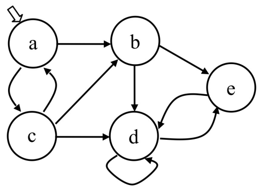
1. Given the Kripke structure below, where ‘a’, ‘b’, ‘c’, ‘d’ and ‘e’ are atomic propositions and the initial state is denoted with an arrow. Please verify the correctness of the following temporal formulae by the explicit modeling checking technique. Explain why.

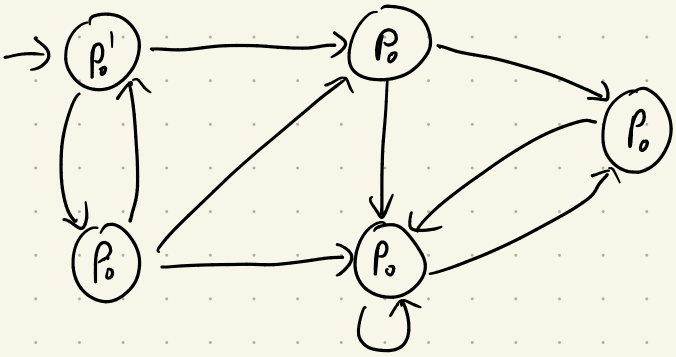
(a) AGF (a → X d)

(b) EG (b → AF d)

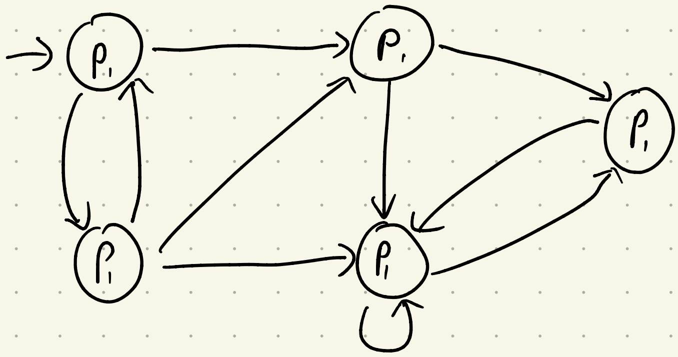
(c) EFAG !c



(a) $P_0 = a \to X d$ $\qquad$ $P_1 = F P_0$

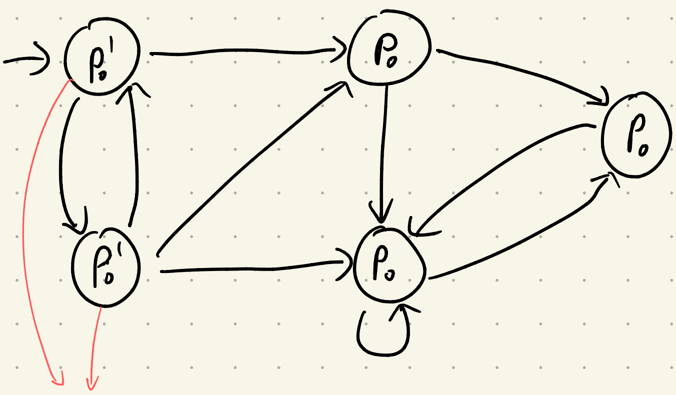


⇒



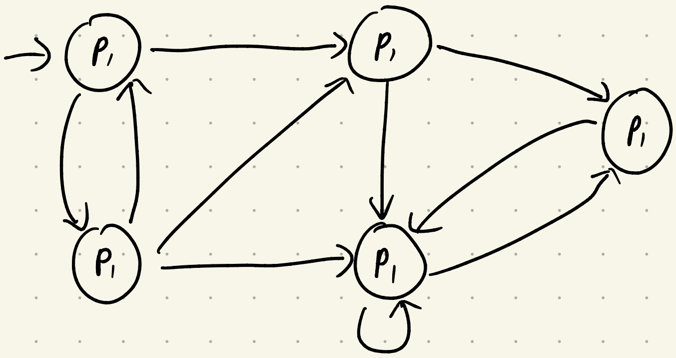
∵ P_1 holds in every states ⇒ AGP_1 = AGF(a→Xd) holds

