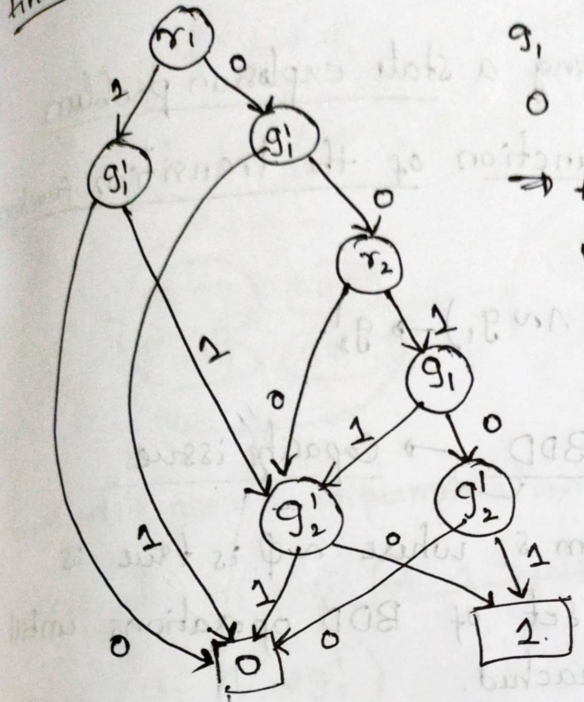
Final BDD

16 → valid paths



g_1	g_2	r_1	r_2	g_1'	g_2'
0	X	0	1	0	1

⇒ All paths from root to the node 1 are the valid ~~function~~ transition.

B_7

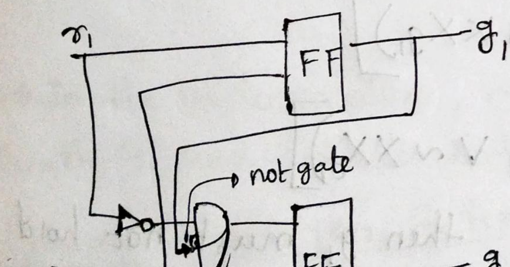
17/10/2023

18/10/2023

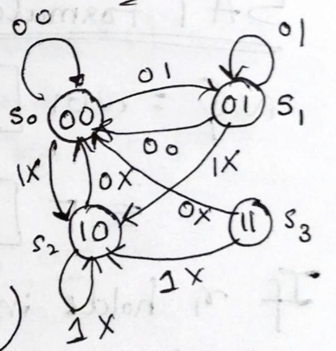
Absent

31/10/2023

SAT based forward property Verification



Kripke structure (transition system) ⟶



states $\langle g_1, g_2 \rangle$

transition $\langle r_1, r_2 \rangle$

open system

↓ convert to

closed system

$$\varphi :- G\,[r_1 \rightarrow Xg_1 \wedge XXg_1]$$

$$M, s_1 \models \varphi\ ?$$

→ Take $\sim\varphi$, and check for a path from S_1 where $\sim\varphi$ is true

→ This may end up having a state explosion problem

→ Take characteristic function of the transition function

$$\left\{\begin{array}{l} C1: \; r_1 \rightarrow g_1' \\ C2: \; (r_2 \wedge \sim r_1 \wedge \sim g_1) \rightarrow g_2' \end{array}\right.$$

→ Construct ROBDD → Capacity issue.

→ Search for path from S_1 where $\sim\varphi$ is true is done effectively by set of BDD operations untill a fixed point is reached.

→ $S_1 S_2 S_0$ path doesn't satisfy φ → $M, S_1 \not\models \varphi$

→ Due to Capacity issue, this method may not work as expected.

SAT Formulation

$$\sim\varphi = \sim G\left[r_1 \rightarrow (Xg_1 \wedge XXg_1)\right]$$

$$= F\left[r_1 \wedge (\sim Xg_1 \vee \sim XXg_1)\right]$$

If r_1 holds in a state, then g_1 must not hold atleast one of the next two states.

Set of variables:

$\bigcup_i (S_0^i, S_1^i, \ldots S_k^i)$ | Example (g_1, g_2, r_1, r_2)

Initial state 'I': $(\sim g_1^0 \wedge g_2^0)$

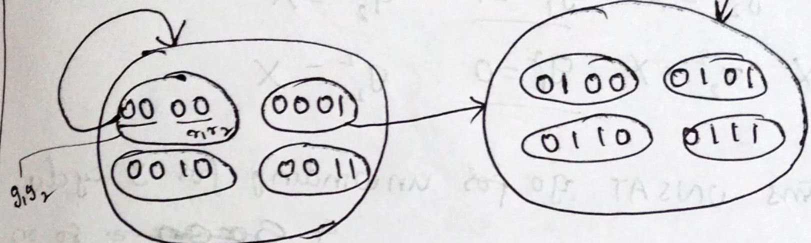
Possible transition from initial state (use characteristic function)

$C_1^1 : r_1^0 \rightarrow g_1^1$

sir wrote $\sim r_2^0$

$C_2^1 : (r_2^0 \wedge \sim g_1^0 \wedge \sim r_1^0) \rightarrow g_2^1$

initial state.



