IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

NPTEL

Lecture 28: Binary Decision Diagrams (Part 2)

PROF. INDRANIL SENGUPTA
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Reduced Ordered BDD (ROBDD)

- An ordered binary decision diagram is said to be *reduced ordered BDD* (ROBDD) if the following two graph reduction rules are applied:
  - Merge any isomorphic subgraphs.
  - Eliminate any node whose two children are isomorphic.
- The advantage of an ROBDD is that it is *canonical* (unique) for a given function.
  - There is exactly one ROBDD for a given variable ordering.
  - This property makes it useful in functional equivalence checking.

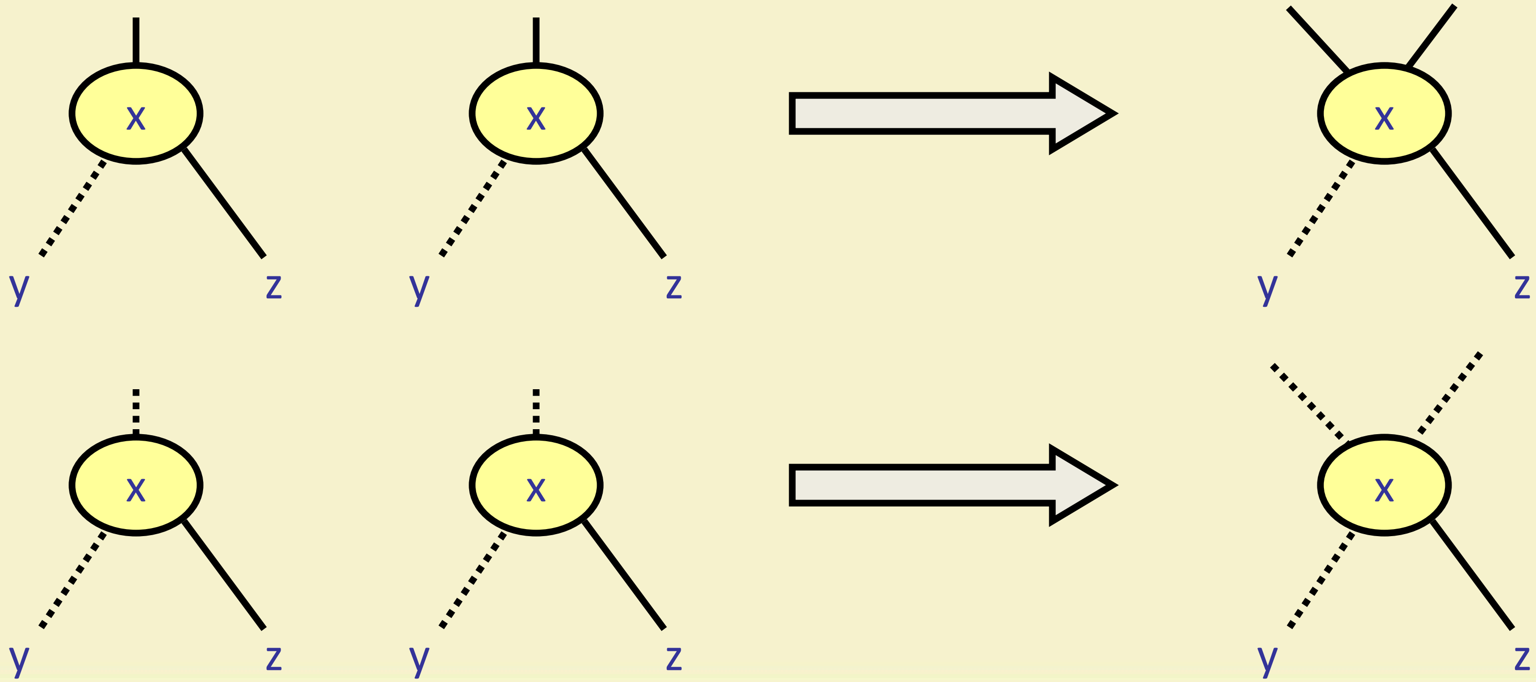
Some Properties of ROBDD

- The following properties hold in a ROBDD:

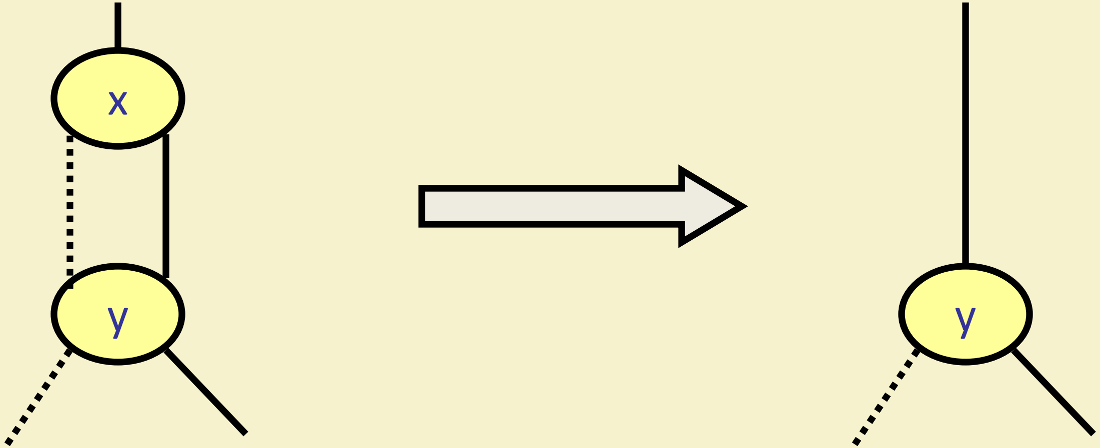
  a) **Uniqueness**: No two distinct nodes *u* and *v* are labeled with the same variable name and have the same low and high successor.

  b) **Non-redundant**: No variable node *u* has identical low and high successor.

