Simple Disjunctive Decomp. On BDDs

- f = w'x'z' + wx'z + w'yz + wyz'
- (w'z')x' + (wz)x' + (w'z)y + (wz')y
- f= F(ø(w, z), x, y)
- Ø = XNOR(w, z);
- Ø' = XOR(w, z)
- Orthonormal Expansion of f w.r.t. Ø= ø x' + ø' y
- Decomposition through partition matrix
- Test for existance of a decomp:
- Reduce partition matrix, get column multiplicity
- If col. Mult. <= 2, you WILL FIND A DECOMP!

Decomp. May or maynot exist

0	1	0	1
0	0	1	0
0	0	1	0
0	1	0	1

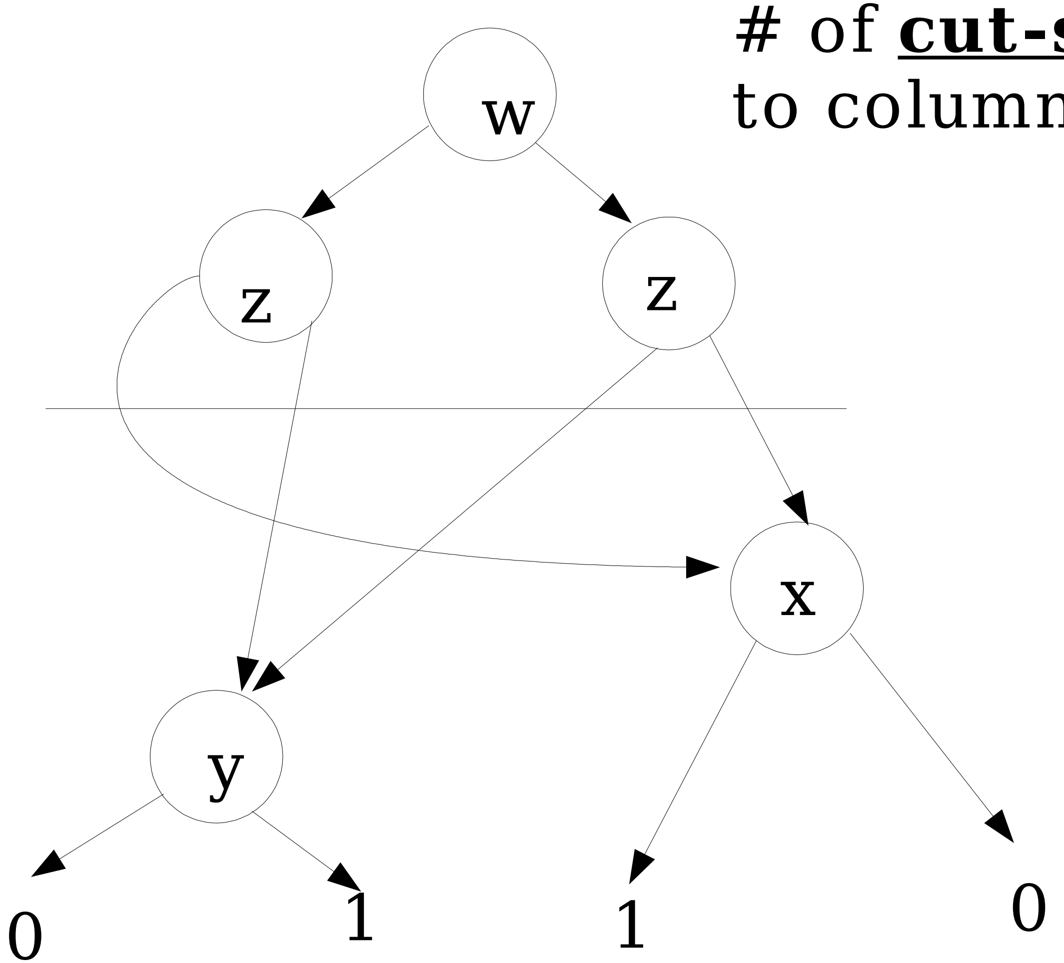
Significance of Column Multiplicity

0	0
1	0
0	1
1	0

- Column mult = 2:
- Bound set function corresponds to 2 values (0 /1 or ø/ø')
- One-bit encoding of Ø(w, z) exists
- This gives us the simple decomp w.r.t. Orthonormal basis

Simple Decomp on BDDs: Basic Concepts

$F = w'z'x' + wzx' + w'zy + wz'y$
$F = H(G(w, z), x, y)$



of **cut-set nodes** corresponds to column multiplicity: BDD MAGIC

Free set term: x'
Associated bound-set cubes:

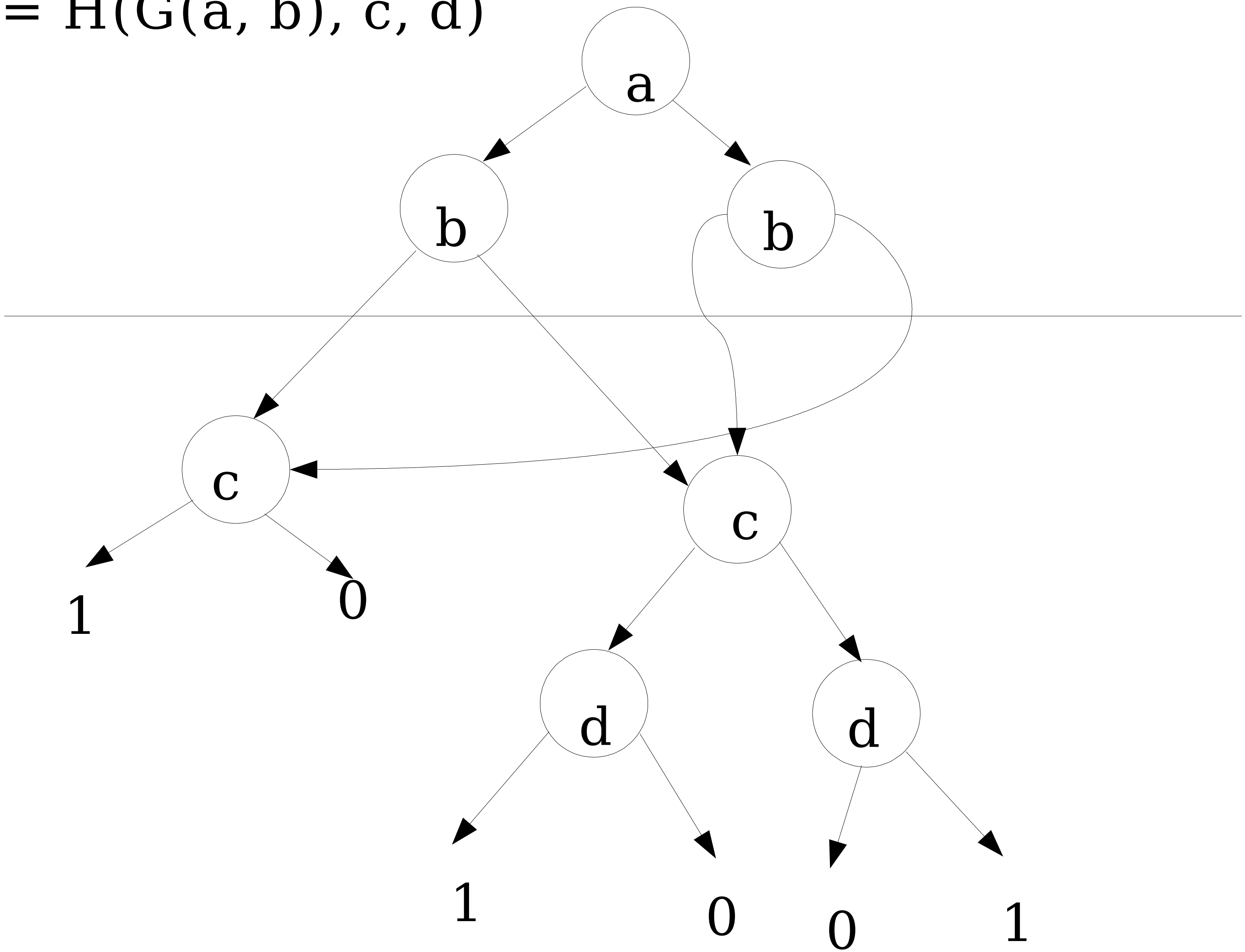
Free-set term: y
Bound-set cubes:

Orthonormality guaranty!

Simple Decomp on BDDs: Basic Concepts

F = a'b'c' + a'bc'd' + a'bcd + abc' + ab'cd' + ab'cd

F = H(G(a, b), c, d)



BDD-Based Simple Decomp. Algorithm

- F = H(G(w, z), x, y)
- Build BDD for F.
- Bound set vars together on top
- Free set on Bottom
- Perform a “CUT”: Partitioning of vars
- #cut-set nodes <= 2: orthonormal expansion corresponding to simple decomp exists! Problem solved!
- **<u>But what if #cut-set nodes = 3 ?</u>**