Finite Boolean Algebra

Lecture 8

Discrete Mathematical Structures

Finite Boolean Algebra

- Part I: Finite Boolean Algebra
  - Boolean algebra: a special type of lattice
  - Substitution rule for Boolean algebra
- Part II: Logical Design
  - Boolean expressions
  - Circuit Design

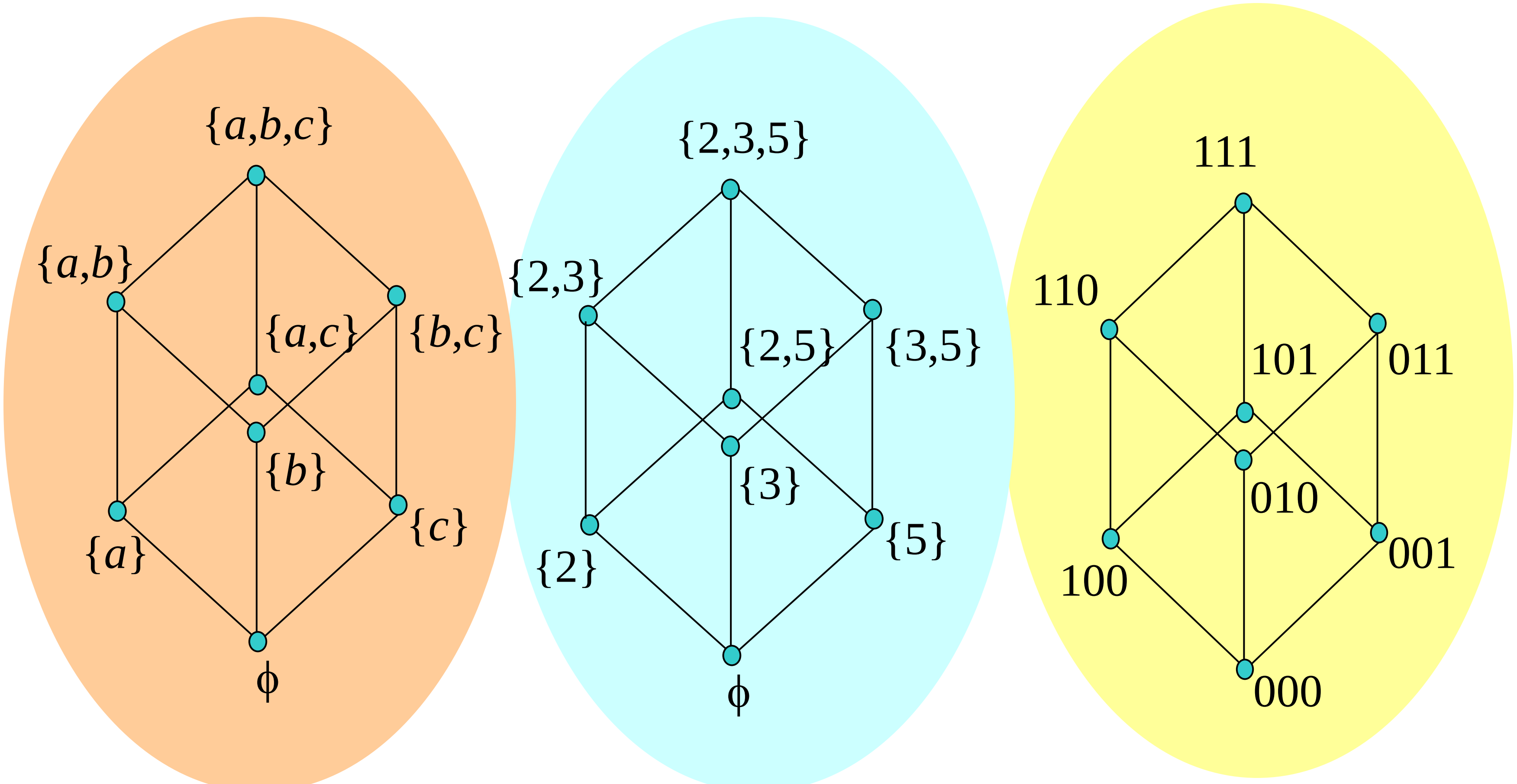
Lattice ($P(S)$,$\subseteq$)

- For a finite set S:
  - The power set of S, $P(S)$, is a finite set of $2^{|S|}$ elements.
  - Set inclusion is a partial order on $P(S)$.
  - ($P(S)$,$\subseteq$) is a lattice
    - For any subsets of S, S_1 and S_2, $S_1 \cup S_2$ is the (unique) least upper bound of S_1 and S_2; and $S_1 \cap S_2$ is the (unique) greatest lower bound of S_1 and S_2

