Symbolic Finite-State Verification

Enumerative methods can handle systems of sizes up to 10^7 ($\sim 2^{24}$) states. The situation has greatly improved with the introduction of Symbolic model-checking methods which can standardly handle systems with up to 2^{150} states.

Symbolic methods are based on set-oriented algorithms in which all the immediate successors (predecessors) of a given set of states can be computed in one step. Their widely spread application has been made possible only due to a highly efficient representation of boolean assertions by the Ordered Binary Decision Diagrams (OBDD) data structure.

Symbolic Model Checking

Define the existential predecessor predicate transformer:

$$\rho \diamond \psi \quad = \quad \exists V' : \rho(V, V') \wedge \psi(V')$$

Obviously

$$s \models \rho \diamond \psi \quad \text{iff} \quad \text{some } \rho\text{-successor of } s \text{ satisfies } \psi.$$

For example, for a transition relation $\rho : x' = x + 1$ and assertion $\psi : x = 5$ the predecessor computation yields

$$\begin{aligned}(x' = x + 1) \diamond (x = 5) \quad &= \quad \exists x' : x' = x + 1 \;\wedge\; x' = 5 \\ &\sim \quad x + 1 = 5 \quad \sim \quad x = 4\end{aligned}$$

characterizing all the states whose ρ-successor satisfies $x = 5$.

Here and elsewhere, we employ the useful simplification rule

$$\exists y : y = e \;\wedge\; p \quad \sim \quad p[y \leftarrow e],$$

where $p[y \leftarrow e]$ is obtained from p by replacing every occurrence of variable y by the expression e.

A Symbolic Algorithm for Model Checking Invariance

Algorithm SMC-INV $(\mathcal{D}, p)$: **assertion** — Check that FDS $\mathcal{D}$ satisfies $Inv(p)$, using symbolic operations

new, old : **assertion**

1. $old := 0$
2. $new := \neg p$
3. **while** $(new \neq old)$ **do**
   begin
4. $old := new$
5. $new := new \vee (\rho_{\mathcal{D}} \diamond new)$
   end
6. **return** $\Theta_{\mathcal{D}} \ \wedge \ new$

The algorithm returns an assertion characterizing all the initial states from which there exists a finite path leading to violation of p. It returns the empty (false) assertion iff $\mathcal{D}$ satisfies $Inv(p)$.

Illustrate on MUX-SEM

We iterate as follows:

$$\varphi_0 : \quad \pi_1 = C \wedge \pi_2 = C$$

$$\varphi_1 : \quad \varphi_0 \vee \begin{pmatrix} \cdots \\ \vee\, \pi_1 = T \wedge y = 1 \wedge \pi_1' = C \wedge y' = 0 \\ \vee\, \pi_2 = T \wedge y = 1 \wedge \pi_2' = C \wedge y' = 0 \end{pmatrix} \diamond (\pi_1 = \pi_2 = C)$$

$$\sim$$

$$\pi_1 = \pi_2 = C \vee \pi_1 = T \wedge \pi_2 = C \wedge y = 1 \vee \pi_1 = C \wedge \pi_2 = T \wedge y = 1$$

$$\varphi_2 : \quad \varphi_1 \vee \pi_1 = N \wedge \pi_2 = C \wedge y = 1 \vee \pi_1 = C \wedge \pi_2 = N \wedge y = 1$$

$$\varphi_3 : \quad \varphi_2 \vee \pi_1 = C \wedge \pi_2 = C \wedge y = 0 \quad \sim \quad \varphi_2$$

The last equivalence is due to the general property $p \vee (p \wedge q) \sim p$.

If we intersect φ_3 with the initial condition $\Theta : \pi_1 = N \wedge \pi_2 = N \wedge y = 1$ we obtain 0 (false). We conclude that MUX-SEM satisfies $Inv(\neg(\pi_1 = C \wedge \pi_2 = C))$.

Symbolic Exploration Progresses in Layers