Lecture 9: Symbolic Model Checking

- Symbolic Model Checking Algorithm
- Representing Transition Systems with BDDs
- Ordered Binary Decision Diagrams (BDDs)
- Basic CTL Fixpoint Theorems
Breakthrough: Symbolic Model Checking

• BDDs traditionally used to represent boolean functions.
• Can handle much larger designs – hundreds of state variables.
• Uses boolean encoding for state machines and sets of states.

Method used by most “industrial strength” model checkers.


Carl Pixley at Motorola independently developed a similar algorithm, as did the French researchers, Coudert and Madre.

BDDs enabled handling much larger concurrent systems. (usually, an order of magnitude increase in hardware latches!)
$d \in \text{EF} \land d = \text{EF}$
Fixpoint Algorithms (cont.)

Key properties of $\text{EF}$:

1. $\Omega \Omega \text{EX} \land d = \Omega$
2. $\Omega \text{EX} \land d = \Omega$
3. $\text{EX} \land d = \Omega$

We write $d \in \text{LFP}(\text{EF} \cup \Omega \text{EX})$. How to compute $\text{EF}$:

$\Omega \subseteq d \in \text{LFP}(\text{EF} \cup \Omega \text{EX})$ implies $d \in \text{EF} \cup \Omega \text{EX} \cup d = \Omega$

Key properties of $\text{EF}$:
\( \emptyset = ^0\Omega \)

\[
\mathcal{d} \mathcal{M} = ^0s \mathcal{N}
\]
\( \exists d \in I \)
\[ \text{\textsuperscript{1}}\Omega \text{x} \Omega \wedge d = \text{\textsuperscript{1}}\Omega \]

\[ \text{\textsuperscript{d}d} \text{ \textsuperscript{d}d} = 0s \text{ } \mathcal{W} \]
\( \exists \Omega \exists d = \varepsilon \Omega \)

\[ \exists d \mathcal{E} N \Rightarrow 0s \mathcal{W} \]
Ordered Binary Decision Trees and Diagrams

Ordered Binary Decision Trees for the two-bit comparator, given by the formula

\[ (q_2 \leftrightarrow q_2) \lor (q_1 \leftrightarrow q_1) = (q_2, q_1, q_1, q_1) \]

is shown in the figure below:
An Ordered Binary Decision Diagram (OBDD) is an ordered decision tree where

- All isomorphic subtrees are combined, and
- All nodes with isomorphic children are eliminated.

Given a parameter ordering, OBDD is unique up to isomorphism.


From Binary Decision Trees to Diagrams
If we use the ordering $a_2 > a_1 > b_2 > b_1$, we obtain the OBDD below:

---

**OBDD for Comparator Example**
Variable Ordering Problem

If we use the ordering function, for the comparator function, we get the OBDD below:
Moreover, there are boolean functions that have exponential size OBDDs for any variable ordering. For an \( n \)-bit comparator:

- If we use the ordering \( a_1 > \cdots > a_n > 1 \), the number of vertices is \( 3^n + 2 \).
- If we use the ordering \( a_1 > \cdots > 1 > a_n \), the number of vertices is \( 3^n + 2 \).

An example is the middle output (or one output) of a combinational circuit to multiply two \( n \) bit integers.

Variable Ordering Problem (Cont.)
Logical operations on OBDDs

Logical negation: Replace each leaf by its negation

Logical conjunction: Use Shannon's expansion as follows.

Always combine isomorphic subtrees and eliminate redundant nodes.

To break problem into two subproblems, solve subproblems recursively:

\[(\overline{a} \cdot \overline{b} \cdot f) \cdot a + (\overline{a} \cdot \overline{b} \cdot f) \cdot \overline{a} = \overline{b} \cdot f\]

Number of subproblems bounded by

Hash table stores previously computed subproblems.
Using the above operations, we can build up OBDDs for complex boolean functions from simpler ones.

1. For all nodes $k$ in the tree:
   
   - Replace all nodes by right sub-tree.
   
   
   $\nu |f \wedge \nu |f = f : \forall \exists$

2. By definition:

   $(p,q,c,f) : \forall \exists$ Boolean quantification.

Logical operations (cont.)
Symbolic Model Checking Algorithm

How to represent state-transition graphs with Ordered Binary Decision Diagrams:

Assume that system behavior is determined by boolean state variables $u_1, \cdots, u_n$.

The transition relation $N$ will be given as a boolean formula in terms of the state variables:

$\left( u_{\alpha}, \cdots, u_{\gamma}, u_{\alpha}, \cdots, u_{\gamma} \right)^N$

Now convert $N$ to an OBDD!!

where $v_1, \cdots, v_k$ represents the current state and $v_{k+1}, \cdots, v_n$ represents the next state.
Symbolic Model Checking (cont.)

Representing transition relations symbolically:

\[
\begin{align*}
(q' & \lor p' \lor q \lor q) \land \\
(q' \lor p' \lor q \lor q) \land \\
(q' \lor p' \lor q' \lor q)
\end{align*}
\]

Now, represent as an OBDD!

Boolean formula for transition relation:

Representing transition relations symbolically:

Symbolic Model Checking (cont.)
Consider EX.

Now, introduce state variables and transition relation:

$\text{Compute OBDD for relational product on right side of formula.}$

$[(\checkmark d \land (\checkmark N) N] \land \mathcal{E} = (\checkmark) f$

Consider $d \mathbf{EX} = f$.
Convergence can be detected since the sets of states \( \Omega \) are represented as OBDDs.

Until convergence:

\[
\cdots (\varnothing \Omega)^0 \cap (\varnothing \Omega)^1 \cap (\varnothing \Omega)^2 \cap (\varnothing \Omega)^3 \cdots
\]

Now, compute the sequence

\[
[(\varnothing \Omega) \lor (\varnothing \Omega)] \varnothing \in \land (\varnothing \Omega)^d \Omega d \Omega = d \Omega\ EF
\]

Introduce state variables:

\[
\Omega \land (\varnothing \Omega)^d \Omega d \Omega = d \Omega\ EF
\]

How to evaluate fixpoint formulas using OBDDs:

Symbolic Model Checking (cont.)