Boolean functions

$$f : \{0,1\}^n \to \{0,1\}$$

Examples:

$\bar{0} \stackrel{\text{def}}{=} 1$

$\bar{1} \stackrel{\text{def}}{=} 0$

$$x \cdot y = \begin{cases} 1 & \text{if } x = y = 1 \\ 0 & \text{o.w.} \end{cases}$$

$$x + y = \begin{cases} 0 & \text{if } x = y = 0 \\ 1 & \text{o.w.} \end{cases}$$

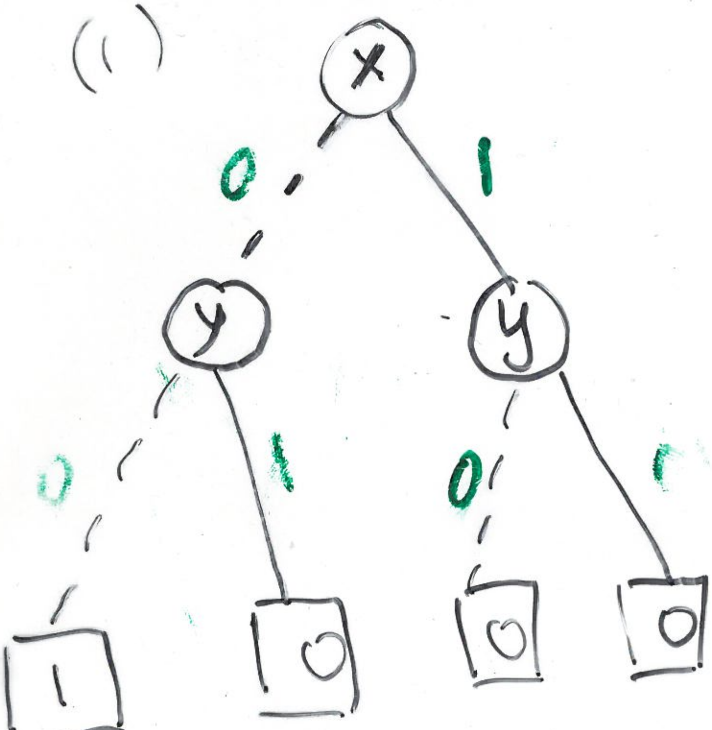
$x \oplus y = 1$ iff exactly one of x and y equals 1

Representing boolean func.

- truth tables
- propositional formulas
- prop. formulas in CNF or DNF
- binary decision trees
- binary decision diagrams (BDDs)

xy	f(x,y)
00	1
01	0
10	0
11	0

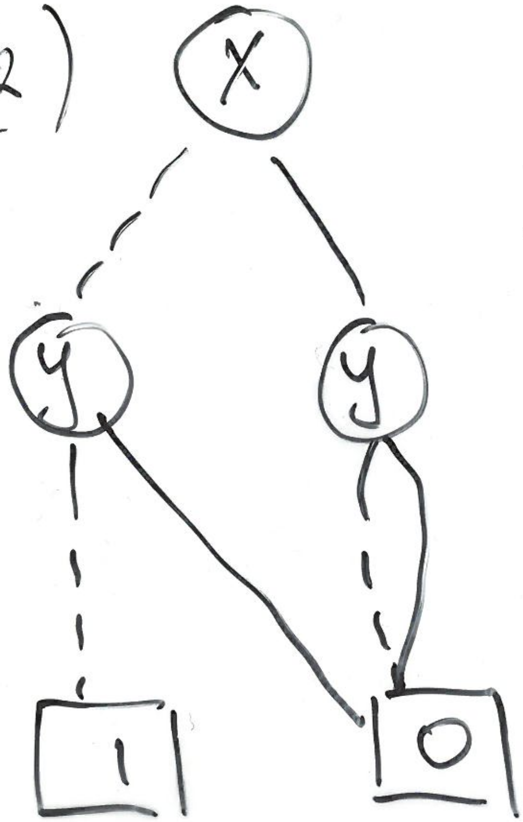
(1)