Isomorphism of Finite Lattices

- If $S_1=\{x_1,x_2,\ldots,x_n\}$ and $S_2=\{y_1,y_2,\ldots,y_n\}$ are any two finite sets with the same number of elements, then $(P(S_1),\subseteq)$ and $(P(S_2),\subseteq)$ are isomorphic.
- Proof:
  - A one-to-one correspondence: $f(x_i)=y_i$ for $i=1,2,\ldots,n$.
  - A one-to-one correspondence from $P(S_1)$ to $P(S_2)$: f(A)
    - For any subsets A,B of S_1, $A\subseteq B$ iff. $f(A)\subseteq f(B)$

Hasse Diagrams of Isomorphic Lattices



Lattice B_n

- B_n has 2^n elements.
- Each element is labeled by a sequence of 0's and 1's of length n.
- For any elements $x=a_1a_2...a_n$, $y= b_1b_2...b_n$, in B_n (each a_i,b_i is 0 or 1):
  - $x \preccurlyeq y$ iff. $a_k \preccurlyeq b_k$ for $k=1,2,...,n$.
  - $x \wedge y = c_1c_2...c_n$, where $c_k=\min\{ a_k,b_k \}$
  - $x \vee y = d_1d_2...d_n$, where $d_k=\max\{ a_k,b_k \}$

