

# Discrete Event Systems

## Solution to Exercise Sheet 11

### 1 Sets Representation

During the design phase of a system, it is common to qualify some of the accessible states as *faulty*, or *error* states. It is precisely one of the goal of the design phase to ensure that the system won't reach such states, or at least only in expected ways. Let us identify several set of states:

- $X$ : the overall set of states,
- $N$ : the set of nominal states,
- $E$ : the set of error states,
- $O$ : the set of state where there is a memory overflow.

We denote by  $\psi_Q$  the characteristic function of the set  $Q$ , i.e.,  $x \in Q \Leftrightarrow \psi_Q(\sigma(x)) = 1$  where  $\sigma(x)$  is the binary encoding of the state  $x$ .

- "The nominal and error sets cover all the state space altogether". Express this property in term of sets and characteristic functions.
- "No overflow state can also be a nominal state". Express this property in term of sets and characteristic functions.
- Describe  $Q_1$ , the set of error states which are not an overflow, in term of sets and characteristic functions.
- Describe  $Q_2$ , satisfying " $O \Rightarrow E$ ", i.e., the set of state for which this property holds, in term of sets and characteristic functions.

$$\text{a) } N \cup E = X \quad \Leftrightarrow \quad \psi_N + \psi_E = 1$$

$$\text{b) } N \cap O = \emptyset \quad \Leftrightarrow \quad \psi_N \cdot \psi_O = 0$$

$$\text{c) } Q_1 = E \setminus O \quad \Leftrightarrow \quad \psi_{Q_1} = \psi_E \cdot \overline{\psi_O}$$

$$\begin{aligned} \text{d) } Q_2 = (O \cap E) \cup \overline{O} &= (O \cup \overline{O}) \cap (E \cup \overline{O}) & \Leftrightarrow & \quad \psi_{Q_2} = \psi_E + \overline{\psi_O} \\ &= X \cap (E \cup \overline{O}) \\ &= E \cup \overline{O} \end{aligned}$$

## 2 Binary Decision Diagrams

For an Ordered Binary Decision Diagram (OBDD), we denote by  $\Pi : x_1 < x_2 < \dots < x_n$  the variable order, where  $x_1$  is the highest variable of the tree,  $x_2$  the second highest, and so on. An ordering  $\Pi_1$  is said to be better than  $\Pi_2$  for an OBDD  $G$  if  $G$  contains less nodes when using  $\Pi_1$  rather than  $\Pi_2$  (eventually after merging equivalent nodes).

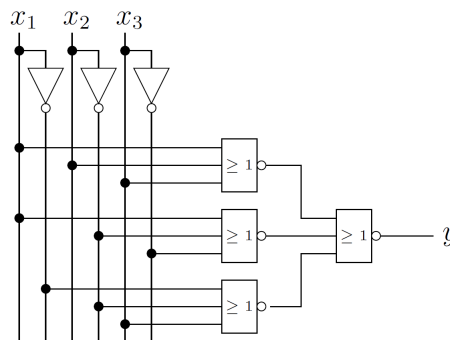
In the following, use the following notation to represent BDDs: A solid arc (——) if the variable labeling the parent node evaluates to 1, and the dashed arc (---) otherwise. **Do not** use color (it is a bad habit to take...).

### 2.1 Verification using BDDs

An engineer wants to implement the following function on a digital circuits

$$f_1 : (x_1\bar{x}_2 + x_1x_3 + \bar{x}_2x_3 + \bar{x}_1x_2\bar{x}_3)$$

Allowing solely inverters and NOR-gates, the synthesis program returns:



The old fashion team-leader does not trust these new fancy software so much, so he asks to verify this circuit does indeed behave like  $f_1$ . This can be done using BDDs.

