FORMAL VERIFICATION OF FINITE STATE MACHINES

Why?

- Bugs in hardware are expensive
- Simulation is inadequate to guarantee correctness
- Synthesis tools may also have bugs
- A specification may not match the specifier's intent

Approaches for Formal Verification

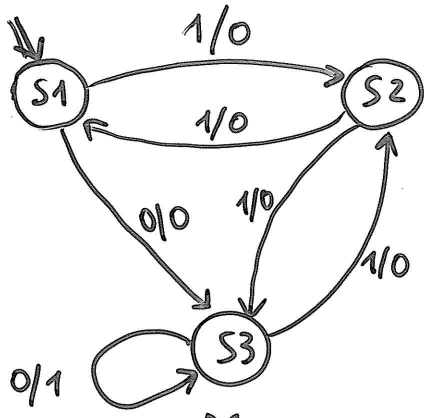
- Process algebras and trace theory
  (CCS, CSP)

- Automatic theorem provers
  (Boyer-Moore, HOL)

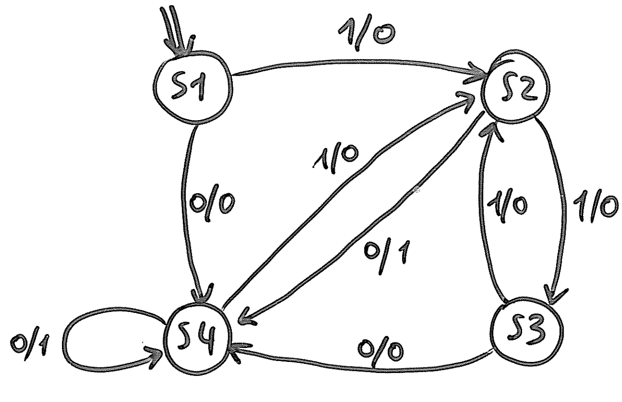
- Temporal logic
  (CTL)