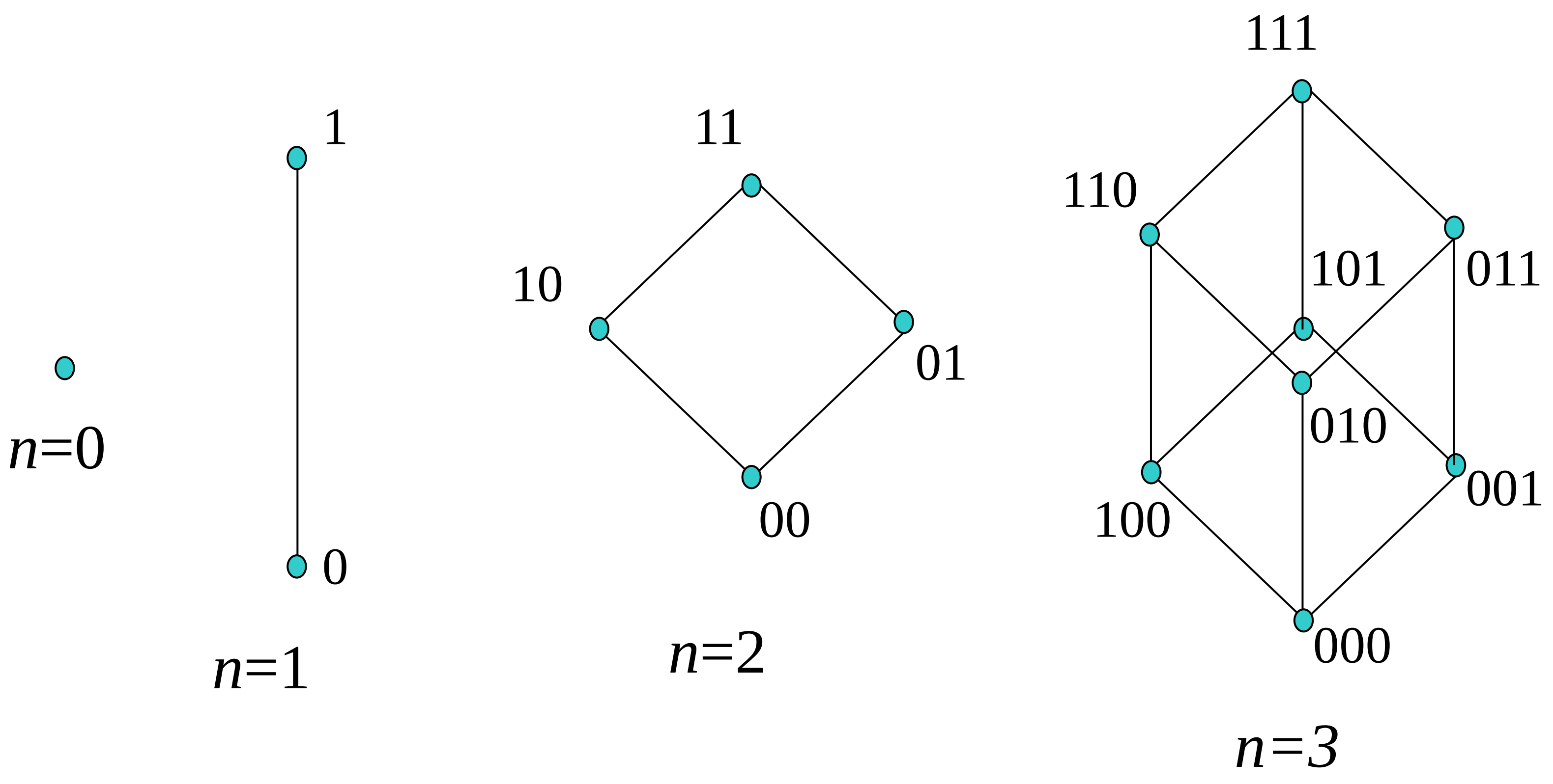
B_n as Product of n B's

- B_1, $(\{0,1\}, \wedge, \vee, 1, 0, ')$, is denoted as B.
- For any $n \geq 1$, B_n is the product $B \times B \times \ldots \times B$ of B, n factors, where $B \times B \times \ldots \times B$ is given the product partial order.

Product partial order:

$$x \leq y \text{ if and only if } x_k \leq y_k \text{ for all } k.$$

Hasse Diagrams of B_n



Boolean Algebra

- A finite lattice isomorphic with B_n is called a Boolean Algebra.
- An example, D_6
  - The set of D_6 is all positive integer divisors of 6
  - The partial order with D_6 is divisibility
  - D_6 is isomorphic with B_2

$$f : D_6 \to B_2 : \ f(1)=00, f(2)=10, f(3)=01, f(6)=11$$

B_n is distributive and Complemented

- For any x in B_n, x has a complement $x' = z_1z_2\ldots z_n$, where z_k=1 if x_k=0, and z_k=0 if x_k=1.
- For any elements $x=a_1a_2\ldots a_n$, $y= b_1b_2\ldots b_n$, $z=c_1c_2\ldots c_n$, in B_n , (each a_i,b_i,c_i is 0 or 1):
  - $x\wedge(y\vee z)=(\min\{a_1,\max\{b_1,c_1\}\})(\min\{a_2,\max\{b_2,c_2\}\})\ldots(\min\{a_n,\max\{b_n,c_n\}\}) = (\max\{\min\{a_1,b_1\}, \min\{a_1,c_1\}\})(\max\{\min\{a_2,b_2\}, \min\{a_2,c_2\}\})\ldots(\max\{\min\{a_n,b_n\}, \min\{a_n,c_n\}\}) = (x\wedge z)\vee(y\wedge z)$
  - Similarly, $x\vee(y\wedge z)=(x\vee z)\wedge(y\vee z)$
  - So, B_n is distributive.

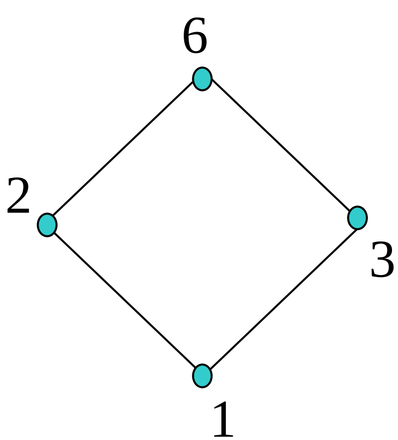
A General Definition of Boolean Algebra

- A distributive and complemented lattice is called a Boolean Algebra.

- This definition is equivalent to the previous one.

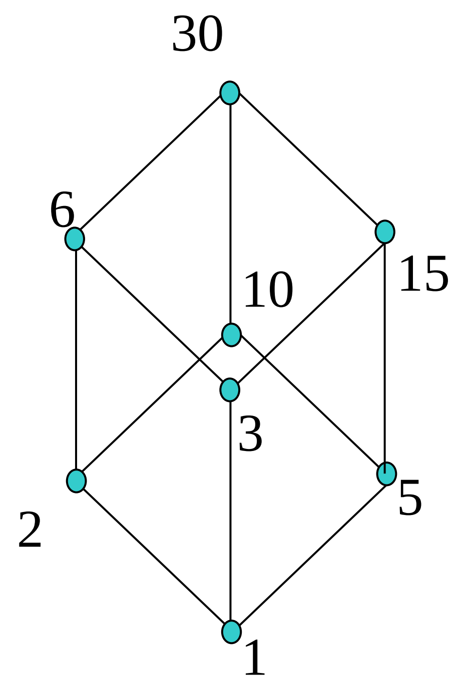
Some Examples

D_n is the poset of all positive divisors of n with the partial order "divisibility".