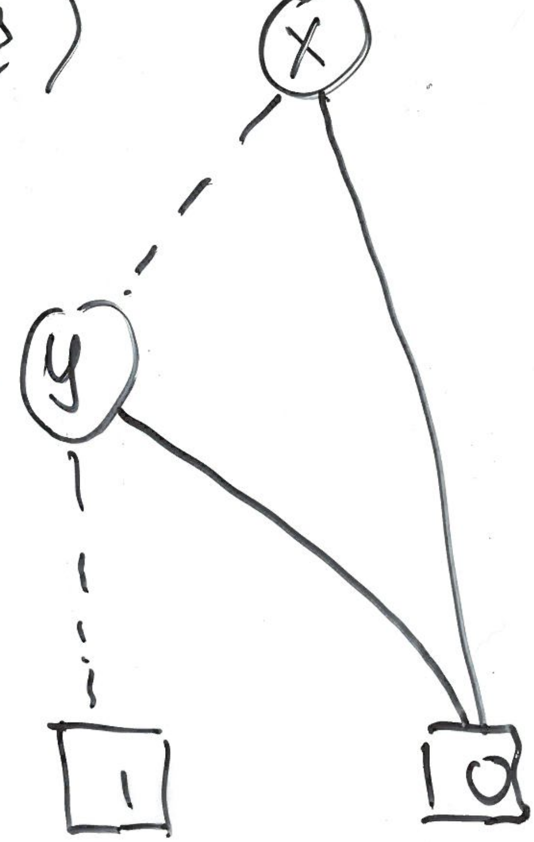
BDDs are a generalization of binary decision trees.

BDPs

(2)



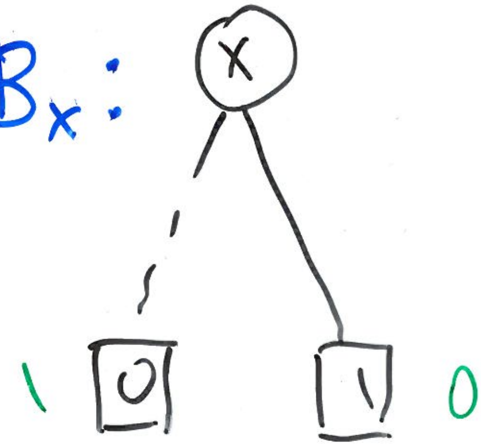
(3)



(1), (2), (3) represent the same function

B_0: 

B_1: 

B_x: 

the constant 0

variable x

Operations $\cdot$ and $+$ on BDDs (a naive way)

Let B_g, B_f represent BDDs for function g, f respectively.

BDD for $g \cdot f$:

attach B_f to B_g in terminals $\boxed{1}$

Idea: $g \cdot f$ is 1 iff both g and f return 1.