- Finite state models

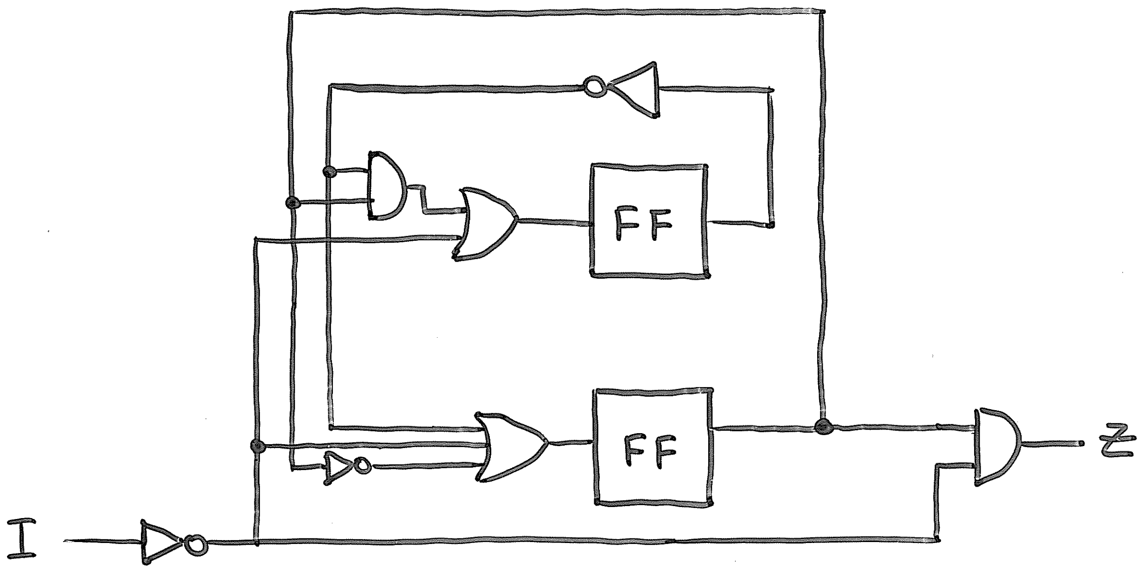
Formal verification of FSMs



M_1



M_2



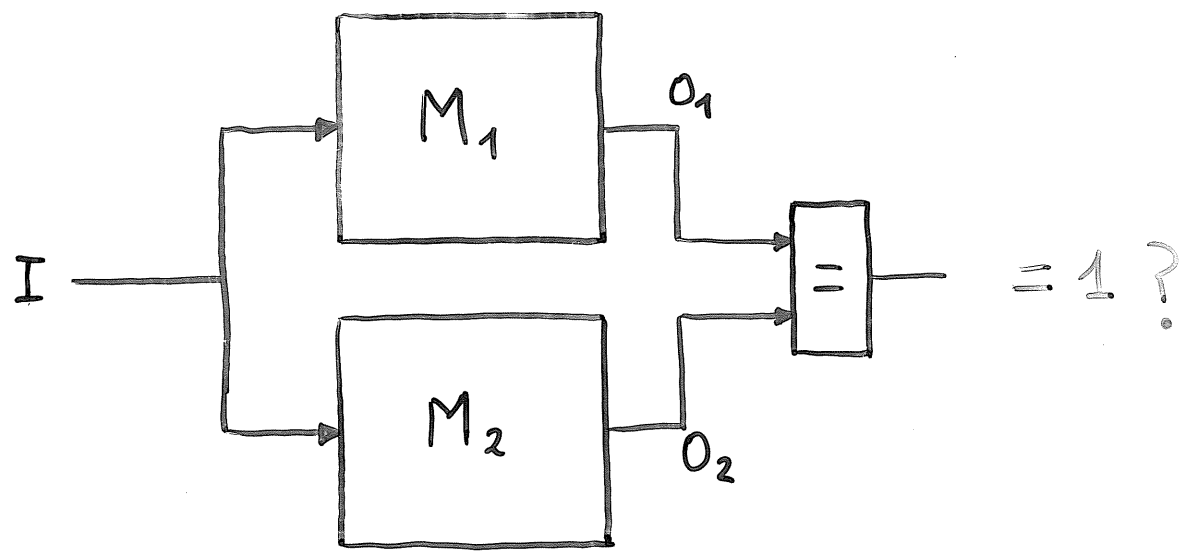
C

- Does M_1 have property X ?
- Are M_1 and M_2 equivalent ?
- Are M_1 and C equivalent ?

Equivalence checking

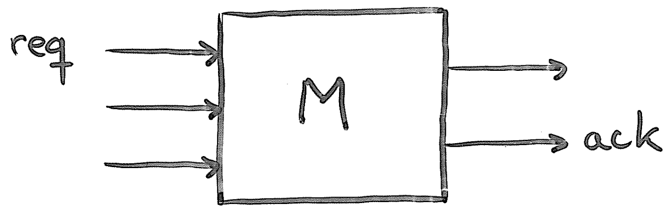
1.- Calculate the Product Machine

2.- Verify "observational equivalence" from the reset state



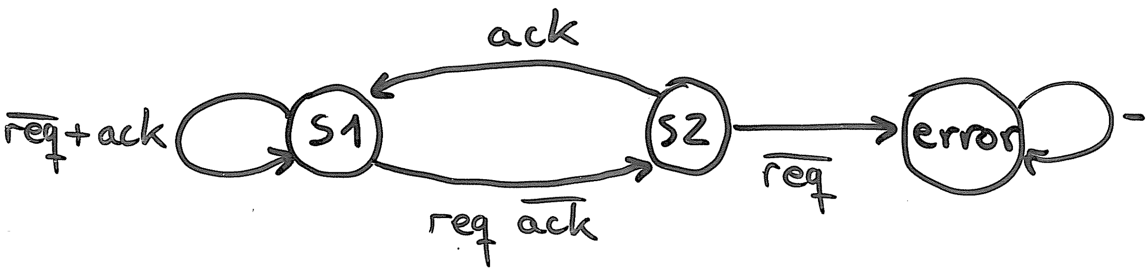
- State: $S = (S_1, S_2)$
- Transition function: $\delta(S, I) = (\delta_1(S_1, I), \delta_2(S_2, I))$
- Output: $O = \overline{O_1 \oplus O_2}$

