Home Page

Title Page

Contents

Page 1 of 16

Go Back

Full Screen

Close

Quit

CS206 Formal Methods in CS

Group 6
IIT Bombay

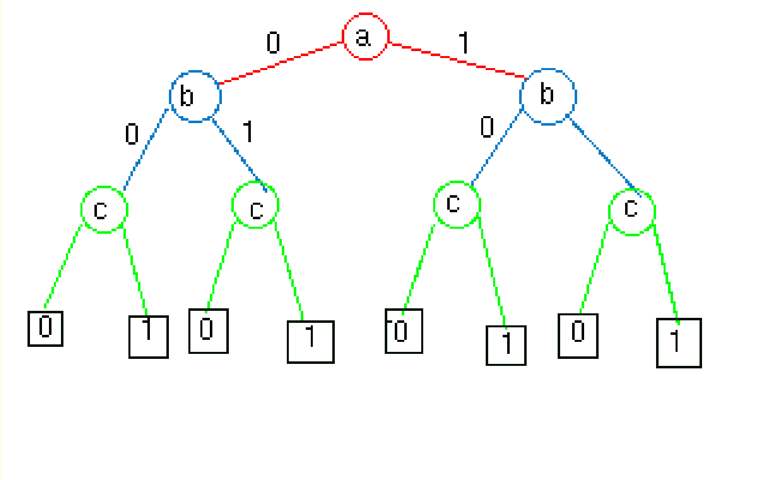
Thu, Feb 13, 2003

Plan for Lecture

- Definition of BDD (Binary Decision Diagram)
- Stepwise Procedure
- Example
- Effect of reordering the branching variables

Definition of OBDD

An OBDD (Ordered Binary Decision Diagram) is a digraph G(V,E) with one root node(node without predecessor).Each non-terminal node has as attribute an index index(v) 1,2...n which contains an input variable of the set x1,x2...xn and successors low(v) and high(v). A terminal node (node without succesor) has as attribute a value ıvalue (v). 0,1



Home Page

Title Page

Contents

◀◀ ▶▶

◀ ▶

Go Back

Full Screen

Close

Quit

Isomorphs

Two OBDDs F=(V,E) and F'=(V',E') are isomorphic, if a bijective function m:V -> V' exists for all v(V) and m(v) in V' :

- if v is a terminal node: m(v) is a terminal node and value(v) = value(m(v)).
- If v is a non-terminal node: m(v) is a non-terminal node, index(v) = index(m(v)) and m(low(v)) = low(m(v)) and m(high(v)) = high(m(v))

Reduced OBDD, etc

An OBDD is called a Reduced OBDD if there exists :

- neither a node V with low(v) = high(v)
- nor a pair of nodes V,V' with isomorphic subgraphs

Theorem of Bryant

For any boolean function f there is a unique (upto isomorphism) reduced function graph denoting f and any other function graph denoting f contains more vertices.

