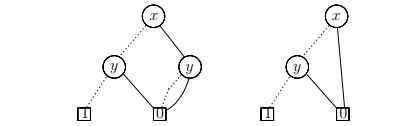
**Binary decision diagrams**

Binary decision diagrams (BDDs) are another way of representing boolean functions. A certain class of such diagrams will provide the implementational framework for our symbolic model-checking algorithm. Binary decision diagrams were first considered in a simpler form called binary decision trees. These are trees whose non-terminal nodes are labelled with boolean variables x, y, z,... and whose terminal nodes are labelled with either 0 or 1. Each non-terminal node has two edges, one dashed line and one solid line.

Let T be a finite binary decision tree. Then T determines a unique boolean function of the variables in non-terminal nodes, in the following way. Given an assignment of 0s and 1s to the boolean variables occurring in T, we start at the root of T and take the dashed line whenever the value of the variable at the current node is 0; otherwise, we travel along the solid line. The function value is the value of the terminal node we reach.

****

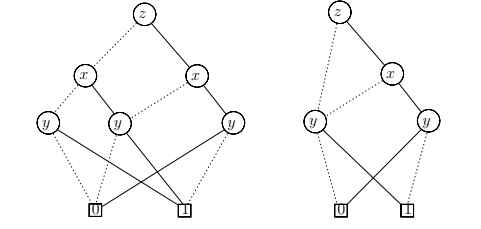
(a) Sharing the terminal nodes of the binary decision tree (b) further optimisation by removing a redundant decision point.

To summarise, we encountered three different ways of reducing a BDD to a more compact form:

C1. Removal of duplicate terminals. If a BDD contains more than one terminal 0-node, then we redirect all edges which point to such a 0-node to just one of them. We proceed in the same way with terminal nodes labelled with 1.

C2. Removal of redundant tests. If both outgoing edges of a node n point to the same node m, then we eliminate that node n, sending all its incoming edges to m.

C3. Removal of duplicate non-terminals. If two distinct nodes n and m in the BDD are the roots of structurally identical subBDDs, then we eliminate one of them, say m, and redirect all its incoming edges to the other one.



after removal of one of the duplicate y-nodes; (b) after removal of another duplicate y-node and

then a redundant x-decision point.