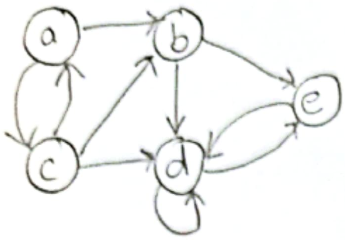
R10943157 Arvind Singh Rathore

Q1.)




a) $AGF(a \to Xd) \equiv AGF(Xd \vee \neg a)$

$\Rightarrow$ For every path, $F(Xd \vee \neg a)$ holds globally in the future

$\Rightarrow$ For every path $(Xd \vee \neg a)$ holds sometime globally in the future

Case I] When node = a

$AGF(Xd)$ holds because d is reachable from a to e
viz $b \to d$, $c \to d$, $d \to d$, $e \to d$, $a \to b \to d$

Case II] When node $\neq$ a

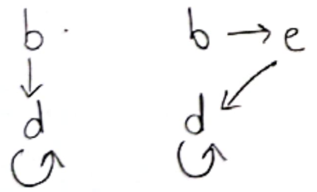
$Xd \vee \neg a$, irrespective of $Xd$, this is always true

$\therefore$ $AGF(a \to Xd)$ is correct

b] $EG(b \to AFd) \equiv EG(AFd \vee \neg b)$

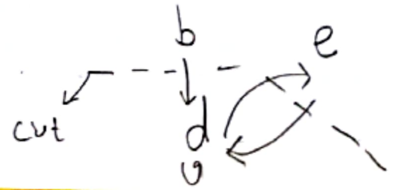
In simpler terms, this means there exists a path globally where $(AFd \vee \neg b)$ holds.

Case I.] When node = b, then $EG(AFd)$
From b, there exists a path where AFd holds globally in future.
This is true because at some point of time d will eventually hold as from b, you compulsarily will have to go through d for all possible paths.




Alternatively, we can think this to be

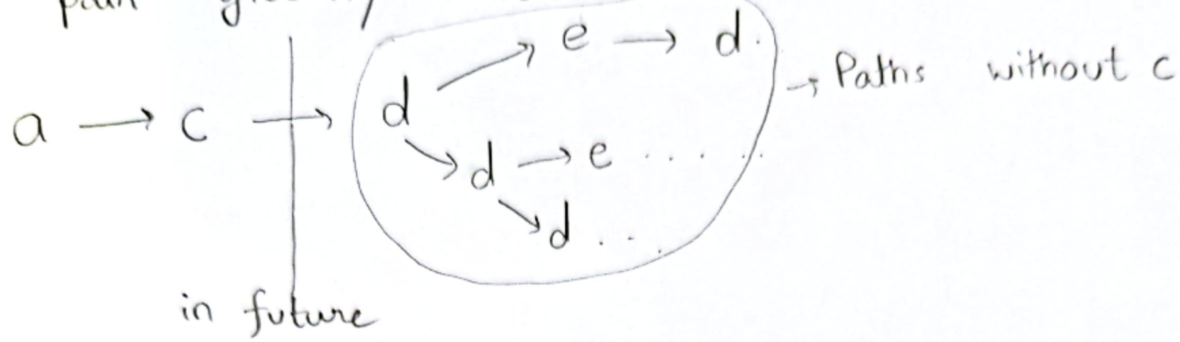
$EG(AFd) \Rightarrow$ There exist path for all nodes where there is cut to d




$\therefore$ $EG(b \to AFd)$ holds.

c.) EFAG!c

In simple terms, there exists a path in future where for every path globally there's no c




∴ EFAG!c holds

Q2 - a.] A simple method instead of keeping a track on all paths is to count number of bubbles, since all paths lead to one

