  - and assigns the same value $b \in \{ 0, 1 \}$ to variable z_i as s to variable y_i

- For f on $\mathit{Eval}(\overline{y}, \overline{x})$, $f\{\overline{z} \leftarrow \overline{y}\}$ on $\mathit{Eval}(\overline{z}, \overline{x})$ is given by

$$f\{\overline{z} \leftarrow \overline{y}\}(s) \;=\; f(\, s\{\overline{z} \leftarrow \overline{y}\}\,)$$

  - that is, $f\{\overline{z} \leftarrow \overline{y}\}([\overline{y} = \overline{b}, \overline{x} = \overline{\mathfrak{c}}]) \;=\; f([\overline{z} = \overline{b}, \overline{x} = \overline{\mathfrak{c}}])$

Transition systems as switching functions

- Encode state s by $n \geqslant \lceil \log |S| \rceil$ Boolean variables $x_1, \ldots, x_n$
  - identify $[x_1 = b_1, \ldots, x_n = b_n]$ with $s \in S$ such that $\mathit{enc}(s) = (b_1, \ldots, b_n)$
  - assume $S = \mathit{Eval}(\overline{x})$
- Represent $I \subseteq S$ by the *characteristic function* $\chi_I : \mathit{Eval}(\overline{x}) \to \{0, 1\}$
- Represent labeling L by a family $(f_a)_{a \in \mathit{AP}}$ of switching functions for $\overline{x}$
  - where for $a \in \mathit{AP}$, $f_a = \chi_{\mathit{Sat}(a)}$ represents the set $\mathit{Sat}(a)$
- Represent $\to \subseteq S \times S$ by its characteristic function

Encoding the transition relation

- Represent $\rightarrow \subseteq S \times S$ by its characteristic function
  - identify $\rightarrow$ with a function $\Delta : S \times S \rightarrow \{ 0, 1 \}$ such that $\Delta(s, t) = 1$ iff $s \rightarrow t$
- Encode start and target state by $\overline{x} = (x_1, \ldots, x_n)$ and $\overline{x}' = (x'_1, \ldots, x'_n)$
  - that is, for each variable x_i introduce a (copy) variable x'_i
- Represent the transition relation $\rightarrow$ by the switching function

$$\Delta : \mathit{Eval}(\overline{x}, \overline{x}') \rightarrow \{ 0, 1 \} \quad \text{with} \quad \Delta(s, t\{\overline{x}' \leftarrow \overline{x}\}) = \begin{cases} 1 & \text{if } s \rightarrow t \\ 0 & \text{otherwise} \end{cases}$$

Example encoding transition relation

- Let $S = \{ s_0, s_1 \}$ and $s_0 \to s_0$, $s_0 \to s_1$, and $s_1 \to s_0$
- Use a single Boolean variable $x_1 = x$ for the encoding
  - say *enc*$(s_0) = 0$ and *enc*$(s_1) = 1$
- $\to$ is represented by switching function Δ : *Eval*$(x, x') \to \{ 0, 1 \}$ with

$$\Delta = \neg x \vee \neg x'$$

  - satisfying assignments are: $\underbrace{[x = 0, x' = 0]}_{s_0 \to s_0}$, $\underbrace{[x = 0, x' = 1]}_{s_0 \to s_1}$, and $\underbrace{[x = 1, x' = 0]}_{s_1 \to s_0}$

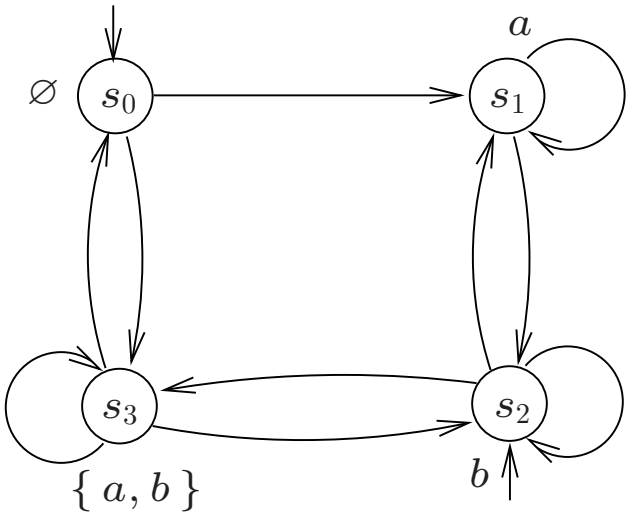
Explicitly representing transition systems

$TS = (S, \rightarrow, I, AP, L)$ with $|S| = n$ and $|AP| = k$:

- Identify the n states by numbers
- Represent the set of initial states I as boolean vector $\underline{i}$
  - $\underline{i}(s_j) = 1$ if and only if state $s_j \in I$
- Represent $\rightarrow$ by a boolean matrix $\mathbf{T}$ of size $n \times n$
  - $\mathbf{T}(s_i, s_j) = 1$ if and only if $s_i \rightarrow s_j$
- Represent L by an $n \times k$-boolean matrix $\mathbf{L}$
  - $\mathbf{L}(s_i, a_j) = 1$ if and only if $a_j \in L(s_i)$

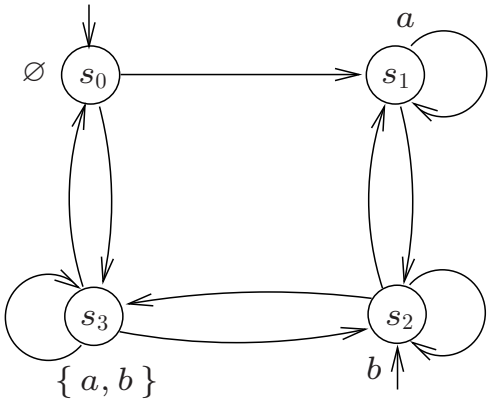
$\Rightarrow$ Use sparse matrix representations for $\mathbf{T}$ and $\mathbf{L}$

Explicit representation: an example



$$\underline{i} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{T} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{L} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Symbolic representation: an example (1)



- States:

state	*enc*(s_i)	*switching function*
s_0	$(0,0)$	$\neg\, x_1 \wedge \neg\, x_2$
s_1	$(0,1)$	$\neg\, x_1 \wedge x_2$
s_2	$(1,0)$	$x_1 \wedge \neg\, x_2$
s_3	$(1,1)$	$x_1 \wedge x_2$

- Initial states:

$$\chi_I(x_1, x_2) = (\neg\, x_1 \wedge \neg\, x_2) \;\vee\; (x_1 \wedge \neg\, x_2)$$

Symbolic representation: an example (2)

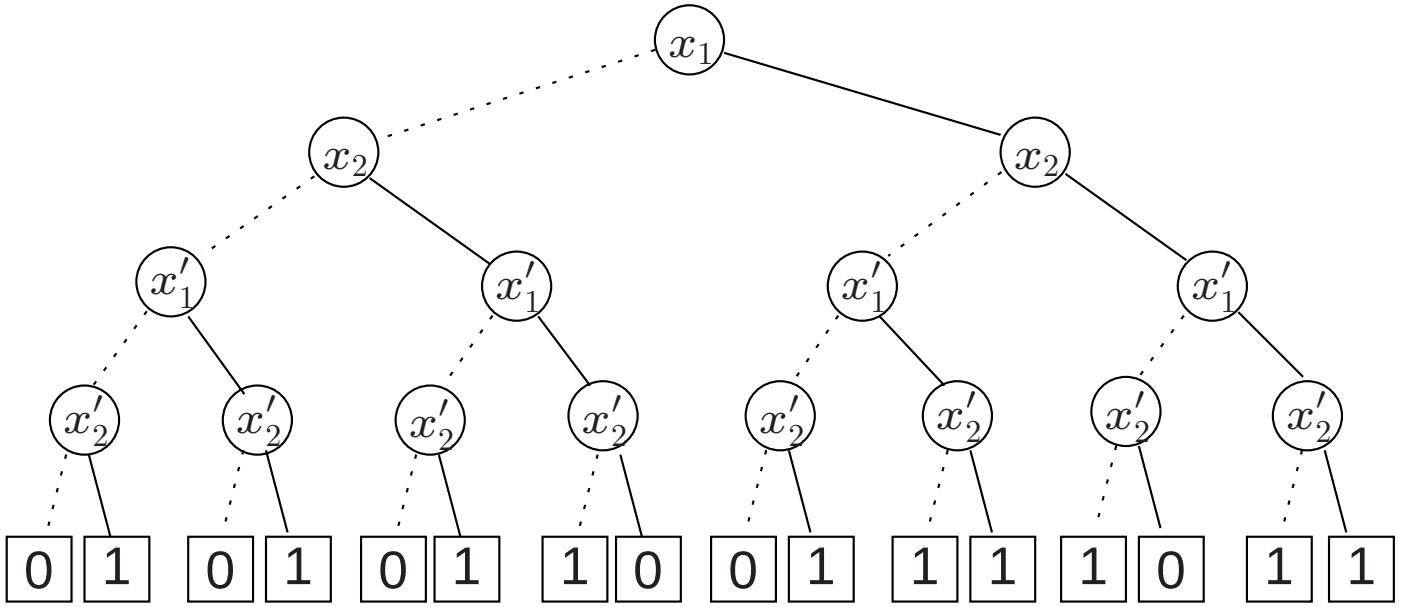
- Transition relation:

