02/17/19 HOMEWORK #02

Saqib Khan
Sak2454

1) $S(a,b,c) = a \oplus b \oplus c$

$\Rightarrow (ab' + a'b) \oplus c$

$\Rightarrow (ab' + a'b)'c + (ab' + a'b)c'$

$\Rightarrow (ab')' \cdot (a'b)' \cdot c + ab'c' + a'bc'$

$\Rightarrow (a' + b)(a + b') + c + ab'c' + a'bc'$

$\Rightarrow abc + ab'c' + a'bc' + a'b'c$

$C(a,b,c) = ab + ac + bc$

a) $\forall a : S(a,b,c) \Rightarrow f|_{a=0} \cdot f|_{a=1}$

$\Rightarrow (bc' + b'c) \cdot (bc + b'c')$

$\Rightarrow \cancel{bc'c} + \cancel{bb'c'} + \cancel{bb'c} + \cancel{b'cc'}$

$\boxed{\forall a : S(a,b,c) \Rightarrow 0}$

b) $\exists a : S(a,b,c) = f|_{a=0} + f|_{a=1}$

$= (bc' + b'c) + (bc + b'c')$

$= b\underbrace{(c + c')}_{1} + b'\underbrace{(c + c')}_{1}$

$= b + b'$

$\boxed{\exists a \; S(a,b,c) = 1}$

c) $\forall a: C(a,b,c) = f|_{a=0} \cdot f|_{a=1}$

$$= bc \cdot (b+c)$$
$$= bc + bc$$

$$\boxed{\forall a: C(a,b,c) = bc}$$

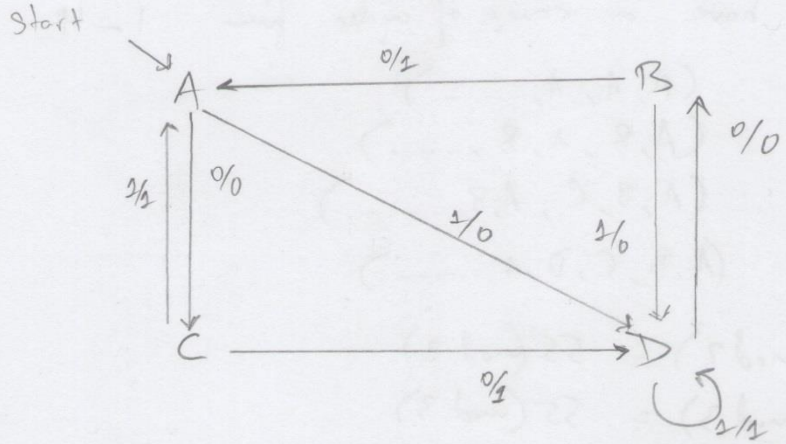
d) $\exists a: C(a,b,c) = f|_{a=0} + f|_{a=1}$

$$= bc + b + c$$
$$= b\underbrace{(c+1)}_{1} + c$$

$$\boxed{\exists a: C(a,b,c) = b + c}$$

2 X: 0 0 1 1 0 0 1 1 0 1 1 0 0 0 1

Z: 0 1 1 1 0 1 0 1 0 0 1 0 1 0 1



This description is not unique because the fsm can possible start at either A, B, C or D. Also, instead of going from A → C, we could easily have gone to B instead.

Since no two states are the same (i.e inputs + outputs), we can't reduce this to a 3-state machine

3 Since the FSM has 4 states, the application is bound to repeat a state after 4 transitions. More specifically if we have consecutive 0's, then we can have a range of cycles from 1–4

cycle 1 :- (A, A, A, ……)

cycle 2 :- (A, B, A, B, ……)

cycle 3 :- (A, B, C, A, B, ……)

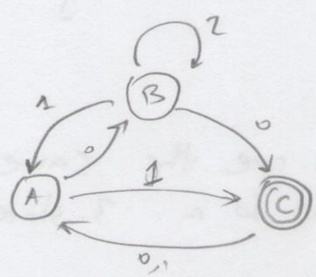
cycle 4 :- (A, B, C, D, A, ……)

Since

$$7 \pmod 2 = 55 \pmod 2$$
$$7 \pmod 3 = 55 \pmod 3$$
$$7 \pmod 4 = 55 \pmod 4$$

Therefore the FSM will be in the same state after 7 or 55 transitions.