i) a, b, c, d, e

(0, 0, 0, 0, 0) ⇒ # bubbles = 4  ∴ $f(0,0,0,0,0) = 1$

(ii) a, b, c, d, e

(1, 0, 1, 0, 1) ⇒ # bubbles = 3  ∴ $f(1,0,1,0,1) = 0$

(iii) a, b, c, d, e

(0, 1, 0, 1, 0) ⇒ # bubbles = 3  ∴ $f(0,1,0,1,0) = 0$

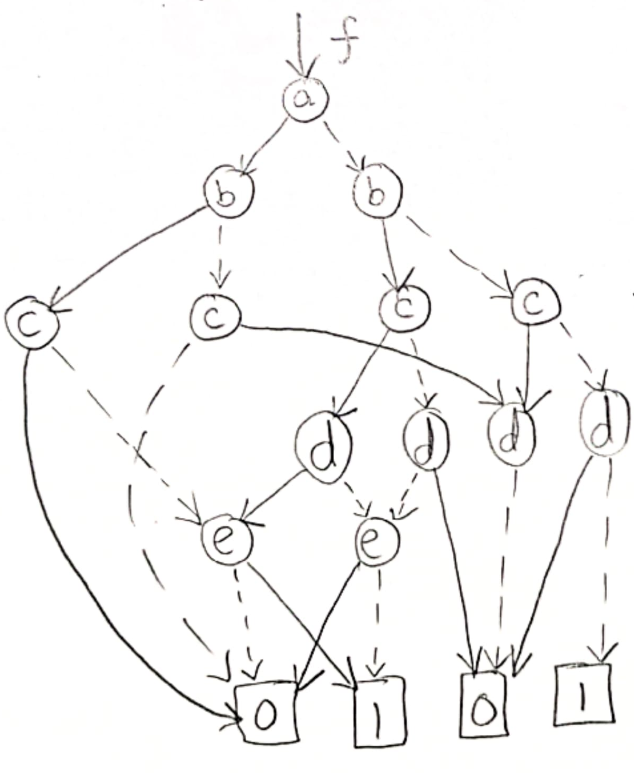
Q2-b.] List of {a, b, c, de} for f = 0

0001X, 0010X, 01001, 0101X, 01101, 01110, 100XX,

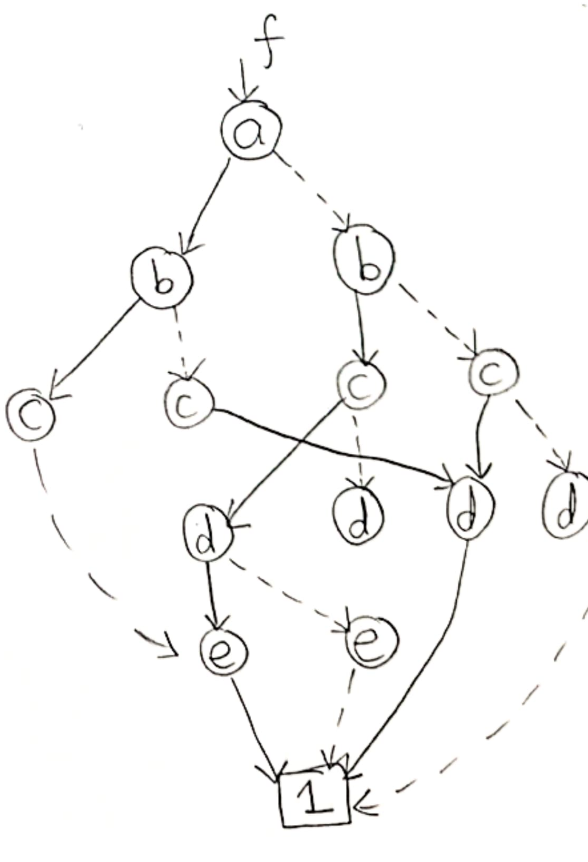
1010X, 110X0, 111XX

Total # = 21

Q2- c]

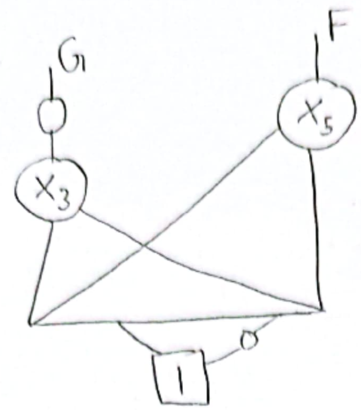



⇒



Q3.]