Verification of properties

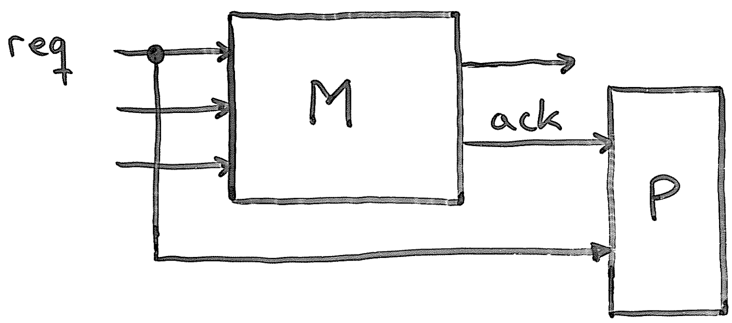


After req rising, ack will always fall before req falling

- FSM for the property (possibly non-deterministic)

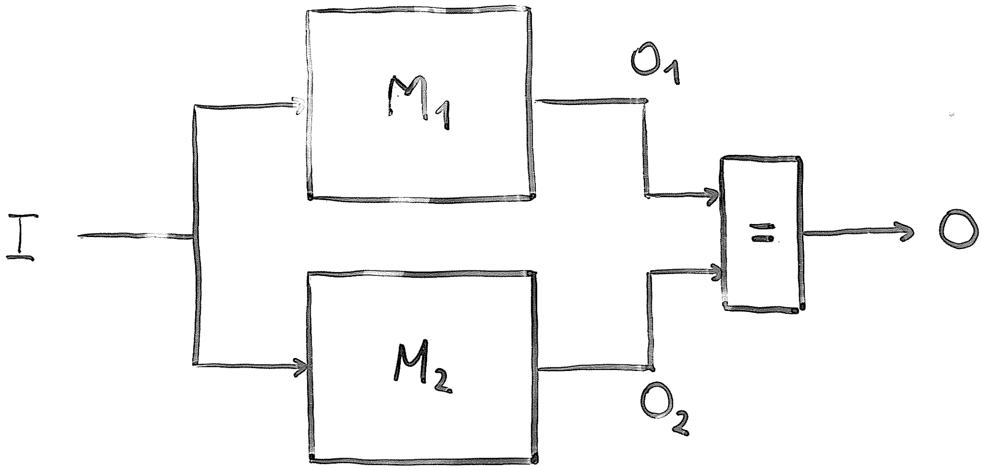


- Verification



correct if the product machine never reaches a state (*, error)

Building a product machine



for equivalence checking:

$$M_1 = (I, S_1, O_1, \delta_1, \lambda_1, S_1^\circ)$$

$$M_2 = (I, S_2, O_2, \delta_2, \lambda_2, S_2^\circ)$$

I: input alphabet, O: output alphabet

S: set of states, S°: set of initial states

δ: transition function λ: output function

$$M = M_1 \times M_2 = (I, S, O, \delta, \lambda, S^\circ)$$

$$S \subseteq S_1 \times S_2 \quad \Leftarrow \text{must be calculated}$$

$$O = \{0, 1\}$$

$$\delta = (\delta_1, \delta_2)$$

$$\lambda = \lambda_1 \oplus \lambda_2$$

$$S^\circ = (S_1^\circ, S_2^\circ)$$

Reachability Analysis (FSM traversal)