x	y	x·y
0	0	0
0	1	0
1	0	0
1	1	1

BDD for $g+f$ – similar, but for $\boxed{0}$

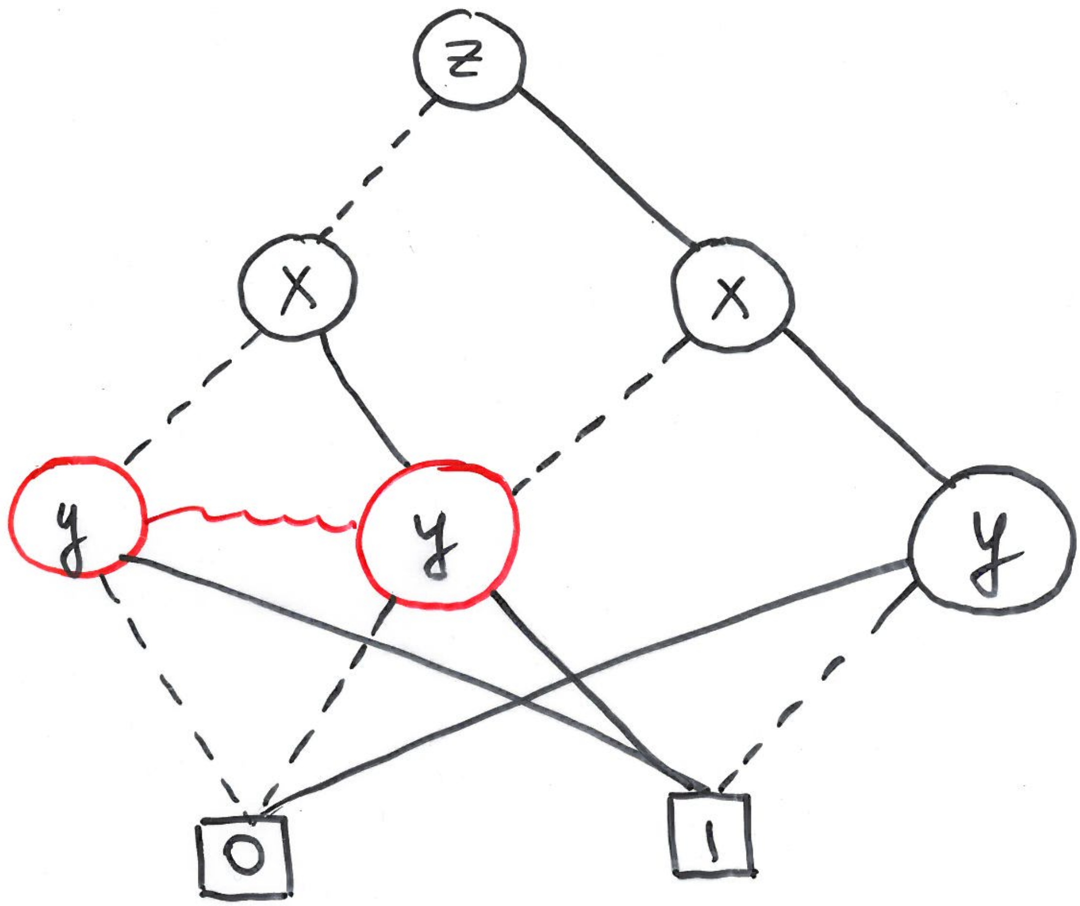
Reducing BDDs

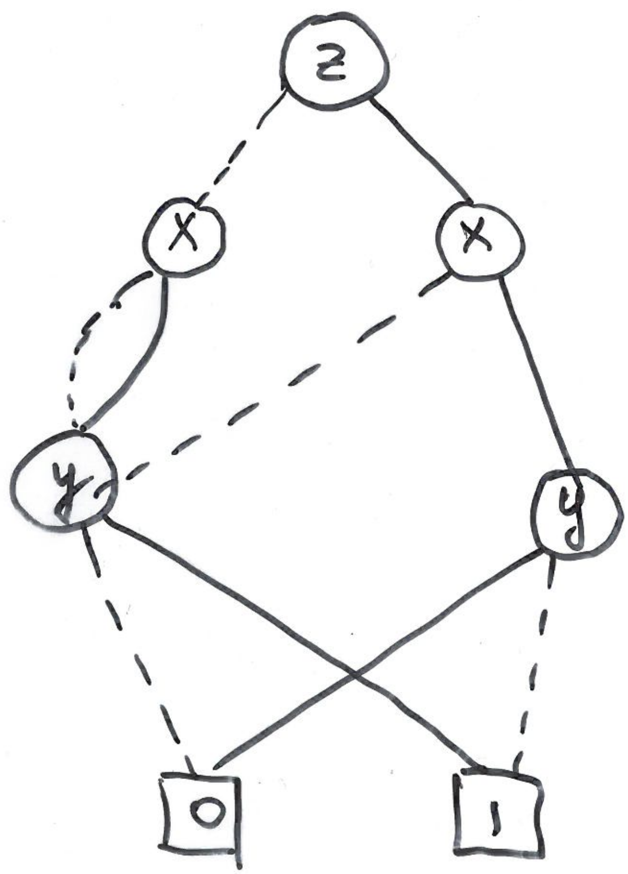
C_1: Remove duplicate terminals

C_2: Remove redundant tests

C_3: Remove duplicate non-terminals

A BDD is called <u>reduced</u> if none of $C_1 - C_3$ are applicable.


z
x
x
y
y
y
0
1


z
x
x
y
y
0
1

Ordered BDDs (OBDDs)