2 BDD nodes




From Page 50 of slide 3

→ Rule 1 is fine

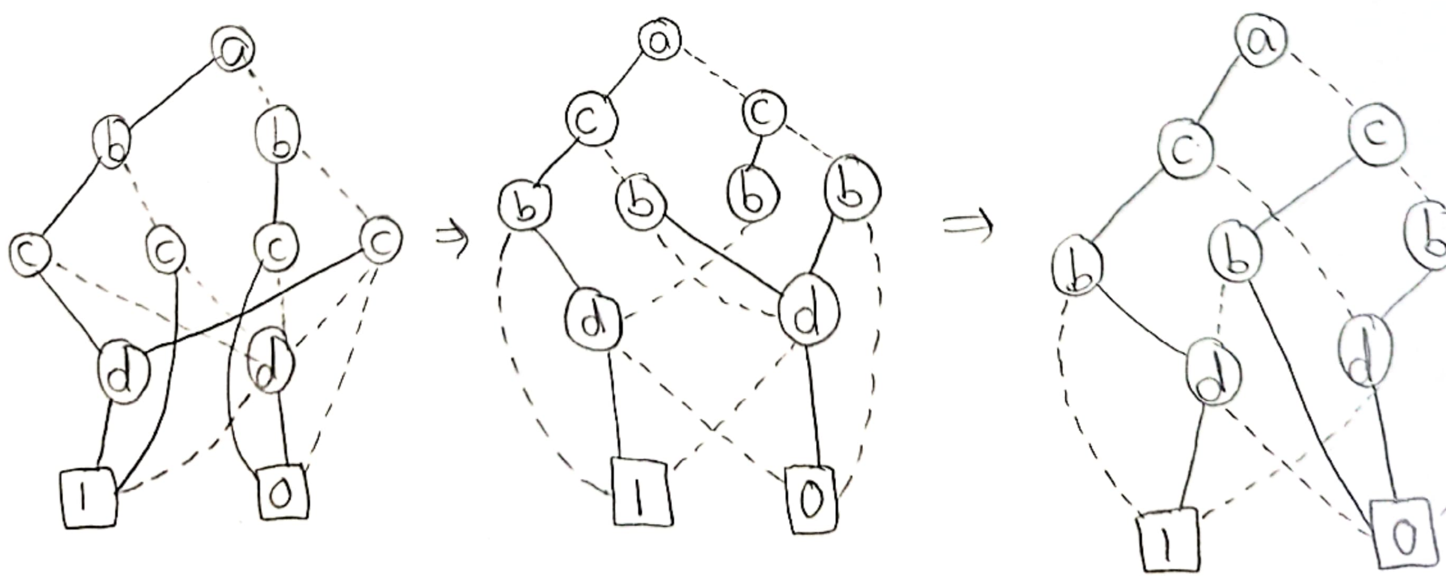
→ Rule 2 ITE(F, 0, G), since $x_5 > x_3$

$ITE(F, 0, G) \rightarrow ITE(\bar{G}, 0, \bar{F})$

→ Rule 3 : If contains complement edge parameters :
The first & second parameters can't be complement edges.

$\therefore ITE(\bar{G}, 0, \bar{F}) \longrightarrow \overline{ITE(\bar{G}, 1, F)}$

Q4.]