(b) $P_0 = AFd$ $\qquad$ $P_1 = b \to P_0$



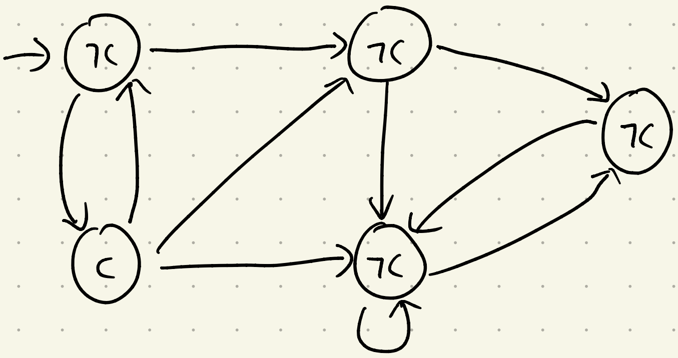
∵ ∃ p = a→c→a→c ···

⇒

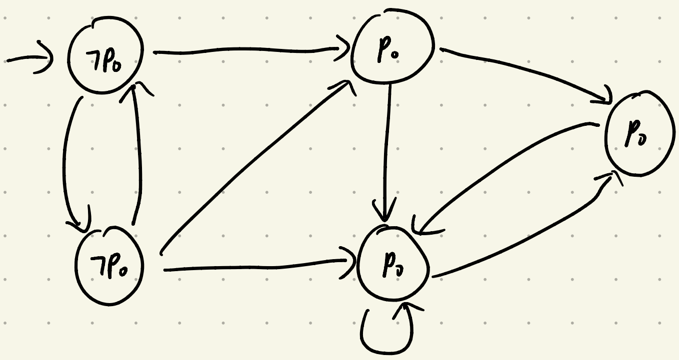


∵ P_1 holds in every states ⇒ EG(b→AFd) holds (ex: a→c→a→c ···)

(c) $P_0 = AG!C$



⇒



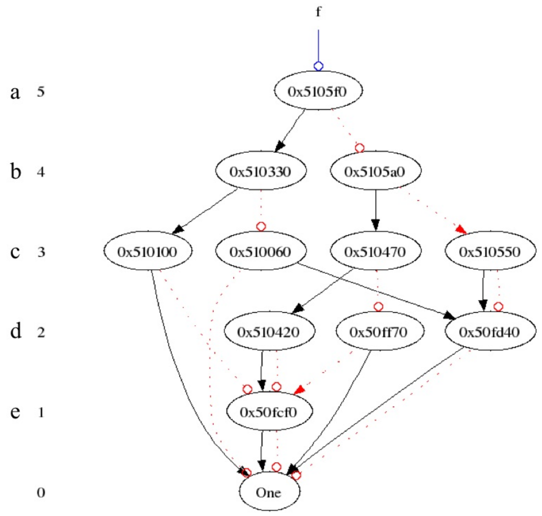
∵ ∃ state that P_0 holds ⇒ EFP_0 = EFAG!C holds (ex: a→b→d→d→···)

2. Given the following BDD *f* ---

(a) What are the values of *f* for (a, b, c, d, e) = (0, 0, 1, 1, 0), (1, 0, 1, 0, 1), and (1, 1, 0, 0, 0) ?

(b) List all the cubes for "*f* = 0" in terms of input variables { a, b, c, d, e } (e.g. "10xxx"). Please list the cubes in ascending order (e.g. 00000, 00001, 0000x)

(c) Covert this BDD *f* to a BDD without complemented edges. Just draw the BDD nodes with input labels { a, b, c, d, e }, not the pointer addresses (e.g. 0x5105f0), and you can ignore the edges to constant '0', such as the BDDs in p18, lecture note #3.



(a)

(0,0,1,1,0):

$(0x5105f0)' \xrightarrow{0} 0x5105a0 \xrightarrow{0} 0x510550 \xrightarrow{1} 0x50fd40 \xrightarrow{1} 1$

(1,0,1,0,1):

$(0x5105f0)' \xrightarrow{1} (0x510330)' \xrightarrow{0} 0x510060 \xrightarrow{1} 0x50fd40 \xrightarrow{0} (1)' = 0$

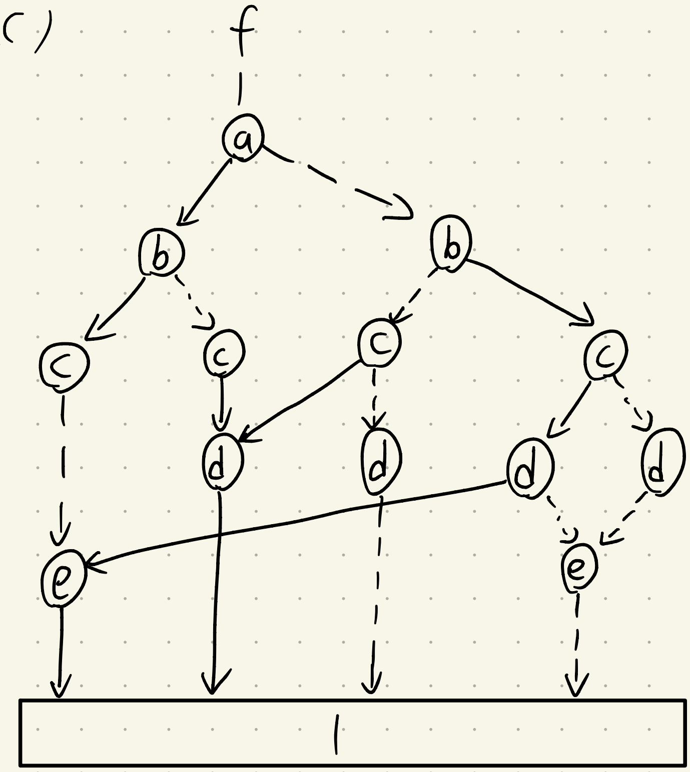
(1,1,0,0,0):