- Algorithm for explicit enumeration

  RS : set of reachable states

  PS : set of states to be processed

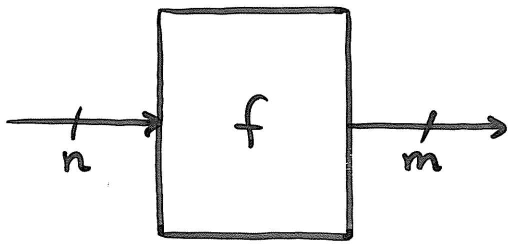
```
RS = {s0}
PS = {s0}
while PS ≠ ∅ do
    ns = first_element (PS) ;   PS = PS - {ns}
    for each state ss successor of ns do
        if ss ∉ RS then
            RS = RS ∪ {ss}
            PS = PS ∪ {ss}
        end if
    end for
end while
```

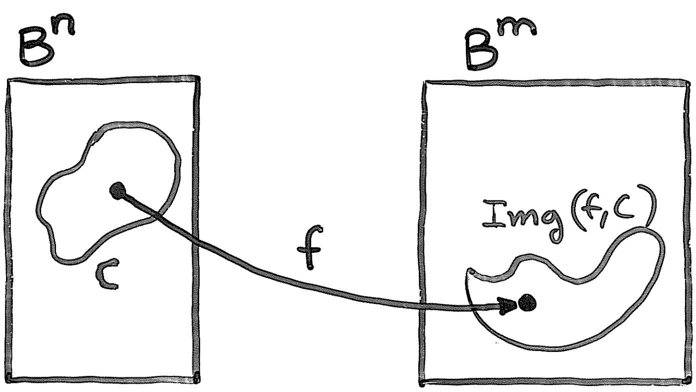
- RS contains all reachable states
- Data structure for PS
  - LIFO ⟶ Depth-First Traversal
  - FIFO ⟶ Breadth-First Traversal
- Problem: the number of states can be very large (exponential on the number of state signals)

Symbolic traversal

- Several states processed in parallel
- Usually breadth-first traversal
- Symbolic representation of
  - sets of states
  - transition function

