

ECE 510 OCE BDDs and Their Applications

Lecture 2. Foundations

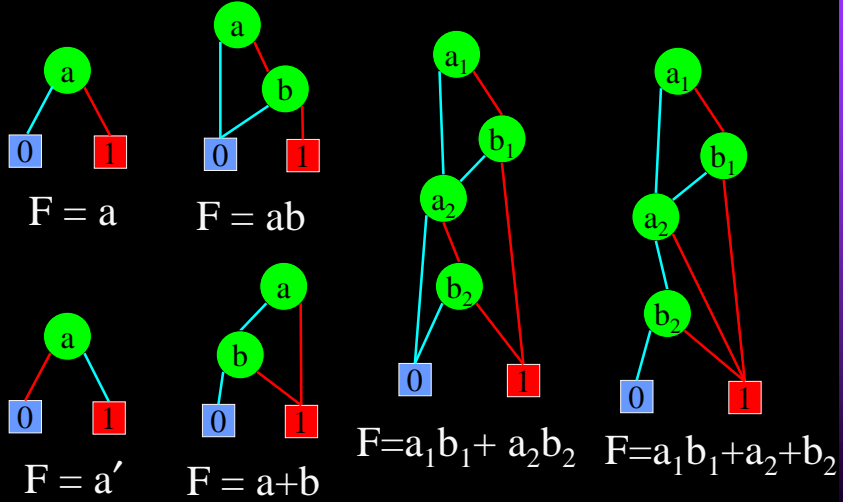
March 30, 2000

Alan Mishchenko

Overview

- Representations of boolean functions
- Canonical and non-canonical forms
- Decision trees and reduction rules
- Building already reduced diagrams
- Zero-Suppressed BDDs
- BDDs with complement edges
- Dynamic variable reordering

Examples



April 3, 2000

ECE 510 OCE: BDDs and Their Applications

3

Known Representations

- Truth table
- SoP (DNF) and PoS (CNF)
- Decision tree
- Reduced decision tree
 - BDD
 - Zero-Suppressed BDD
 - BDD with complement edges

April 3, 2000

ECE 510 OCE: BDDs and Their Applications

4

Truth Table, SoP, and PoS

x_1	x_2	x_3	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

- Sum-of-products

$$F = x_1'x_2x_3 + x_1x_2'x_3 + x_1x_2x_3$$

- Product-of-sums

$$F = (x_1+x_2+x_3) \& (x_1+x_2+x_3') \& (x_1+x_2'+x_3) \& (x_1'+x_2+x_3) \& (x_1'+x_2'+x_3)$$