- Express the function  $f_2$  realized by the circuit.
- Draw and compare the minimized OBDDs of  $f_1$  and  $f_2$  using the ordering of variables  $\Pi : x_1 < x_2 < x_3$ . Verify they express the same behavior.

a)  $f_2 : y = \overline{\overline{x_1 + x_2 + x_3} + \overline{x_1 + x_2 + x_3} + \overline{x_1 + x_2 + x_3}}$

b) For  $f_1$ , we have

Fall  $x_1 = 0$

$$y_{|x_1=0} = \bar{x}_2x_3 + x_2\bar{x}_3$$

Fall  $x_2 = 0$

$$y_{|x_1=0, x_2=0} = x_3$$

Fall  $x_2 = 1$

$$y_{|x_1=0, x_2=1} = \bar{x}_3$$

Fall  $x_1 = 1$

$$y_{|x_1=1} = \bar{x}_2 + x_3 + \bar{x}_2x_3$$

Fall  $x_2 = 0$

$$y_{|x_1=1, x_2=0} = 1$$

Fall  $x_2 = 1$

$$y_{|x_1=1, x_2=1} = x_3$$

For  $f_2$ , we have

Fall  $x_1 = 0$

$$y_{|x_1=0} = \overline{x_2 + x_3 + \bar{x}_2 + \bar{x}_3}$$

Fall  $x_2 = 0$

$$y_{|x_1=0, x_2=0} = \overline{\bar{x}_3 + \bar{1} + \bar{x}_3} = x_3$$

Fall  $x_2 = 1$

$$y_{|x_1=0, x_2=1} = \overline{\bar{1} + \bar{x}_3} = \bar{x}_3$$

Fall  $x_1 = 1$

$$y_{|x_1=1} = \overline{\bar{1} + \bar{1} + \bar{x}_2 + x_3} = \bar{x}_2 + x_3$$

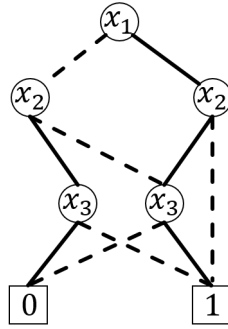
Fall  $x_2 = 0$

$$y_{|x_1=1, x_2=0} = 1$$

Fall  $x_2 = 1$

$$y_{|x_1=1, x_2=1} = x_3$$

Both BDDs have the same falls. They are then equivalent.

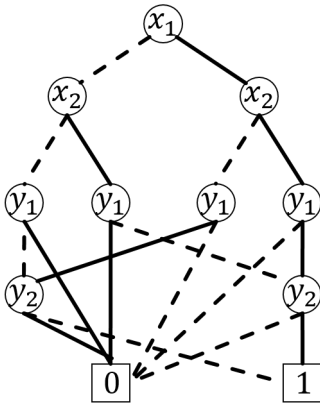


## 2.2 BDDs with respect to different orderings

- a) Consider the boolean function  $g(x_1, x_2, y_1, y_2) = (x_1 \equiv y_1) \cdot (x_2 \equiv y_2)$  and the ordering of variables  $\Pi : x_1 < x_2 < y_1 < y_2$ .  
Give the Boole-Shannon decomposition of  $g$  with respect to  $\Pi$ .
- b) Draw the corresponding OBDD for  $g$ .
- c) Let us now consider the new ordering  $\Pi' : x_1 < y_1 < x_2 < y_2$ . Use it to reconstruct the OBDD of  $g$ . Is  $\Pi'$  a better ordering than  $\Pi$  for  $g$ ?

a)  $g = x_1\{x_2[y_1(y_2) + \overline{y_1}(0)] + \overline{x_2}[y_1(\overline{y_2}) + \overline{y_1}(0)]\} + \overline{x_1}\{x_2[y_1(0) + \overline{y_1}(y_2)] + \overline{x_2}[y_1(0) + \overline{y_1}(\overline{y_2})]\}$

b)



- c) With the new ordering  $\Pi'$ , the Boole-Shannon decomposition becomes

$$g = x_1\{y_1[x_2(y_2) + \overline{x_2}(\overline{y_2})] + \overline{y_1}[0]\} + \overline{x_1}\{y_1[0] + \overline{y_1}[x_2(y_2) + \overline{x_2}(\overline{y_2})]\}$$

This is a better ordering as it leads to a OBDD with fewer nodes as with  $\Pi$  (6 instead of 9).