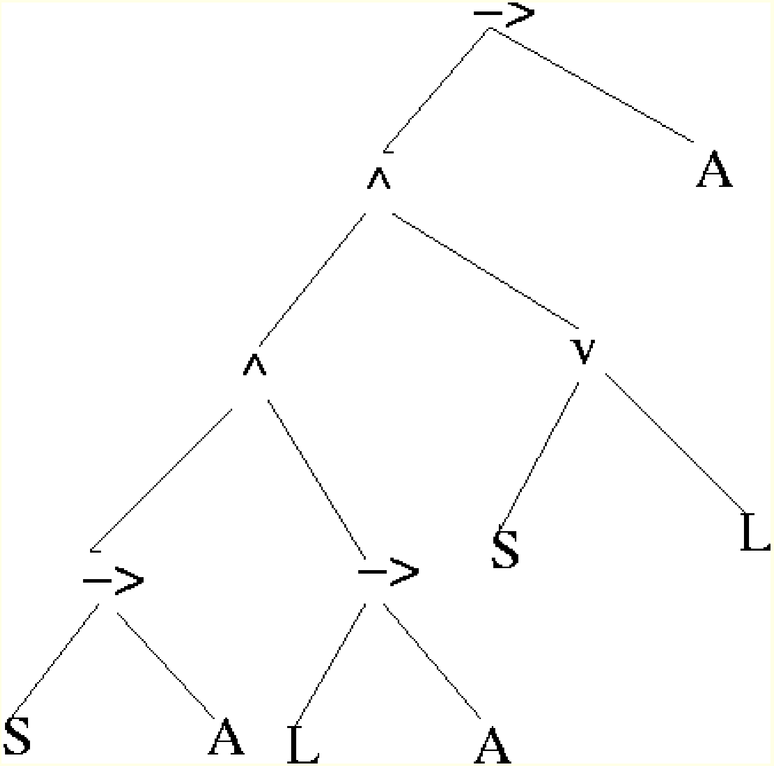
Stepwise procedure

- Decide an ordering of variables (the order in which you will branch on them)
- Draw the full tree, successively branching on all variables
- Merge together identical nodes (nodes with isomorphic subgraphs)
- Eliminate unnecessary nodes (e.g. Node with one branch to '1' and one to '0'
- There should be only a single '0' and '1' node each
- After proper reduction, if the tree reduces to '1', it is a tautology, if it reduces to '0', it is a contradiction, else it is satisfiable.
- Note: Selection of a proper ordering can make the job MUCH easier

Example

Let us consider the following example ..

$$(S \to A) \vee (L \to A) \wedge (S \to L) \to A \tag{1}$$

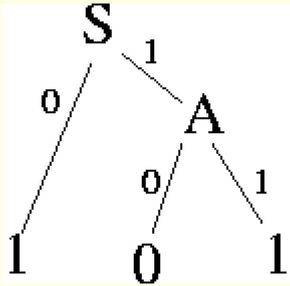


To prove this firstly we have to choose some order of variables.We can choose any order we like.
let us choose S > L > A

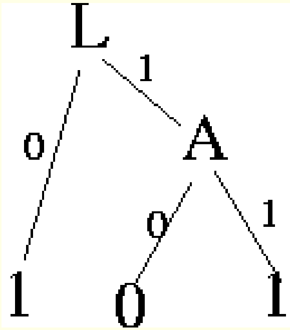
Starting with BDD

Take S -> A part first and convert it in BDD first

Let us start with substituting S = 0 and S = 1 and then try A = 0 and A = 1 for S = 1.



Similarly do for L -> A

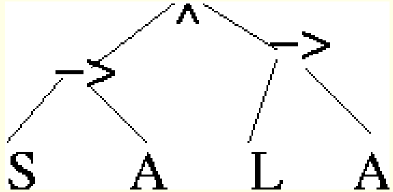


Now we will go one level up and draw bdd for

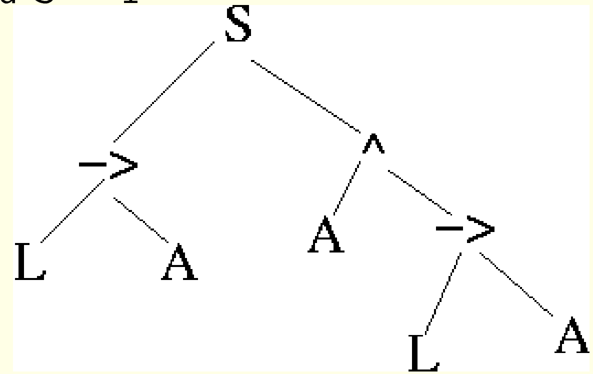
$$(S \rightarrow A) \vee (L \rightarrow A) \tag{2}$$

Home Page
Title Page
Contents
Page 6 of 16
Go Back
Full Screen
Close
Quit

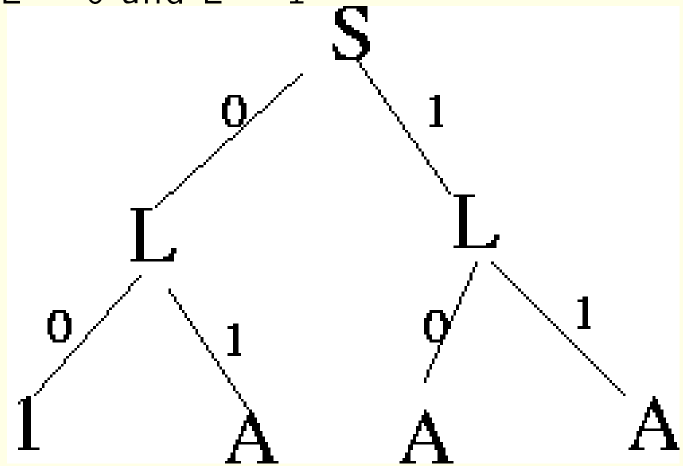
Home Page
Title Page
Contents
Page 7 of 16
Go Back
Full Screen
Close
Quit



