Sequential Equivalence

General verfication problem Given two sequential systems (finite state machines), M_1 and M_2, determine if they have the same input/output behavior, i.e. M_1 and M_2 produce the same output sequence for the same input sequence.

Restricted verification problem Given a finite state machine, M, with a single output $\lambda(x,e)$ over the output alphabet $\{0,1\}$, determine if M always produce the output value 1 for each possible input sequence.

Product machine Let $M_1 = (Q_1, I, O, \delta_1, \lambda_1, q_1)$ and $M_2 = (Q_2, I, O, \delta_2, \lambda_2, q_2)$ be two sequential systems. The **product machine** $M = (Q, I, O, \delta, \lambda, q_0)$ is defined by

- $Q = Q_1 \times Q_2$
- $\delta((s_1, s_2), e) = (\delta(s_1, e), \delta(s_2, e))$
- $\lambda((s_1, s_2), e) = (\lambda_1(s_1, e) \equiv \lambda_1(s_2, e), \cdots \lambda_m(s_1, e) \equiv \lambda_m(s_2, e))$, where m is the number of output bits (bitwise comparison)
- $q_0 = (q_1, q_2)$

Symbolic verification

- Given a set of states, each state are encoded using n state bits ($s_0 s_1 \cdots s_n$. Each such state can be represented by a boolean formula over these state bits.
- A set can be represented using a boolean function (and a BDD) through a **characteristic function** over the element encoding.
- A state transition relation is after all a set and we can use BDD to represent it.

Operator #1: Generalized cofactor *Shannon decomposition* is performed w.r.t. **literals**, x_i and $\overline{x_i}$, as in:

$$f = x_i f_i + \overline{x_i} f_i.$$

Shannn decomposition is performed relative to a special function (one variable) but we can generalize this to a general function.

Let $f, g \in B^n$, and let

$$f = g \cdot f_g + \overline{g} \cdot f_{\overline{g}}$$

be a decomposition of f w.r.t. the orthonormal set $\{g, \overline{g}\}$. Then the cofficient f_g is called **positive generalized cofactor** of f w.r.t. g and the coefficient $f_{\overline{g}}$ is called **negative generalized cofactor** of f w.r.t. g.

Operator #2: Constrain operator Usually, generalized cofactors of a function f w.r.t. a function g is not uniquely determined.

Let the variables $x_1, \cdots, x_n$ be ordered in the order π according to $x_{j_1} < x_{j_2} < \cdots < x_{j_n}$. Let $r = (r_1, \cdots, r_n), s = (s_1, \cdots, s_n) \in B^n$. the **distance** $\| r - s \|$ of r and s w.r.t. the order π is defined by

$$\| r - s \| = \sum_{i=1}^{n} |r_{j_i} - s_{j_i}| 2^{n-i}.$$

For $f, g \in B^n$, the **constrain operator** $f \downarrow g$ is defined by

$$(f \downarrow g)(r) = \begin{cases} f(r) & \text{if} \quad g(r) = 1, \\ f(s) & \text{if} \quad g(r) = 0, g(s) = 1 \text{ and } \| r - s \| \text{ minimal}, \\ 0 & \text{if} \quad g = 0 \end{cases}$$

Operator #3: Quantification For $f \in B^n$, the **existential quantification w.r.t. the variable** x_i is defined by

$$\exists_{x_i} f = f_{x_i} + f_{\overline{x_i}}.$$

The **universal quantification w.r.t.** x_i is defined by

$$\forall_{x_i} f = f_{x_i} \cdot f_{\overline{x_i}}.$$

Operator #4: Restrict operator

Reachability analysis Reachability analysis denotes the efficient computation and compact representation of all states which can be reached from the initial state.

Let $M = (Q, I, O, \delta, \lambda, q_0)$ be a finite state machine. A state $s \in B^n$ is said to be **reachable in exactly** k **steps from the state** r if there is an input sequence $e_0, \cdots, e_{k-1}$ and a state sequence $s_0, \cdots, s_k$ s.t. $s_0 = r, s_k = s$ and

$$\delta(s_i, e_i) = s_{i+1}, \quad 0 \le i \le k$$

Images For a finite state machine M with p input bits, n state bits, and next-state function $\delta : B^{n+p} \to B^n$, let

$$\chi_k(x_1, \cdots, x_n) : B^n \to B$$

denote the **characteristic function** of all states that are reachable in at most k steps.

Let $f : B^n \to B^m$. The **image** $Im(f)$ of the function f is defined by

$$Im(f) = \{v \in B^m : \text{ there exists some } x \in B^n \text{ s.t. } f(x) = v\}.$$

For a subset C of B^n, the **image of** f **w.r.t.** C is defined by

$$Im(f, C) = \{v \in B^m : \text{ there exists some } x \in C \text{ s.t. } f(x) = v\}.$$

Reachability algorithm based on image computation The following algorithm, given a state machine M, computes the set $Reachable$ of reachable states.

```
Traverse(δ, q₀)
/* R: aet of reachable states */
R ← S₀
From ← S₀
repeat
      /* compute image of From */
      To ← Im(δ, From)
      /* newly-reached states */
      New ← To − R
      /* update reachable sets */
      R ← R ∪ New
      From ← New
  until New = ∅
return R
```

Image computation