→ Unwind implementation for 1 cycle.

Unwind of $\sim\phi$ for 1 cycle

$z^1 : (r_1^0 \wedge \sim g_1^1)$

⇒ Check SAT of $(I \wedge z_1^1 \wedge C_1^1 \wedge C_2^1)$

means

⇒ check if $\sim\phi$ holds in the implementation upto 1 cycle from the initial state

⇒ ~~returns~~ returns UNSAT (because z^1, and C_1^1 contradicts) with each other

⇒ In the implementation, $\sim\phi$ does not hold in path length '1' from initial state.

Unwind transition System for two cycles

$C_1^2 : r_1^1 \rightarrow g_1^2$

$C_2^2 : (r_2^1 \wedge \sim g_1^1 \wedge \sim r_1^1) \rightarrow g_2^2$

sir wrote $\sim g_2^1$

Unwind $\sim\phi$ for two cycles

$z^2 : (r_1^0 \wedge (\sim g_1^1 \vee \sim g_1^2)) \vee (r_1^1 \wedge \sim g_1^2)$

⇒ check SAT $(I \wedge C_1^1 \wedge C_2^1 \wedge C_1^2 \wedge C_2^2 \wedge Z^2)$

⇒ return SAT

$$\begin{cases} r_1^0 = 1 & r_2^0 = X & g_1^0 = X & g_2^0 = X \\ r_1^1 = 0 & r_2^1 = X & g_1^1 = 1 & g_2^1 = X \\ r_1^2 = X & r_2^2 = X & g_1^2 = 0 & g_2^2 = X \end{cases}$$

⇒ If it returns UNSAT go for unwinding for 3 cycles & so on...

Bounded Model checking using SAT

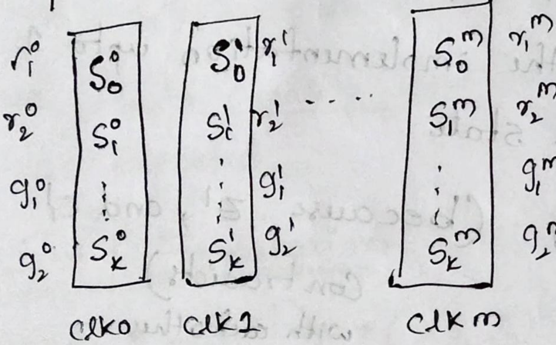
01/11/2023

Implementation (Characteristic func^n) R

Ex:- $\sim r_1 \wedge \sim g_1 \wedge r_2 \to g_2$

Property φ

$\sim\varphi$



~~clk~~ 'k' state variables

{ m is bound

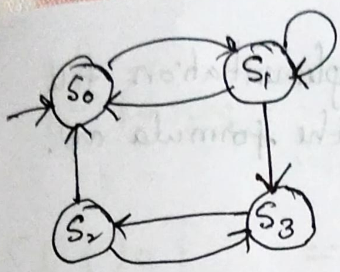
→ $k \times m$ variables in the SAT formula.

Clauses (formulas)

1) I :- that captures the initial state

2) C^j :- unfolding the state machine over time and generating new set of clauses

$$C^j = \bigwedge_{i=0,\, j=1} R(S^i, S^{i+1})$$



$C^0 = \{S_0\}$

$C^1 = \{S_0, S_1\}$

$C^2 = \{S_0, S_1, S_3\}$

$C^3 = \{S_0, S_1, S_2, S_3\}$

⇒ $R(S^i, S^{i+1})$ is state transition from set of states S^i to set of states S^{i+1}

3) Z^j :- Unfolding $\sim\varphi$ till jth clk to generate clauses.

$X g$ ⟶ g^1

$F g$ $\xrightarrow[\text{till } j\text{th}]{\text{unfolding}}$ $g^1 \vee g^2$ ~~not seen~~

⟶ $g^1 \vee g^2$

Bounded Model checking (R, φ, m) — bound

(Check transition 'R' holds φ in m clock cycles)

```
{
    j = 1
    while (j ≤ m) {
        1. Construct I, C^j, Z^j ;
        2. check SAT(I ∧ C^j ∧ Z^j);
           if (SAT returns satisfiable) {
               R doesnot satisfy φ, SAT instance
               is the counter example.
           }
        j++;
    }
    φ satisfies in R within bound of m ;
}
```

$\Rightarrow I \wedge C^i \wedge Z^i$: true $\Rightarrow$ Run on implementation till jth clk satisfy the formula $\sim\varphi$.