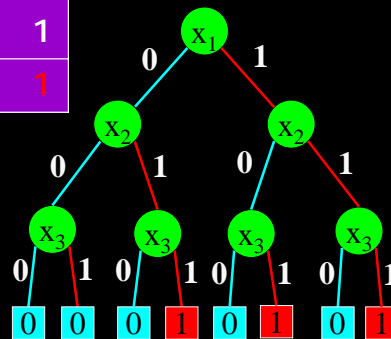
April 3, 2000

ECE 510 OCE: BDDs and Their Applications

5

Truth Table and Decision Tree

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
F	0	0	0	1	0	1	0	1



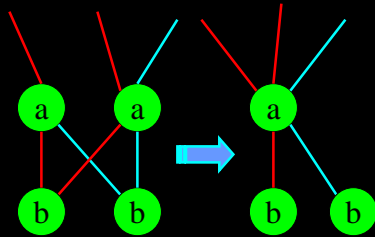
April 3, 2000

ECE 510 OCE: BDDs and Their Applications

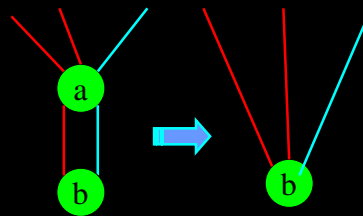
6

Reduction of Decision Tree

Rule 1: Merging Rule:
Nodes must be unique



Rule 2: Elimination Rule:
Redundant tests should not be present

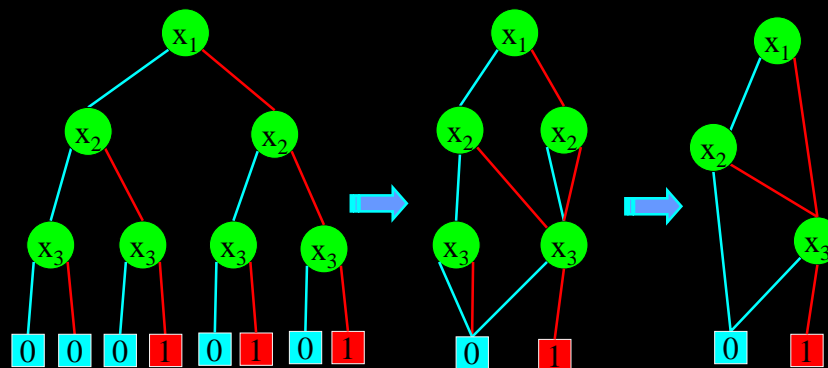


April 3, 2000

ECE 510 OCE: BDDs and Their Applications

7

Example of Decision Tree Reduction



Decision Tree



reduction

BDD

April 3, 2000

ECE 510 OCE: BDDs and Their Applications

8

Shannon Expansion

- A boolean function can be expanded with respect to any variable

$$F(x,y,z) = x' \& F_{x'} + x \& F_x$$

where $F_{x'}$ and F_x are positive (negative) cofactors

$$F_{x'} = F(0,y,z), F_x = F(1,y,z)$$

Building Already Reduced Tree

Build a reduced tree for **F** by calling **Build(F, 1)**, if variable ordering $x_1 < x_2 < x_3 < \dots < x_n$ is given

```
function Build ( F, i )
  if ( i > n )
    if ( F == 0 ) return Node0
    else       return Node1;
  else
    v0 = Build ( Fx', i+1 );
    v1 = Build ( Fx , i+1 );
    return CreateNode ( i, v0, v1 );
```

Creating a Unique Node

assuming there is a node table with functions `CheckExists()` and `Insert()`

```
function CreateNode( Var, Fx', Fx )  
if ( Fx' == Fx )  
    return Fx';  
else if ( CheckExists( Var, Fx', Fx ) )  
    return the existing node;  
else Insert( Var, Fx', Fx );  
    return the new node;
```

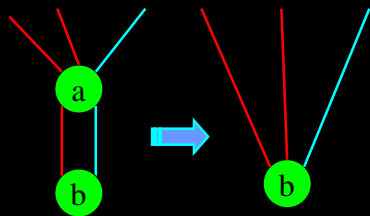
April 3, 2000

ECE 510 OCE: BDDs and Their Applications

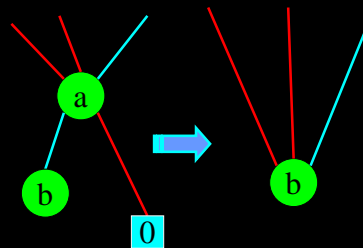
11

Zero-Suppressed BDDs

Rule 2: Elimination Rule:
Redundant tests should
not be present



Rule 2: Elimination Rule:
Remove nodes with 1-edge
pointing to 0-terminal



April 3, 2000

ECE 510 OCE: BDDs and Their Applications

12

Complement Edges

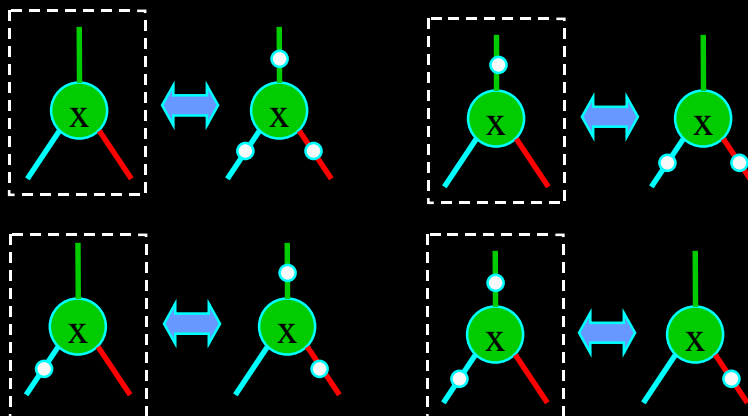
- BDD edges can carry attributes
- Using complementation as an attribute for edges saves BDD nodes and makes NOT a constant time operation
- To maintain canonicity, the use of attribute edges should be constrained

April 3, 2000

ECE 510 OCE: BDDs and Their Applications

13

Equivalent Pairs of Functions



April 3, 2000

ECE 510 OCE: BDDs and Their Applications

14