D_6

D_{20} is not a Boolean algebra



D_{30}

D_n as Boolean Algebra

- Let $n=p_1p_2\ldots p_k$, where the p_i are distinct primes. Then D_n is a Boolean algebra.
- Sketch of proof:
  - Let $S=\{p_1,p_2,\ldots p_k\}$, and for any subset T of S, a_T is the product of the primes in T.
  - Note: any divisor of n must be some a_T. And we have $a_T|n$ for any T.
  - For any subsets V,T, $V\subseteq T$ iff. $a_V|a_T$, *and* $a_V\wedge a_T=\mathrm{GCD}(a_V, a_T)$ and $a_V\vee a_T=\mathrm{LCM}(a_V, a_T)$.
  - $f: P(S)\to D_n$ given by $f(T)= a_T$ is an isomorphism from $P(S)$ to D_n.

D_n as Boolean Algebra (cont.)

- If n is a positive integer and $p^2|n$, where p is a prime number, then D_n is not a Boolean algebra.
- Proof:
  - Since $p^2|n$, $n= p^2q$ for some positive integer q. Note that p is also a element of D_n, then if D_n is a Boolean algebra, p must have a complement p', which means GCD(p,p')=1 and LCM(p,p')=n. So, pp'=n, which leads to p'=pq. So, GCD(p,p') = GCD(p,pq)=p, contradiction.
- So, D_n is a Boolean algebra **if and only** if $n=p_1p_2\ldots p_k$, where the p_i are distinct primes.

Operation Correspondence

- Any formula involving ∪ or ∩ that holds for arbitrary subsets of a set *S* will continue to hold for arbitrary elements of a Boolean algebra *L* if ∧ is substituted for ∩ and ∨ for ∪ .

$$(x')' = x \Leftrightarrow \overline{(\overline{A})} = A$$

$$(x \wedge y)' = x' \vee y' \Leftrightarrow \overline{(A \cap B)} = \overline{A} \cup \overline{B}$$

$$(x \vee y)' = x' \wedge y' \Leftrightarrow \overline{(A \cup B)} = \overline{A} \cup \overline{B}$$

$$x \leq y \text{ iff. } x \vee y = y \Leftrightarrow A \subseteq B \text{ iff. } A \cup B = B$$

$$x \leq y \text{ iff. } x \wedge y = x \Leftrightarrow A \subseteq B \text{ iff. } A \cap B = A$$

$$x \vee 0 = x, x \wedge 0 = 0 \Leftrightarrow A \cup \phi = A, A \cap \varphi = \phi$$

$$x \vee 1 = 1, x \wedge 1 = x \Leftrightarrow A \cup S = S, A \cap S = A$$

and more!

Proof of Non-Boolean Algebra

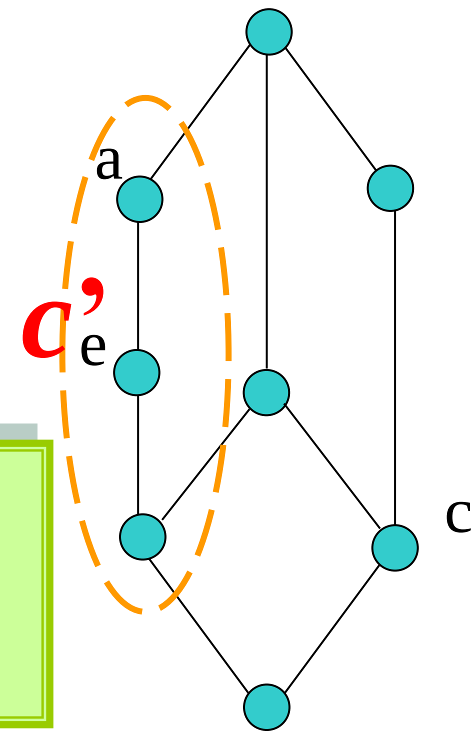
- For a given poset, if any of the formula satisfied by set operations can't be satisfied, the poset ***is not*** a Boolean algebra.