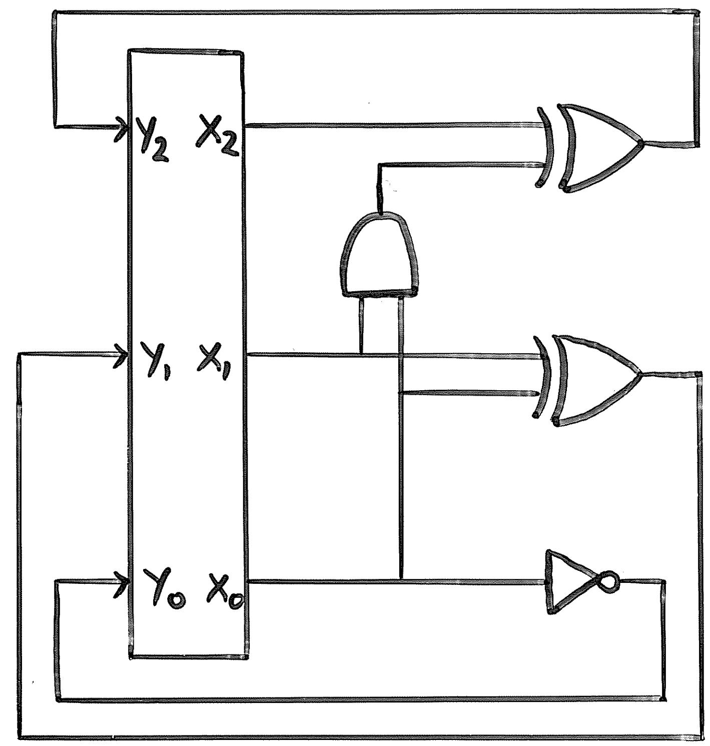
Image computation





$$Img\,(f, C) = \{\, y \in B^m \mid \exists x \in C,\ f(x) = y \,\}$$

Example : Modulo 8 counter



$X = (x_2, x_1, x_0)$

$Y = (y_2, y_1, x_0)$

$Y = (X + 1) \bmod 8$

Transition function: $\quad Y = \delta(X)$

$$(y_2, y_1, y_0) = (\delta_2(X), \delta_1(X), \delta_0(X))$$

$$y_0 = \delta_0(X) = \overline{x_0}$$

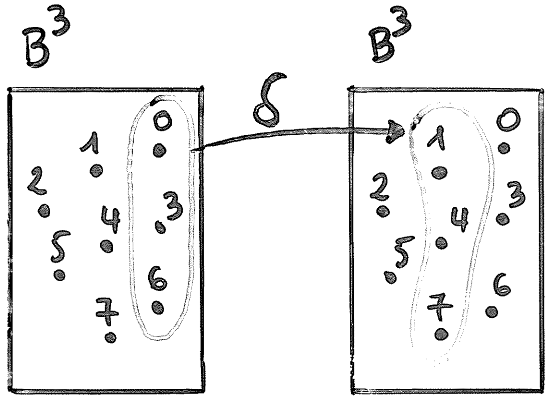
$$y_1 = \delta_1(X) = x_0 \oplus x_1$$

$$y_2 = \delta_2(X) = (x_0 x_1) \oplus x_2$$

Image computation (example)

$\mathrm{Img}(\delta, \text{"multiples of 3"})$

$C = \{0, 3, 6\}$



- Sets of states can be represented by characteristic functions

$$C(x) = \overline{x_0} \cdot \overline{x_1 \oplus x_2} + \overline{x_2}\, x_1\, x_0$$

$$= \overline{x_2}\,\overline{x_1}\,\overline{x_0} + \overline{x_2}\, x_1\, x_0 + x_2\, x_1\, \overline{x_0}$$

Transition relations

- The transition function defines a <u>relation</u> between pairs of states

$$x \, R \, y \iff y = \delta(x)$$

- Relations can be represented by boolean formulae

$$x \, R \, y \iff T_R(x, y) = 1$$

- Let M be an FSM and $\delta = (\delta_1, \dots, \delta_m)$ its transition function. The <u>transition relation</u> of M is

$$T_R(x_1, \dots, x_n, y_1, \dots, y_m) = \bigwedge_{i=1}^{m} \left(y_i \equiv \delta_i(x_1, \dots, x_n) \right)$$

$$a \equiv b \iff \overline{a \oplus b}$$

Transition relation (example)

$$y_0 = \delta_0(x) = \overline{x}_0$$
$$y_1 = \delta_1(x) = x_0 \oplus x_1$$
$$y_2 = \delta_2(x) = (x_0 x_1) \oplus x_2$$

$$T_R(x,y) = \left[y_0 \equiv \overline{x}_0 \right] \cdot \left[y_1 \equiv (x_0 \oplus x_1) \right] \cdot \left[y_2 \equiv ((x_0 x_1) \oplus x_2) \right]$$

$$\begin{aligned} T_R(x,y) = \ & \overline{x}_2 \overline{x}_1 \overline{x}_0 \ \overline{y}_2 \overline{y}_1 y_0 \ + \\ & \overline{x}_2 \overline{x}_1 x_0 \ \overline{y}_2 y_1 \overline{y}_0 \ + \\ & \overline{x}_2 x_1 \overline{x}_0 \ \overline{y}_2 y_1 y_0 \ + \\ & \overline{x}_2 x_1 x_0 \ y_2 \overline{y}_1 \overline{y}_0 \ + \\ & x_2 \overline{x}_1 \overline{x}_0 \ y_2 \overline{y}_1 y_0 \ + \\ & x_2 \overline{x}_1 x_0 \ y_2 y_1 \overline{y}_0 \ + \\ & x_2 x_1 \overline{x}_0 \ y_2 y_1 y_0 \ + \\ & x_2 x_1 x_0 \ \overline{y}_2 \overline{y}_1 \overline{y}_0 \end{aligned}$$

$(2,4) \in R$?