~~Unfolding properties~~

Unfolding state machines :-

till first clk

$$\begin{cases} \sim g_1 \wedge \sim r_1 \wedge r_2 \rightarrow g_2 \\ r_1 \rightarrow g_1 \end{cases}$$

$$C^0 = \begin{cases} \sim r_1^0 \wedge \sim g_1^0 \wedge r_2^0 \rightarrow g_2^1 \\ r_1^0 \rightarrow g_1^1 \end{cases}$$

till 2nd clk

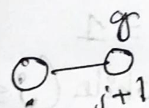
$$C^0 \wedge C^1$$

$$C^1 \begin{cases} \sim r_1^1 \wedge \sim g_1^1 \wedge r_2^1 \rightarrow g_2^2 \\ r_1^1 \rightarrow g_1^2 \end{cases}$$

Unfolding properties :-

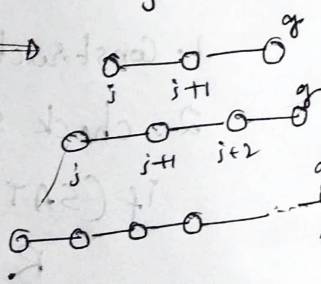
$[F]_{j,m}$ = set of clauses to be considered inorder to determine whether a property F is true at j^{th} clk, where $j < m$

$$[X_F]_{j,m} = (j<m) \wedge [F]_{j+1,m}$$

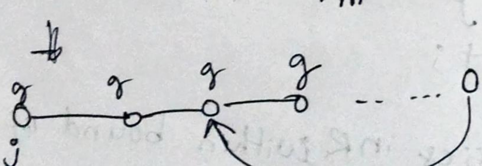


$$[F_g]_{j,m} = \bigvee_{i=j \ldots m} [g]_{i,m}$$

g holds in future



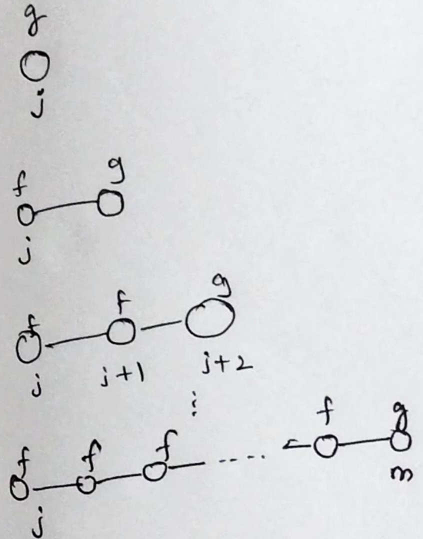
$$[G_f]_{j,m} = \bigwedge_{i=j,\ldots,m} [f]_{i,m} \wedge \text{loop}_m$$



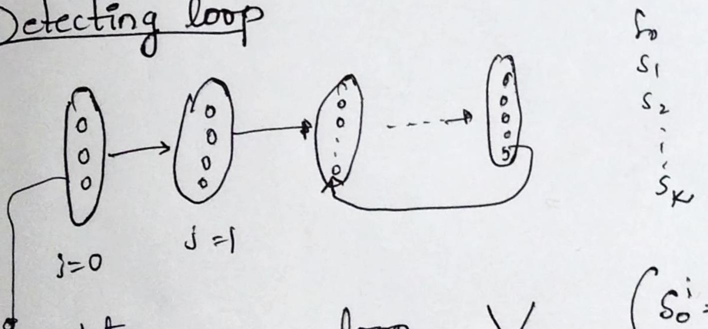
Check for loop also.

$$[G f]_{j,m} = \bigwedge_{i=j,\dots,m} [f]_{i,m} \wedge loop_m$$

$$[f \, U g]_{j,m} = \bigvee_{i=j,\dots,m} \left([g]_{i,m} \wedge \bigwedge_{n=j,\dots,i-1} [f]_{n,m} \right)$$



Detecting loop



set of states
in clk j
that are reachable
from j-1
if j=0 ⇒ initial state

$$loop_{0,1} = \bigvee_{j=0,\, i=1} \left(s_0^i = s_0^j \wedge s_1^i = s_1^j \wedge \dots \wedge s_k^i = s_k^j \right)$$