$(0x5105f0)' \xrightarrow{1} (0x510330)' \xrightarrow{1} (0x510100)' \xrightarrow{0} 0x50fcf0 \xrightarrow{0} (1)' = 0$

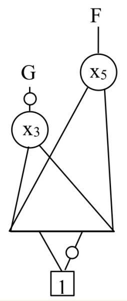
(b) f = 0:

abcde
0001x
0010x
01001
0101x
01101
01110
100xx
1010x
110x0
111xx

(c)



3. Let F, G be two BDD nodes, where G has the smaller (i.e. lower) top variable index than F. In addition, G has complemented edge while F has not (as shown below). Let R = ITE(F, 0, G). Please use the rules in the lecture notes to standardize this ITE call for the entry of the computed cache.



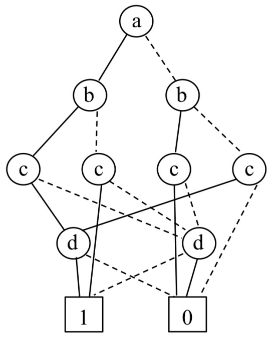
note that both 0 and G has complement edge

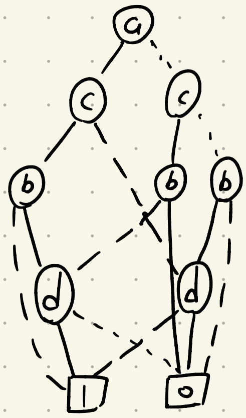
$R = ITE(F, 0, G)$

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

$= \overline{ITE(\bar{G}, 1, F)}$

4. Given the BDD below, perform dynamic variable reordering for 'b' and 'c' and draw the final BDD. What is the difference in the number of nodes?





⇒ number of nodes decreased by 1.

5. Let X be an n-bit bit-vector <$x_{n-1}, x_{n-2}, \ldots, x_1, x_0$> where x_{n-1} and x_0 are the most and the least significant bits, respectively.