Reduction Rules: *Merge Isomorphic Subtrees*

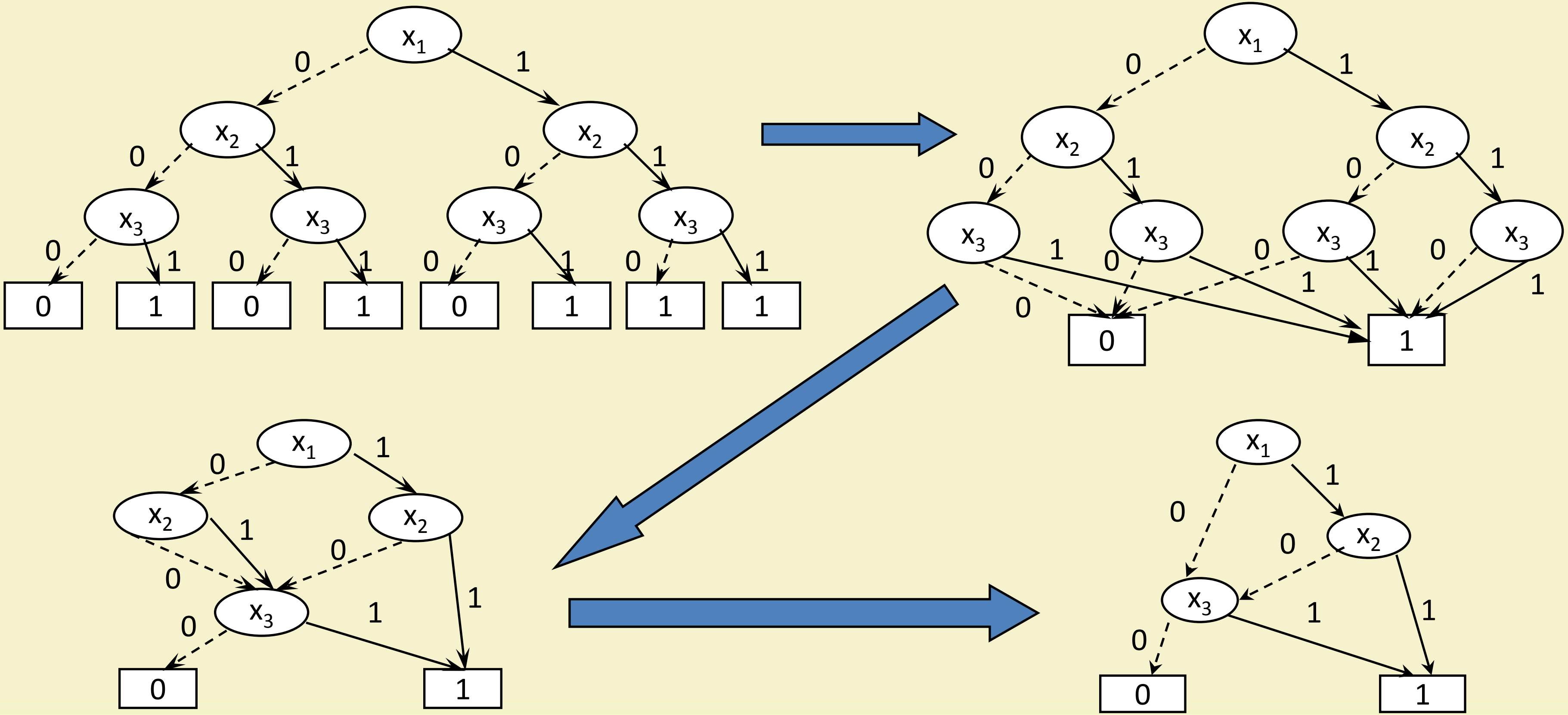


Reduction Rule: *Remove Redundant* Nodes



Construction of ROBDD: an example

Given function: $f\,(x_1, x_2, x_3) \;=\; x_1.x_2 + x_1'.x_3 + x_1.x_2'.x_3$

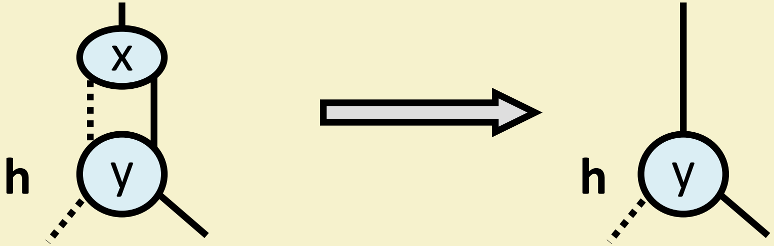

x1
x2
x3
0
1

Some Benefits of BDD

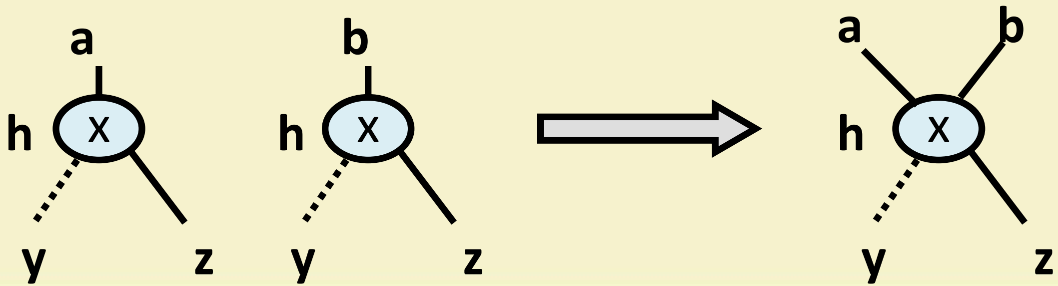
- Checking for tautology is trivial.
  - BDD is a constant 1.
- Complementation.
  - Given a BDD for a function f, the BDD for f' can be obtained by simply interchanging the terminal nodes.
- Equivalence check.
  - Two functions f and g are equivalent if their BDDs (under the same variable ordering) are the same.

Use of BDD in Synthesis

- BDD is canonical for a given variable ordering.
- It implicitly uses factored representation:



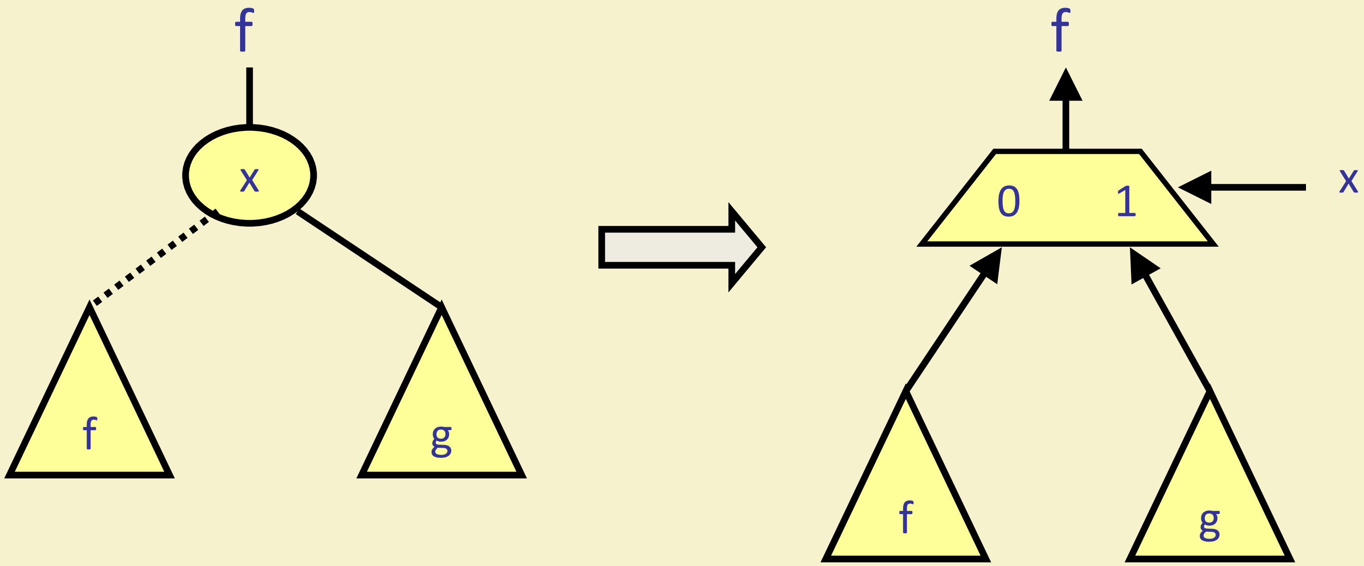
$x'h + xh \;=\; h$



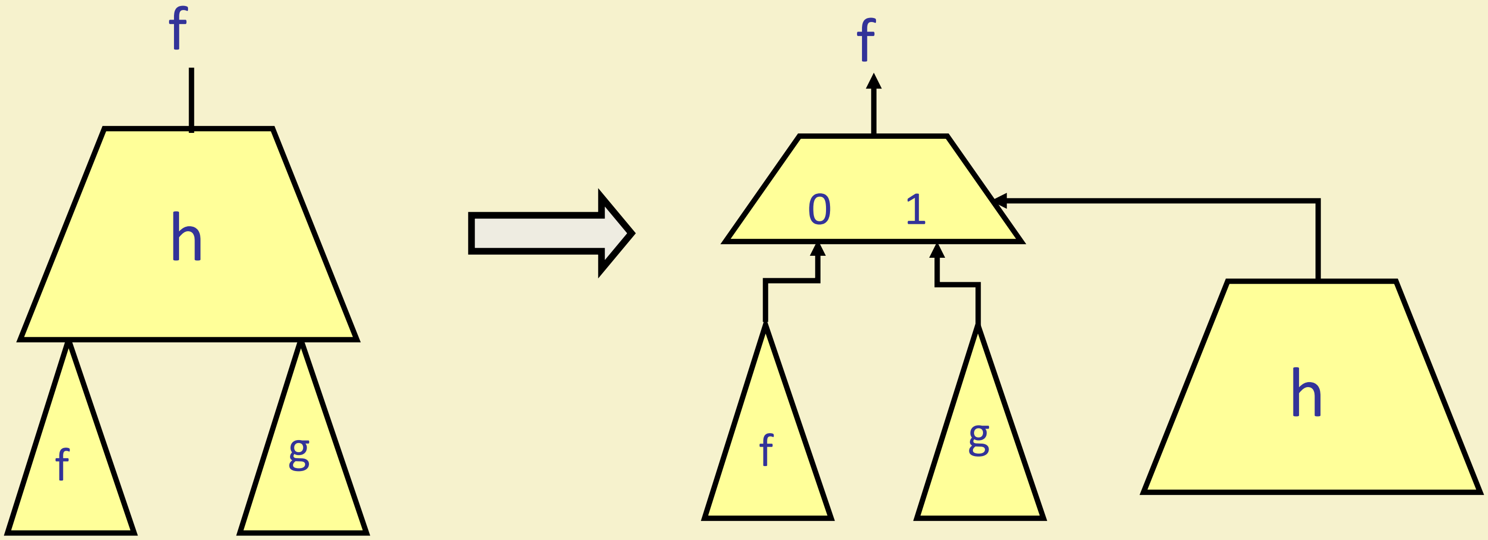
$ah + bh \;=\; (a+b)h$

- Variable reordering can reduce the size of BDD.
  - Implicit logic minimization.
- Some redundancy is also removed during the construction of BDD itself.