4)



Possible states :

AC

ABC

ABB…C

ABABC

ABAC

$$\therefore \underbrace{\left(1(0+1) + 02^*0(0+1) + 02^*1\right)^*}_{\text{STAY IN A}} \left(\underset{\uparrow \; AC}{1} + \underbrace{02^*0}_{ABC}\right)$$

5

a) No, can't be recognized by FSM. It needs infinite # of states to identify the set.

b) Yes, can be recognized by FSM using 5 states

c) Yes, since k is a specified number

d) No, because it needs infinite # of states to recognize

e) Yes

f) Yes, if length == 7, then there are $2^7 = 128$ sequences that can justify this FSM.

6 a) $(011^* + 01^* 0)^*$ and $(00)^* 01 (0+1)^* + X$

Not Equal

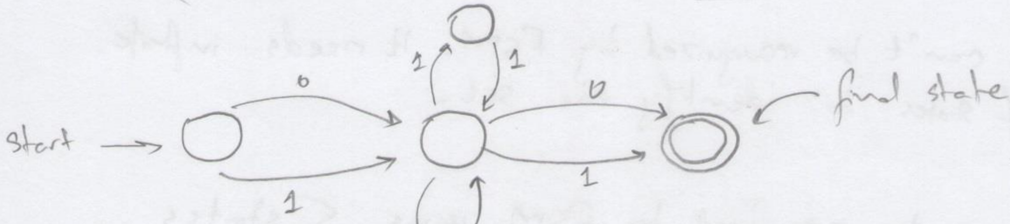
b) $(1^* 0 + 001)^* 01$ and $(1^* 001 + 00101)^*$

Not Equal

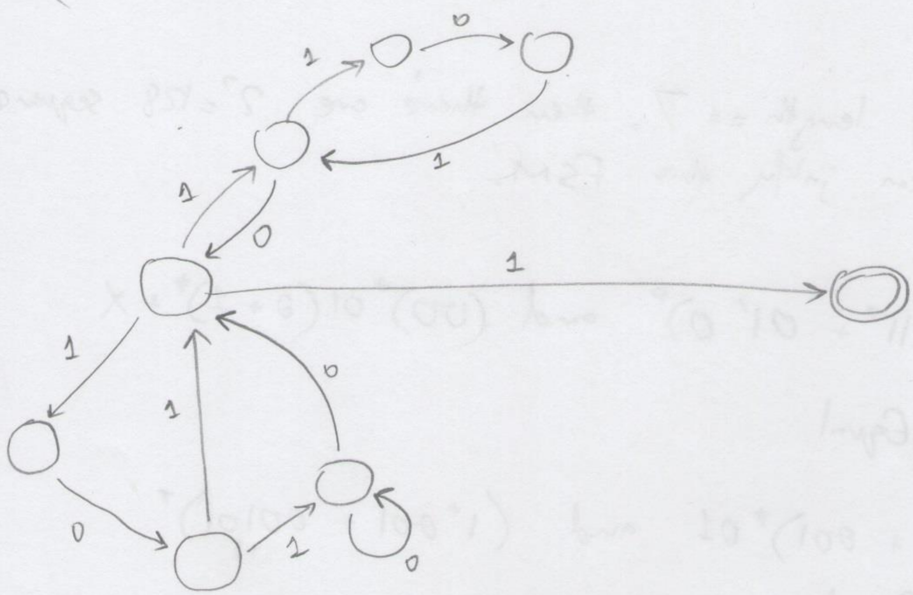
c) $0^* 1 (0 + 10^* 1)^*$ and $(1 + 00^* 1) + (1 + 00^* 1)(0 + 10^* 1)^* (0 + 10^* 1)$

Equal

7) a) $(0+1)(11+0^*)^*(0+1)$



b) $(1010^* + 1(101)^*0)^*1$



8 a) This technique is not scalable

b)
```
assert property ( @(posedge clk) disable iff (!rst_n)
    !(grant0 && grant1));
```

9) p S_until q

$$\underbrace{p \to p \to p \to p \to q}_{\text{Assertion holds in these states}} \to \bigcirc$$

10) \$rose (expression) :- true if the least significant bit of the expression changed to 1; false otherwise

@posedge :- it is an event that is triggered on the positive edge of the clock (or any expression) and has no returns.

11)

a) property p_start;

@(posedge clk) $rose (sel_bit) |-> ##1 trdy ##1 !trdy;

endproperty

b) property p_stop;

@(posedge clk) $fell (sel_bit) |=> trdy