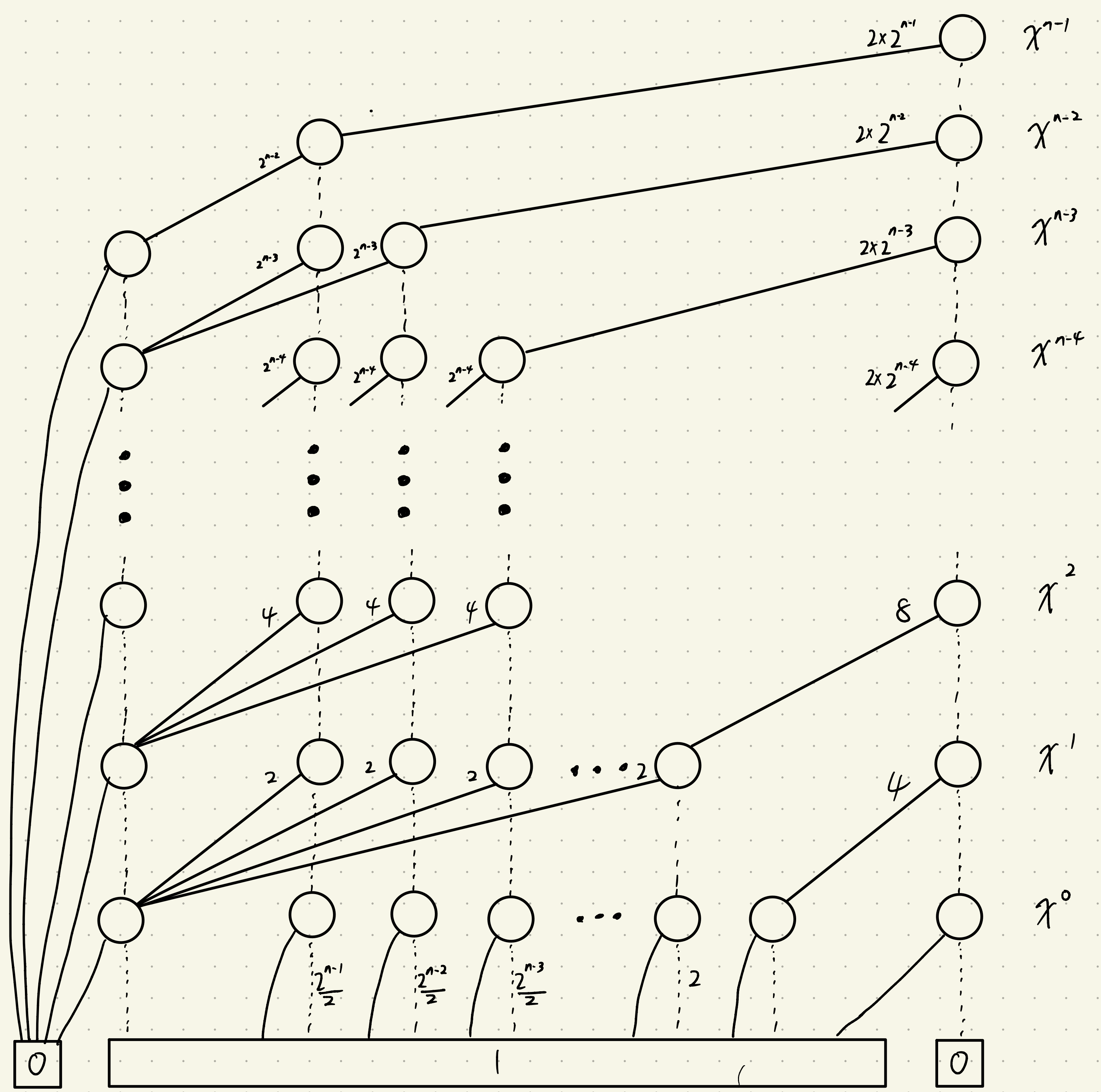
(a) Represent the square of X (i.e. X^2) in *BMD with the variable order: { $x_{n-1}, x_{n-2}, \ldots, x_1, x_0$ }, where x_{n-1} and x_0 are the top and the bottom variables, respectively.

(b) Design an algorithm (with pseudo code) to derive the integral part of the square root of a positive integer Y using the *BMD representation in sub-problem (a). For example, the integral part of the square root of 10 is 3, and that of 20 is 4, respectively. Discuss the computational complexity of your algorithm. Please make the algorithm as efficient as possible.

$$X = 2^0 x_0 + 2^1 x_1 + \cdots + 2^{n-1} x_{n-1}$$

$$\Rightarrow X^2 = \sum_{i=0}^{n-1} 2^{2i} x_i^2 + \sum_{j=i+1}^{n-1} \sum_{i=0}^{n-1} 2^{i+j+1} x_i x_j$$

(a)



Square Root $(Y = \langle y_{n-1}, y_{n-2} \cdots y_1, y_0 \rangle)$

Construct a *BMD with $\lceil \frac{n}{2} \rceil$ variables $(x_{\lceil \frac{n}{2} \rceil -1}, x_{\lceil \frac{n}{2} \rceil -2} \cdots x_1, x_0)$

Set = { }

for i from $\lceil \frac{n}{2} \rceil$ to 1 **do**

  tmp = 0

  tmp = tmp + traverse *BMD with $(x_i = 1, x_j = 0$ for $i \neq j)$

  if tmp > Y **then**

    continue

  end if

  for each $j \in$ Set **do**

    tmp = tmp + traverse *BMD with $(x_i = 1, x_j = 1, x_k = 0$ for $k \neq i \wedge k \neq j)$ ($\because$ all the edge weight ≥ 1 in *BMD, when traversing from $x_{\lceil \frac{n}{2} \rceil -1}$, if the current value derived already $> (Y - \text{tmp})$, return the value in advance to reduce traversing time.)

    if tmp > Y **then**

      break

    end if

  if tmp $\leq$ Y **then**

    Set = Set $\cup$ {i}

    Y = Y − tmp

    if Y = 0 **then**

    return $\{X = \langle x_{\lceil \frac{n}{2} \rceil} \cdots x_0 \rangle \mid x_i = \begin{cases} 1 & \text{if } i \in \text{Set} \\ 0 & \text{else} \end{cases} \}$

    end if

  end if

  end for each

end for

return $\{X = \langle x_{\lceil \frac{n}{2} \rceil} \cdots x_0 \rangle \mid x_i = \begin{cases} 1 & \text{if } i \in \text{Set} \\ 0 & \text{else} \end{cases} \}$

time complexity : at worst traverse the *BMD $O(n^2)$ times

each traversal traverses $O(n)$ nodes

$\Rightarrow$ time complexity : $O(n^3)$