OBDD

Ordering

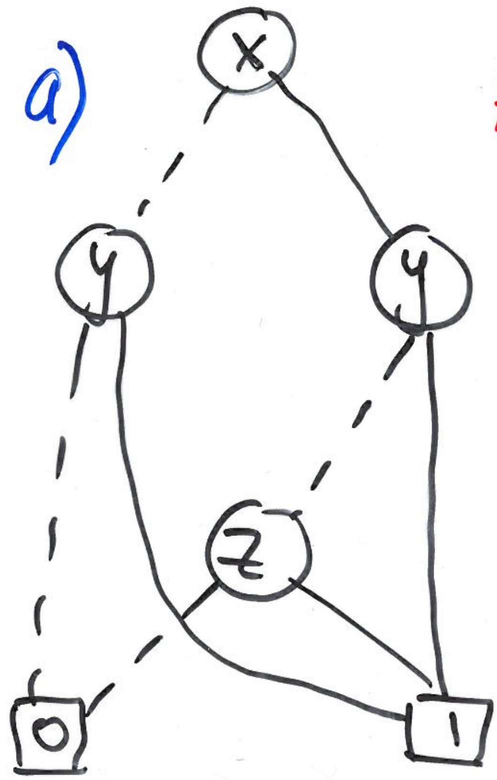
[x, y, z] *

another ordering

[w, x, y, v, z] **

Orderings * and ** are compatible

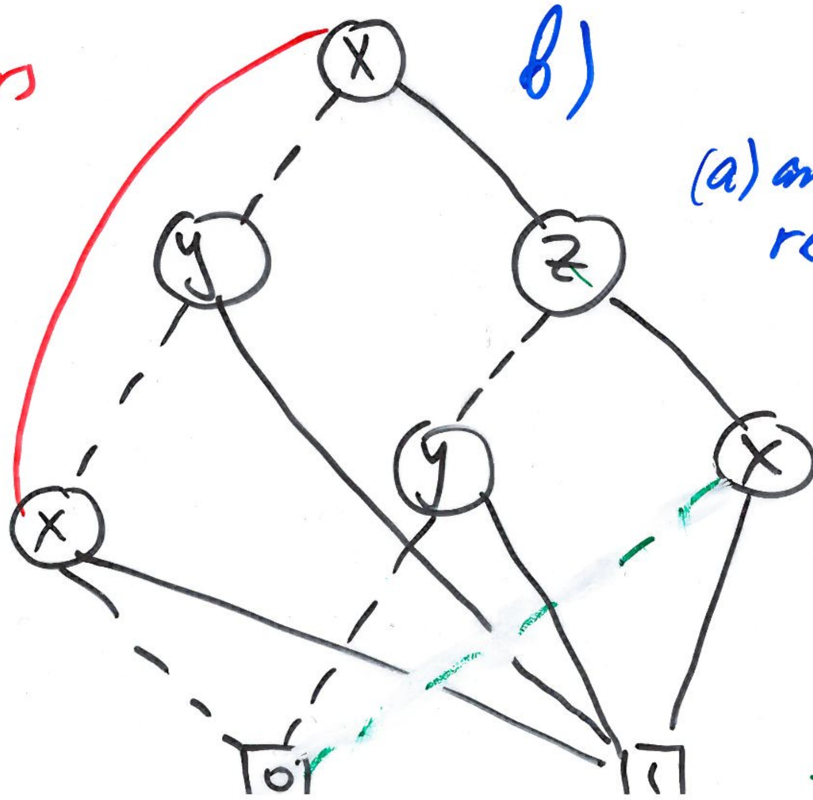
a)



For each path, ~~all end~~ all variables of the path (if used at all) are used in that order

BDD but not OBDD:

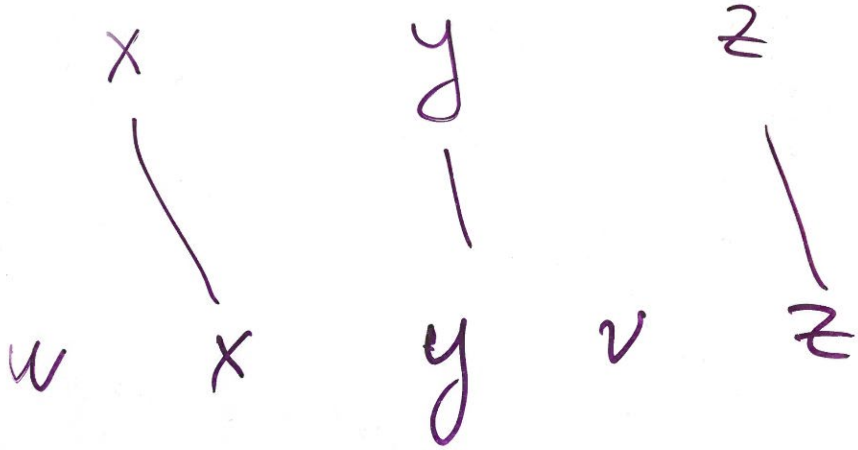
b)