Δ	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$(0,0)$	0	1	0	1
$(0,1)$	0	1	1	0
$(1,0)$	0	1	1	1
$(1,1)$	1	0	1	1

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

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

Transition relation as a BDT



A BDT representing Δ for our example using ordering $x_1 < x_2 < x'_1 < x'_2$

Successor sets

- A switching function for $\mathit{Post}(s) = \{\, s' \in S \mid s \rightarrow s' \,\}$ is obtained by:

$$\chi_{\mathit{Post}(s)} \;=\; (\Delta|_{x_1=b_1,\ldots,x_n=b_n})\,\{\overline{x}' \leftarrow \overline{x}\}$$

- Example for our two-state transition system: $\mathit{Post}(s_0) = \{\, s_0, s_1 \,\}$

$$(\Delta|_{x=0})\,\{x' \leftarrow x\} \;=\; \underbrace{((\neg x \vee \neg x')|_{x=0})}_{=1}\{x' \leftarrow x\} \;=\; 1$$

- And for $\mathit{Post}(s_1) = \{\, s_0 \,\}$ we obtain

$$(\Delta|_{x=1})\,\{x' \leftarrow x\} \;=\; \underbrace{((\neg x \vee \neg x')|_{x=1})}_{=\neg x'}\{x' \leftarrow x\} \;=\; \neg x$$

Symbolic model checking (1)

- For the propositional fragment of CTL, *Sat* can be computed symbolically
- What about the temporal operators $\exists\bigcirc$, $\exists\,\mathsf{U}$ and $\exists\Box$?
- Enumeratively:

$$\mathit{Sat}(\exists\bigcirc\,\Phi) \;=\; \{\, s \in S \mid \mathit{Post}(s) \cap \mathit{Sat}(\Phi) \neq \varnothing \,\}$$

- Symbolically:

$$\chi_{\mathit{Sat}(\exists\bigcirc\,\Phi)} \;=\; \exists \overline{x}'.\,(\; \underbrace{\Delta(\overline{x},\overline{x}')}_{s' \,\in\, \mathit{Post}(s)} \;\wedge\; \underbrace{\chi_{\mathit{Sat}(\Phi)}(\overline{x}')}_{s' \in \mathit{Sat}(\Phi)} \;)$$

Symbolic model checking (2)

Computation of $\mathit{Sat}(\exists(\Phi \,\mathsf{U}\, \Psi))$:

$T_0 := \mathit{Sat}(\Psi); j := 0;$
repeat
$T_{j+1} := T_j(\overline{x}) \cup \Big(\mathit{Sat}(\Phi) \cap \{\, s \in S \mid \exists s' \in S.\, s' \in \mathit{Post}(s) \cap T_j \,\}\Big);$
$j := j + 1;$
until $T_j = T_{j-1}$
return T_j.

Symbolic computation of $\mathit{Sat}(\exists(\Phi \,\mathsf{U}\, \Psi))$:

$f_0(\overline{x}) := \chi_{\mathit{Sat}(\Psi)}(\overline{x}); j := 0;$
repeat
$f_{j+1}(\overline{x}) := f_j(\overline{x}) \vee \Big(\chi_{\mathit{Sat}(\Phi)}(\overline{x}) \wedge \exists \overline{x}'.\, (\underbrace{\Delta(\overline{x},\overline{x}')}_{s' \in \mathit{Post}(s)} \wedge \underbrace{f_j(\overline{x}')}_{s' \in T_j})\Big);$
$j := j + 1;$
until $f_j(\overline{x}) = f_{j-1}(\overline{x})$
return $f_j(\overline{x})$.

Symbolic model checking (3)

Computation of $\mathit{Sat}(\exists\Box\Phi)$:

$T_0 := \mathit{Sat}(\Phi); j := 0;$
repeat
  $T_{j+1} := T_j \cap \{ s \in S \mid \mathit{Post}(s) \cap T_j \neq \varnothing \};$
  $j := j + 1$
until $T_j = T_{j-1}$;
return T_j.

Symbolic computation of $\mathit{Sat}(\exists\Box\Phi)$:

$f_0(\overline{x}) := \chi_{\mathit{Sat}(\Phi)}(\overline{x}); j := 0;$
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 $f_j(\overline{x})$.