$p = \overline{x}_2 x_1 \overline{x}_0 \ y_2 \overline{y}_1 \overline{y}_0$ $\qquad p \cdot T_R = 1$?

Manipulation with characteristic functions

$$\text{odd}(x) = x_0$$
$$m3(x) = \overline{x}_0 \cdot \overline{x_1 \oplus x_2} + \overline{x}_2 x_1 x_0$$

(multiple of 3, odd) $\in R$?

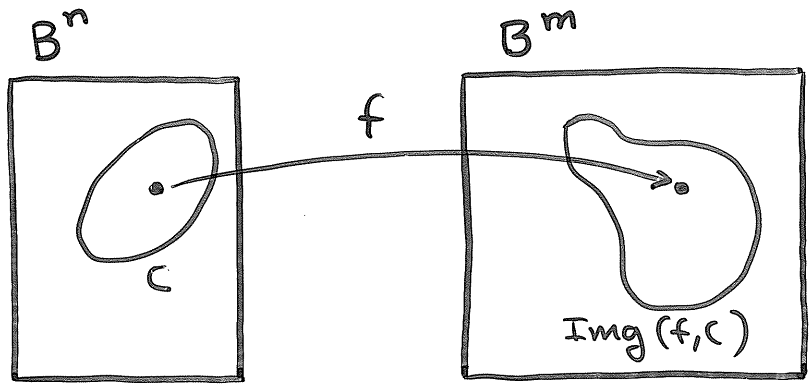
$$m3(x) \cdot \text{odd}(y) \cdot T_R(x, y)$$

$$\left(\overline{x}_0 \, \overline{x_1 \oplus x_2} + \overline{x}_2 x_1 x_0 \right) \cdot y_0 \cdot T_R(x, y) =$$

$$= \overline{x}_2 \overline{x}_1 \overline{x}_0 \; \overline{y}_2 \overline{y}_1 y_0 + x_2 x_1 \overline{x}_0 \; y_2 y_1 y_0$$

$(0, 1) \qquad (6, 7)$

Image computation with transition relations



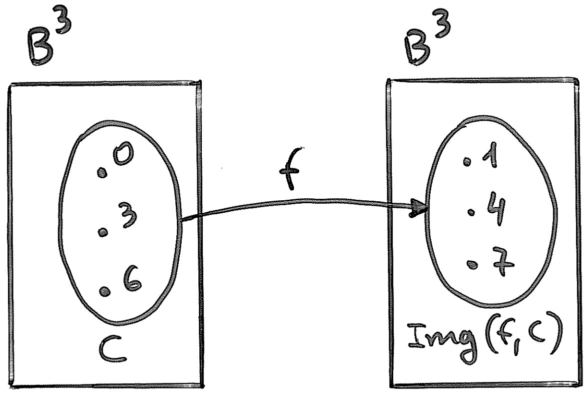
$$Y = \mathrm{Img}(f, C) = \exists_X \left[T_R(X, Y) \cdot C(X) \right]$$

Two steps:

1.- $P = T_R(X, Y) \cdot C(X)$
finds all pairs (x, y) such $C(X)$

2.- $Y = \exists_X P$
calculates the existential abstraction with respect to X

Image computation (example)



$$C(x) = \bar{x}_0 \, \overline{x_1 \oplus x_2} + \bar{x}_2 x_1 x_0$$

1 $P = T_R(x,y) \cdot C(x) =$

$$= \bar{x}_2 \bar{x}_1 \bar{x}_0 \bar{y}_2 \bar{y}_1 y_0 + \bar{x}_2 x_1 x_0 y_2 \bar{y}_1 \bar{y}_0 + x_2 x_1 \bar{x}_0 y_2 y_1 y_0$$