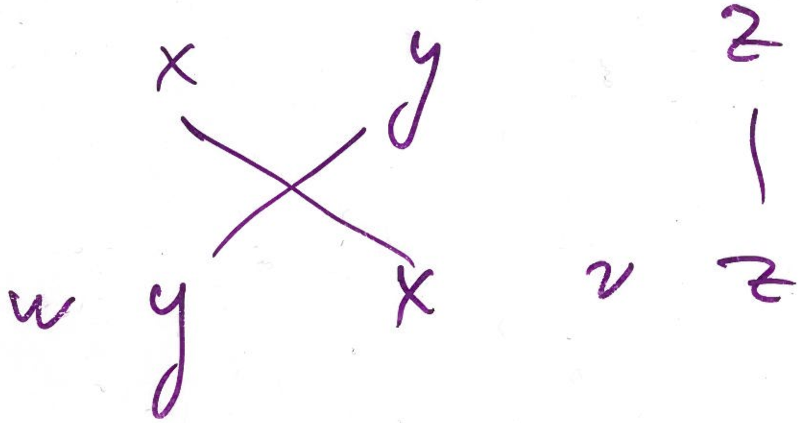
x occurs twice on this path

(a) and (b) represent the same function

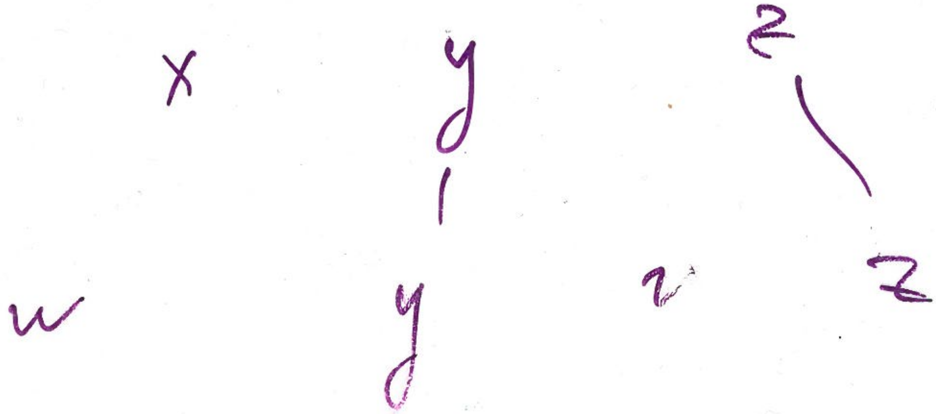
x	z	x	f
1	1	0	0



compatible



not compatible



compatible

- If B_1 and B_2 are two **reduced** OBDDs with compatible var. ordering then ~~and~~ they have identical structure iff they represent the same boolean function

- For a given ordering, the reduced OBDD for a given function is unique

Note: the choice of ordering affects the size of the resulting BDD

finding an optimal ordering is computationally expensive, but there are good heuristics

Advantages of the canonical representation

- If $f(x_1, .., x_n)$ does not depend on x_i, then the reduced OBDD does not contain it.

- Easy to check equivalence of functions

- Easy to check for validity: f is valid iff its reduced OBDD is B_1: $\boxed{1}$

- — " — satisfiability

  f is satisfiable iff its reduced OBDD is not $\boxed{0}$

- Easy to check for implication:

$\boxed{f \text{ implies } g}$ (i.e., if f computes 1, so does g)

Construct the reduced OBDD for $f \cdot \bar{g}$.

This is B_0: $\boxed{0}$ iff the implication holds

$f \to g \equiv \neg f \vee g$

negate: $(f \wedge \neg g) \equiv \bot$

iff

$f \to g \equiv \top$

$f \cdot \bar{g}$