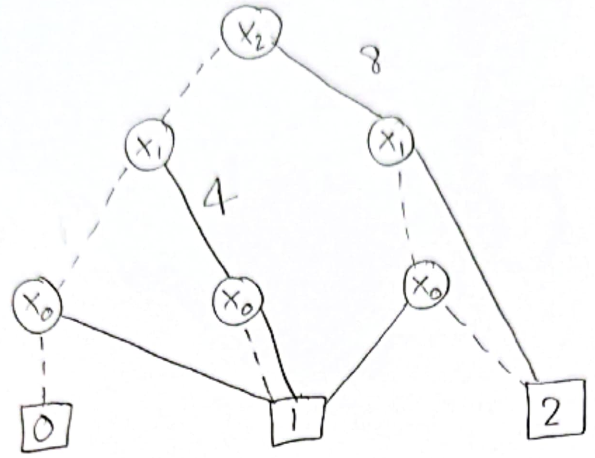


Q5.] a) For 3 bit vector $x_2 x_1 x_0$ $(x)$

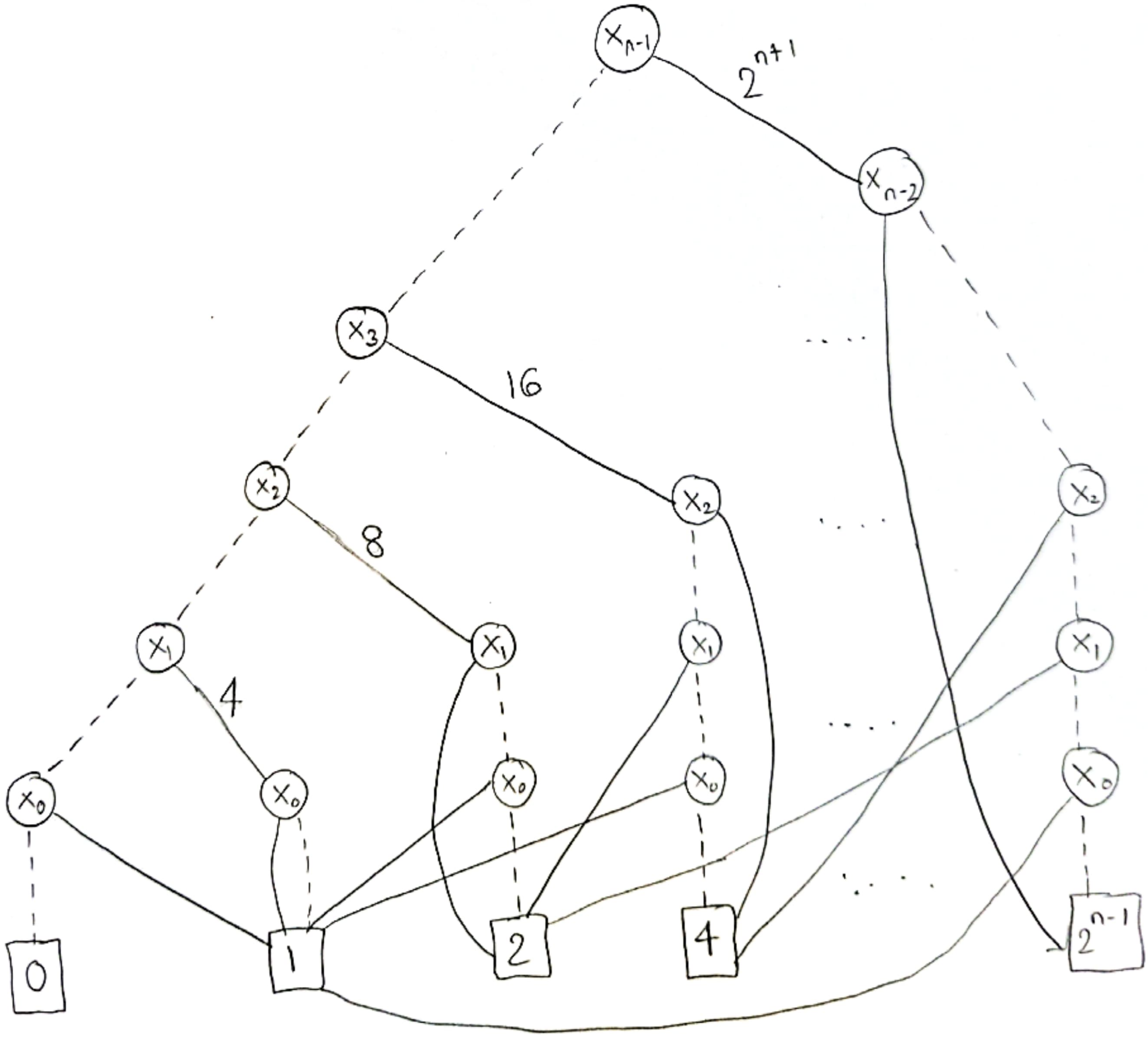
$$x^2 = (4x_2 + 2x_1 + x_0)(4x_2 + 2x_1 + x_0)$$

$$x^2 = 16x_2^2 + 16x_1x_2 + 8x_0x_2 + 4x_0x_1 + 4x_1^2 + x_0^2$$

& the corresponding *BMD is




For a general n bit case



Q5.) b.] Since we are interested in finding the square root (integral part), we need to think of a way to convert the *BMD into a form which describes the truth table of the desired function.

MTBDD best serves this purpose as it models the function according to the truth table with word values.

So the algorithm: $Y \rightarrow$ input value

1.) From the *BMD of $X^2$ in part 5(a), convert it to BMD

2.) Convert the BMD to MTBDD

3.) Once we get the MTBDD, it's leaf nodes will consist of square values.

4.) Check Y for values on leaves.

```
for i in range (number of leaves):
  if (Y > leaf (i))
    go to next leaf  // i++
  if (Y <= leaf (i))
    go back to previous leaf
    return square root of this leaf value
```

We can also use efficient searching algorithms on trees like BST to reduce the computational complexity