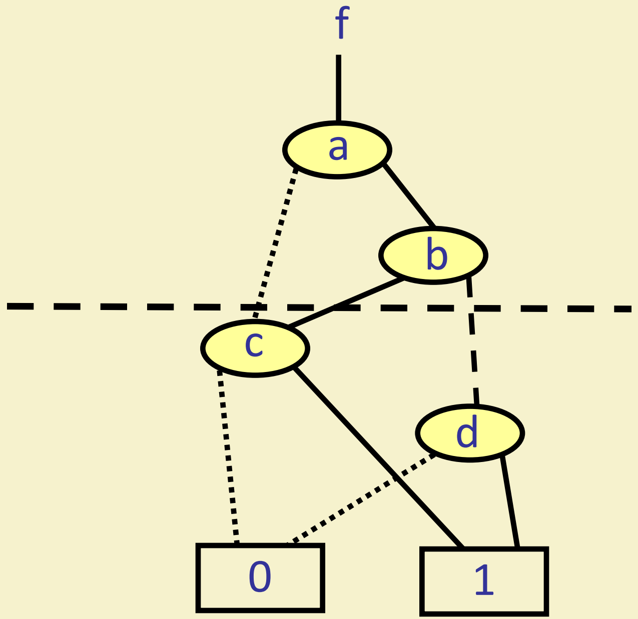
MUX realization of functions

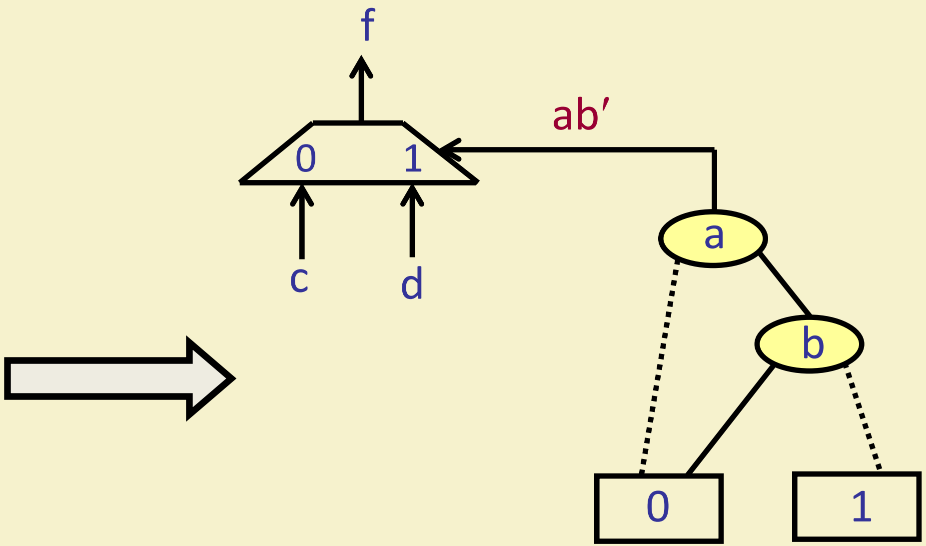


MUX-based Functional Decomposition

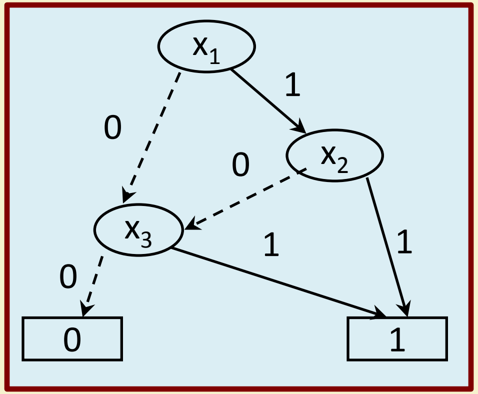


An example ===>


f
a
b
c
d
0
1


f
0
1
c
d
ab′
a
b
0
1

A Complete Mapping Example



To Summarize

- BDDs have been used traditionally to represent and manipulate Boolean functions.
  - Used in synthesis systems.
  - Used in formal verification tools.
  - Efficient packages to manipulate BDDs are available.

END OF LECTURE 28