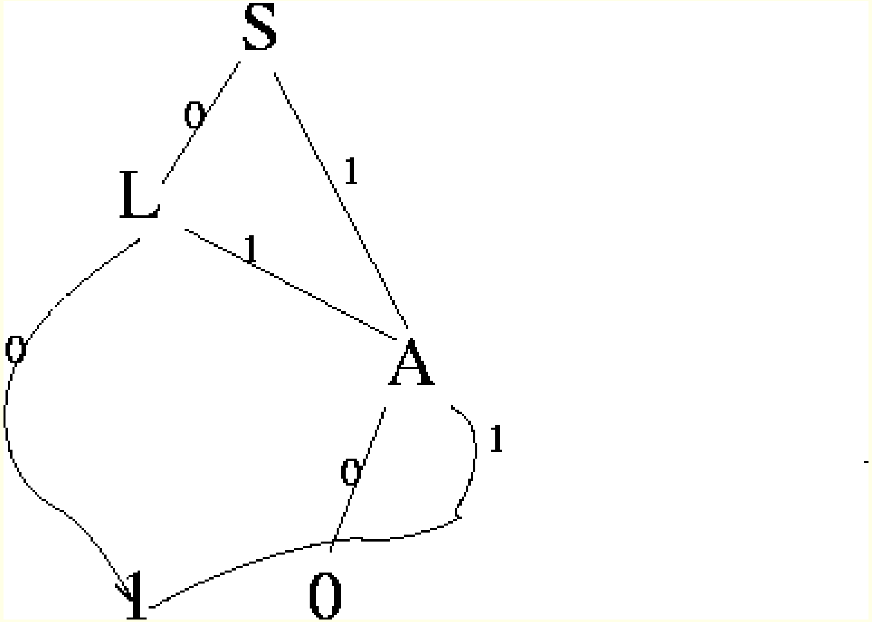
Start with S = 0 and S = 1



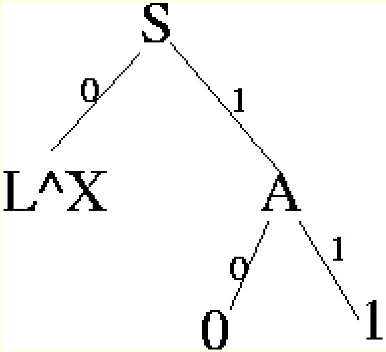
Now branch with L = 0 and L = 1



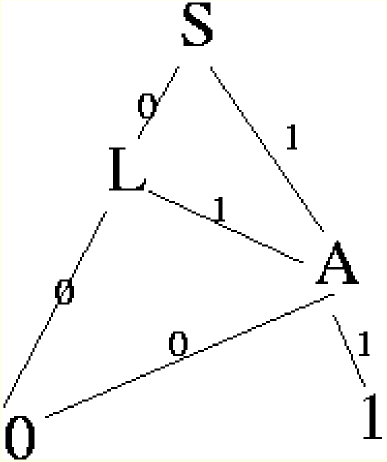
Hence after branching over A we finally get something like now if we call this as X



