Design Verification

Lecture 18 - Model Checking III

1. Symbolic Model Checking (SMC)

→ So far, we are checking a given property by traversing the FSM and find the fixed point at which the states obtained are closed. What happens when the size of FSM is huge?

→ Instead of explicitly traversing the FSM, how about implicitly traversing the FSM via use of BDD's?

Recall that in explicit traversal, computation of IMG(A) is:

\[
\text{for each state } s \text{ in } A \\
\quad \text{for each input combination } i \\
\quad \text{nextState } t = \text{next\_state}(s, i); \\
\quad \text{addState}(A, t);
\]

Problem with this approach: set \( A \) gets extremely large. Look for a better data structure such that we can still support addState, for each state, etc.

2. SMC - Characteristic functions and BDD’s

Let us represent all the states in initial set \( A \) as \( F_A() \) where \( F_A(q_0, q_1, \ldots, q_n) = 1 \)

if state represented by \((q_0, q_1, \ldots, q_n) \in A\)

Thus, addState\((A, s)\) is easy, so is for\_each\_state().

Let us represent the transition relation functions the same way: Let \( y_0, y_1, \ldots, y_n \) and \( x_0, x_1, \ldots, x_n \) be representations for the next state and present state, respectively. And let \( u_0, u_1, \ldots, u_k \) be the primary inputs.

Define \( F_T(y_0, y_1, \ldots, y_n, x_0, x_1, \ldots, x_n, u_0, \ldots u_k) = 1 \) exactly when \((y_0, \ldots, y_n)\) is the next state of \((x_0, \ldots, x_n)\) when input \((u_0, \ldots, u_k)\) is applied.

Define \( G(y_0, \ldots, y_n) = \)

\[
(\exists x_0 \ldots \exists x_n \exists u_0 \ldots \exists u_k)[F_A(x_0, \ldots, x_n)F_T(y_0, \ldots, y_n, x_0, \ldots, x_n, u_0, \ldots, u_k)]
\]

→ \( G(y_0, \ldots, y_n) = 1 \) exactly when there exists values for \( x_0, \ldots, x_n, \) and \( u_0, \ldots u_k \) such that \( F_A(x_0, \ldots, x_n) = 1 \) and the next state of \((x_0, \ldots x_n)\) on inputs \((u_0, \ldots, u_k)\) is \((y_0, \ldots, y_n)\).

→ Build BDD's for \( F_T, F_A, F_TF_A \).
→ How to compute $\exists x F$? Recall that $\exists x F = F_x + F_x$
Computing $IMG(A)$ using BDD’s?

$IMG(A, F_A, F_T, PS, PI, NS)$
/* $PS, PI$, and $NS$ are variables representing present state, primary inputs, and next state */
begin
    $bdd_1 = bdd\_and(F_T, F_A)$;
    $bdd_2 = bdd\_exists(bdd_1, PS + PI$ variables);
    $bdd_3 = bdd\_compose(bdd_2, NS replaced by PS variables)$;
end

3. Processing CTL formulas

Now with BDD’s, can we compute $EX f$?

→ Suppose the states that satisfies $f$ is $A$. We need to compute the preimage of $A$. This can easily be done by $Pre - image(x_0, ..., x_n) =$
$(\exists y_0 \ldots \exists y_n, \exists u_0 \ldots \exists u_k)[F_T(y_0, ..., y_n, x_0, ..., x_n, u_0, ..., u_k)F_A(y_0, ..., y_n)]$

Example 1
Example 2

→ How about computing $E(f \cup g)$ and $EG f$?
Solution: use successive Pre-image computations.
4. Bottlenecks in Symbolic Model Checking

- size of BDD - dynamic variable ordering to keep it small (but don’t waste too much time on re-ordering)
- Partition transition relation $F_T$, since the BDD for $F_T$ is generally very large.

But what problems do we have?

$\exists x_0 \ldots \exists x_n \exists u_0 \ldots \exists u_k [y_0 \oplus f_1(x_0, \ldots, x_n, u_0, ..., u_k) \ldots y_n \oplus f_n(x_0, ..., x_n, u_0, ..., u_k) F_A(x_0, ..., x_n)]$

In general, $\exists x [p(x, y) q(x, y)] \neq [(\exists x) p(x, y)] [(\exists x) q(x, y)]$

Why? And when will it be safe to distribute existential quantification?

5. Model Checking Using ATPG

→ Based on transformation of the underlying circuit, and generate a test for some stuck-at fault. This is especially useful for computing the CTL formula $EF \phi$, where $\phi$ represents some state, internal node combination, or output.

Example 4
6. How about $EX\phi$?

7. $EG\phi$
8. $E(\phi U \psi)$