**Ordered BDDs**

We have seen that the representation of boolean functions by BDDs is often compact, thanks to the sharing of information afforded by the reductions C1–C3. However, BDDs with multiple occurrences of a boolean variable along a path seem rather inefficient. Moreover, there seems no easy way to test for equivalence of BDDs. Neither of them can be optimised further by applying the rules C1–C3.



**The complement of the BDD**

****

**A BDD representing the same function as the BDD but having the variable ordering [x, y, z].**

However, testing whether they denote the same boolean function seems to involve as much computational effort as computing the entire truth table for f(x, y, z).

We can improve matters by imposing an ordering on the variables occurring along any path. We then adhere to that same ordering for all the BDDs we manipulate.

Let [x1,...,xn] be an ordered list of variables without duplications and let B be a BDD all of whose variables occur somewhere in the list. We say that B has the ordering [x1,...,xn] if all variable labels of B occur in that list and, for every occurrence of xi followed by xj along any path in B, we have i<j.

An ordered BDD (OBDD) is a BDD which has an ordering for some list of variables.