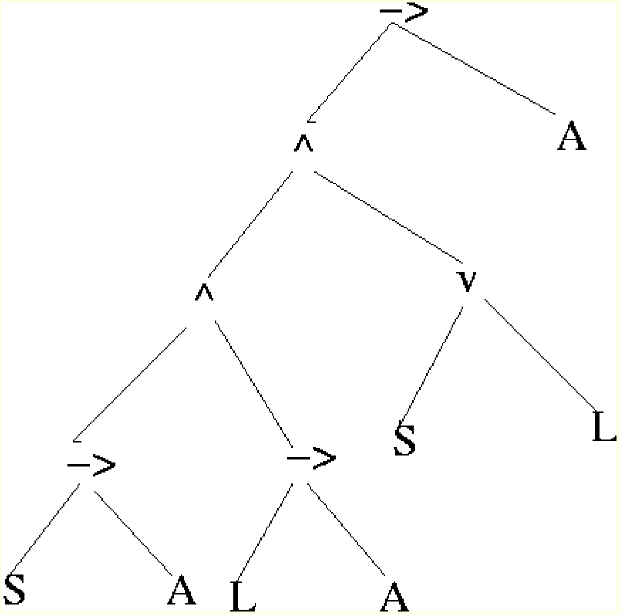
Then we can go one level further high to