2 $\exists_x P = \bar{y}_2 \bar{y}_1 y_0 + y_2 \bar{y}_1 \bar{y}_0 + y_2 y_1 y_0$

Symbolic Breadth-First Traversal

symbolic_traversal (δ, s_0)

Reached = From = S_0 ;

<u>repeat</u>

To = Img (δ, From);

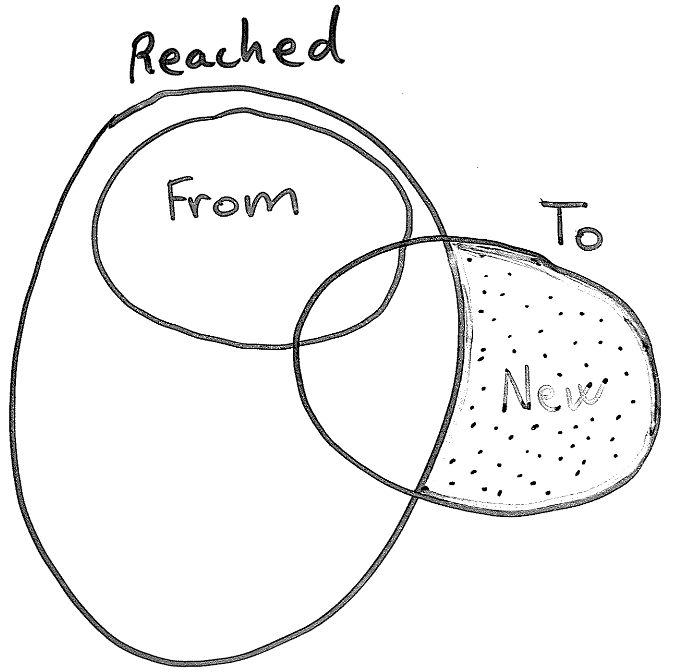
New = To − Reached ;

From = New ;

Reached = Reached $\cup$ New;

<u>until</u> New = $\emptyset$

return Reached



Symbolic Breadth-First Traversal

- Number of iterations : sequential depth of the FSM i.e. the longest of the shortest path connecting each state to an initial state

- Implementation

$$A \cup B \longrightarrow \chi_A + \chi_B$$

$$A \cap B \longrightarrow \chi_A \cdot \chi_B$$

$$A - B \longrightarrow \chi_A \cdot \overline{\chi_B}$$

χ_A is the characteristic function of A

- Efficient manipulation of characteristic functions by representing them with BDDs

Backward Traversal

backward_traversal (δ, S^0, P)

/* P is the property to be verified */

Reached = From = $\overline{P}$;

repeat

To = Img (δ^{-1}, From)

New = To − Reached

if $S^0 \in$ New **return** FALSE

From = New

Reached = Reached $\cup$ New

until New = $\emptyset$

return TRUE

Transition Relation for δ^{-1} (Preimage)

$$TR^{-1}(X, Y) = TR(Y, X)$$

Results

circuit	# latches	# size	# states	depth	CPU (sec)
sand	6	1310	32	4	16.9
scf	8	2208	115	16	18.7
s344	15	269	2625	7	34.6
s444	21	352	8865	151	186.7
s526	21	445	8868	151	126.1
s713	19	591	1544	7	63.7
s953	29	766	504	11	70.2
s1238	18	1042	2616	3	43.6
cbp.32.4	32	480	4.29e+09	2	14.1
minmax32	96	1874	1.32e+28	4	444.4
sbc	28	1670	154593	10	2903.7
key	228	3865	1.35e+68	17	5706.2

(from Touati et al., ICCAD-90)

Conclusions

- Formal verification is crucial for the design of complex systems

- Many formal verification techniques suffer the state explosion problem

- Symbolic traversal enables to perform reachability analysis with succint representations of sets of states

- Much effort has been focused to control circuits (FSM models)

- On-going research for data-path circuits