ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and Networks Laboratory
Computer Engineering Group (TEC)

HS 2015 Prof. L. Thiele / R. Jacob

Discrete Event Systems

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 ψ_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.

a) "*The nominal and error sets cover all the state space altogether*". Express this property in term of sets and characteristic functions.

b) "*No overflow state can also be a nominal state*". Express this property in term of sets and characteristic functions.

c) Describe Q_1, the set of error states which are not an overflow, in term of sets and characteristic functions.

d) 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.

2 Binary Decision Diagrams

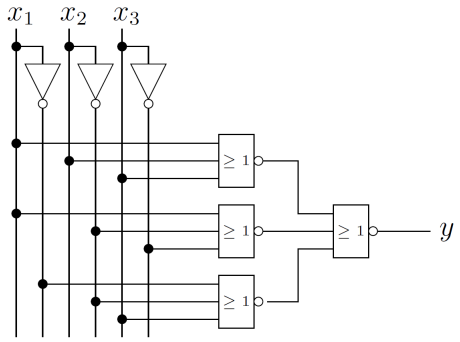
For an Ordered Binary Decision Diagram (OBDD), we denote by $\Pi : x_1 < x_2 < \ldots < x_n$ the variable order, where x_1 is the highest variable of the tree, x_2 the second highest, and so on. An ordering Π_1 is said to be better than Π_2 for an OBDD G if G contains less nodes when using Π_1 rather than Π_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

$$f_1 : (x_1\overline{x_2} + x_1x_3 + \overline{x_2}x_3 + \overline{x_1}x_2\overline{x_3})$$

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



The old fashion team-leader does trust these new fancy software so much, so he asks to verify this circuits does indeed behave like f_1. This can be done using BDDs.

a) Express the function f_2 realized by the circuit.

b) Draw and compare the minimized ODBBs of f_1 and f_2 using the ordering of variables $\Pi : \ x_1 < x_2 < x_3$. Verify they express the same behavior.

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 Π.

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 Π' a better ordering than Π for g?