which is equivalent to



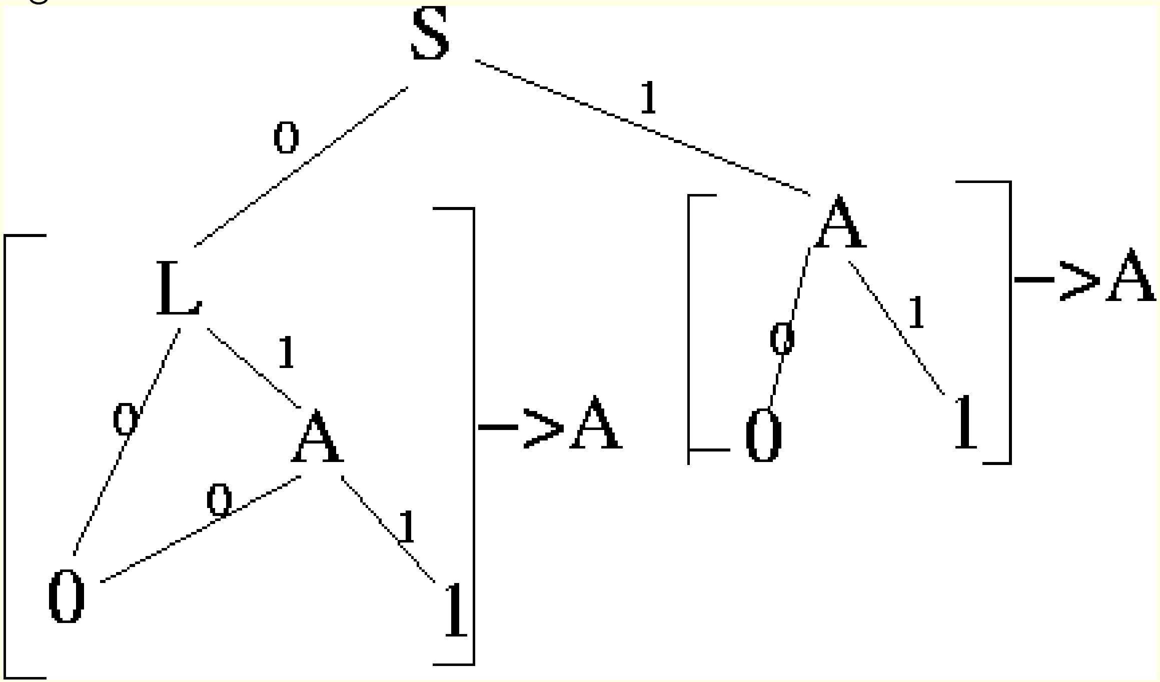
now we are ready to prove



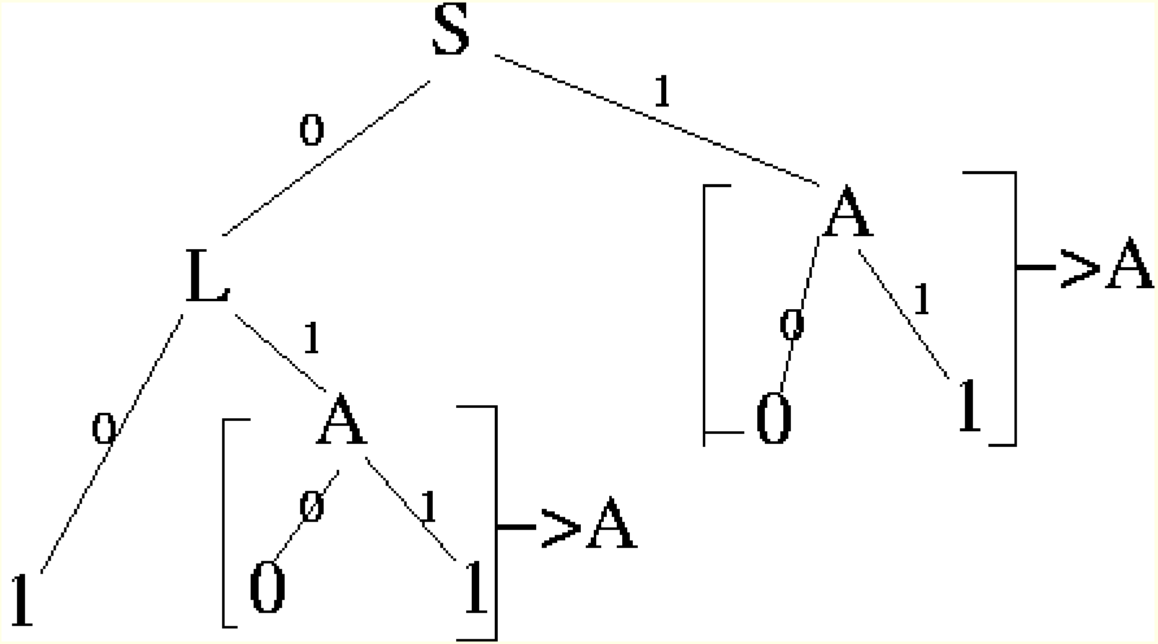
Home Page
Title Page
Contents
Page 10 of 16
Go Back
Full Screen
Close
Quit

Finally....

Putting S=0 and S=1

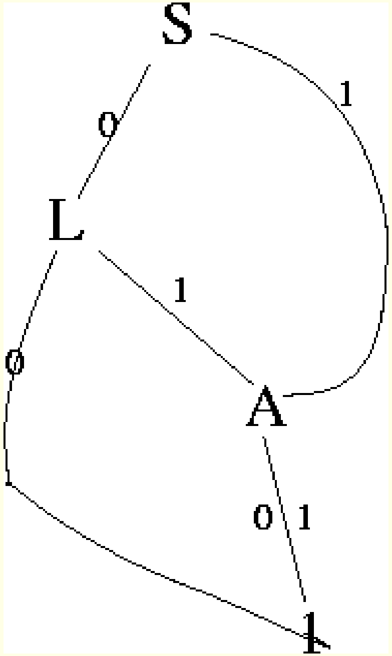


Now we can see that right branch of S will actually reduce to A as if A=0 then we will get 0 -> 0 which is true , simillarly for A=1 1 -> 1 which also is true.



now removing A->A we will get ..

which is



removing all redundant nodes it is actually 1

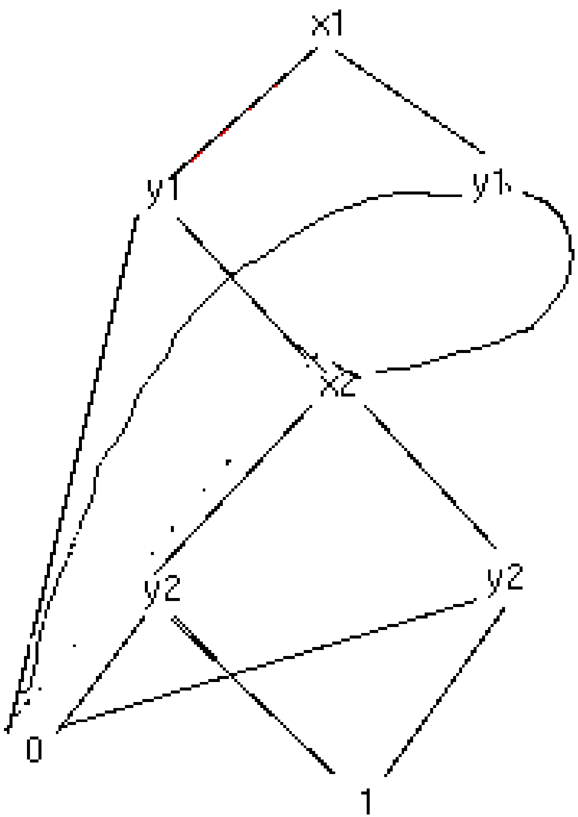
hence this cpmplex equation

$$(S \to A) \vee (L \to A) \wedge (S \to L) \to A \tag{3}$$

actully reduces to 1

Re-ordering Effects

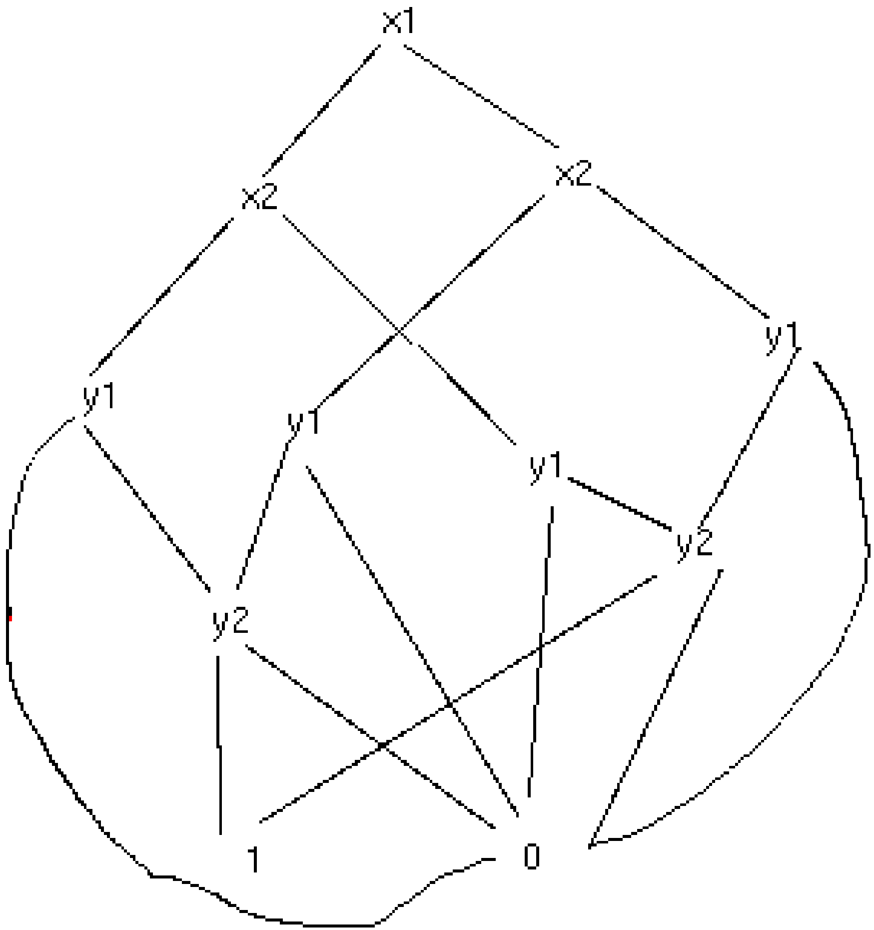
However there is nothing special about this order (S > L > A)
Instead we could have chosen and order we like. Same problem can be tried with say L > A > S .
But in all the cases it should reduce to 1 (as it is a tautology). In the next slides we shall see the effect of this re-ordering.



OBDD of (x1 <=> x2)/\(x2<=>y2) with order x1>y1>x2>y2

With this ordering, see the size of the tree

Now with another ordering...



OBDD of $(x1 <=> y1) \wedge (x2 <=> y2)$ with $x1 > x2 > y1 > y2$

Summary

- OBDD is a special type of graph
- OBDD is useful to vivdly represent decision making. With reduction, we can prove equivalence of two statements
- Proper ordering can have significant effect in size and complexity of the tree

Limitation

- BDDs can grow exponentially in size – verification becomes unfeasible