For $\rho(S)$, every element A has a unique complement $\sim A$, such that:

$$A \cup \sim A = S \text{ and } A \cap \sim A = \phi$$

For L, every element x has a unique complement x', such that:

$$x \vee x' = 1 \text{ and } x \wedge x' = 0$$



Boolean Polynomials

- $x_1, x_2, \ldots x_n$ are all Boolean polynomials (expressions).
- The symbols 0 and 1 are Boolean Polynomials.
- If $p(x_1, x_2, \ldots x_n)$ and $q(x_1, x_2, \ldots x_n)$ are two Boolean polynomials, then so are:

$$p(x_1, x_2, \ldots x_n) \vee q(x_1, x_2, \ldots x_n)$$

$$p(x_1, x_2, \ldots x_n) \wedge q(x_1, x_2, \ldots x_n)$$

$$(p(x_1, x_2, \ldots x_n))'$$

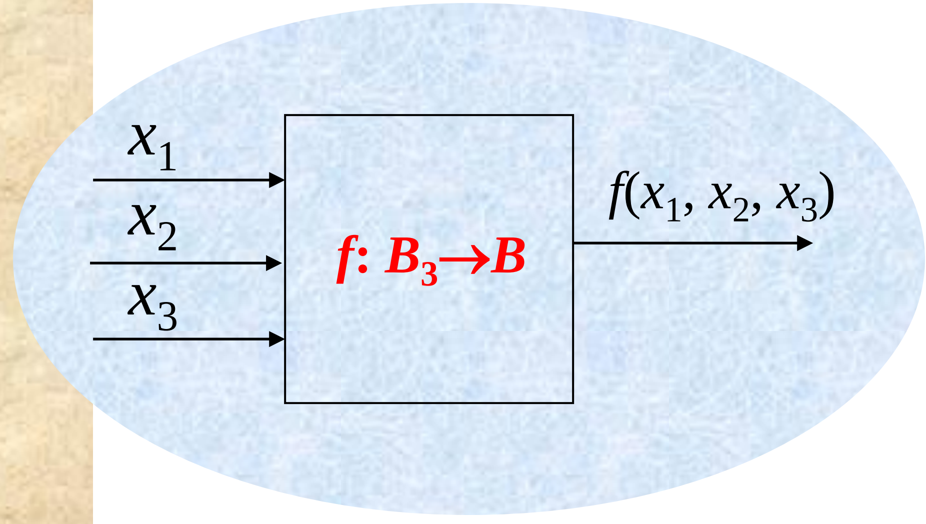
- There are no Boolean polynomials in the variables x_k other than those that can be obtained by repeated use of the rules above.

Interpreting Boolean Polynomials

- Boolean polynomials may be interpreted as representing Boolean computations with unspecified elements of *B*, that is, with 0's and 1's.
- Boolean polynomials are subject to the rules of Boolean algebra.
- Two Boolean polynomials are considered equivalent if one can be turned into the other with Boolean manipulations.
  - Or equivalently, two Boolean polynomials are equivalent if they have the truth tables with the same structure.

Truth Table

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



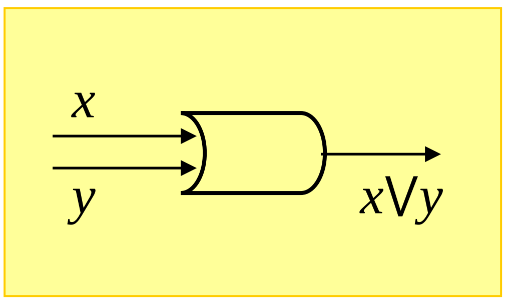
Truth Table: an Example

$p(x_1,x_2,x_3) = (x1 \wedge x2) \vee (x1 \vee (x2' \wedge x3))$

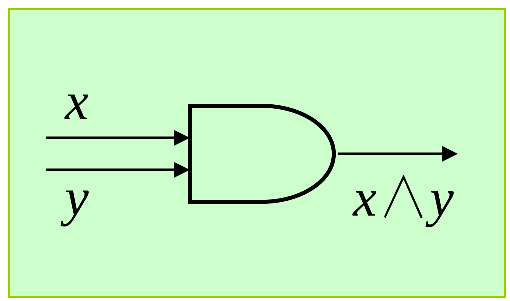
x_1	x_2	x_3	$(x_1 \wedge x_2) \vee (x_1 \vee (x_2' \wedge x_3))$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Logic Diagrams for Boolean Polynomials

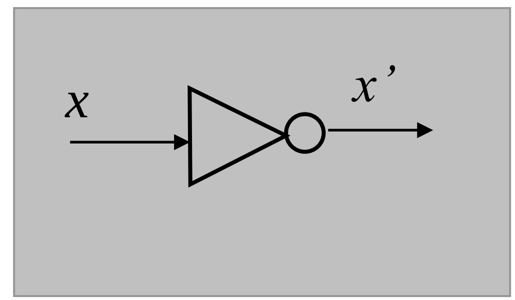
Basic components:



or gate



and gate

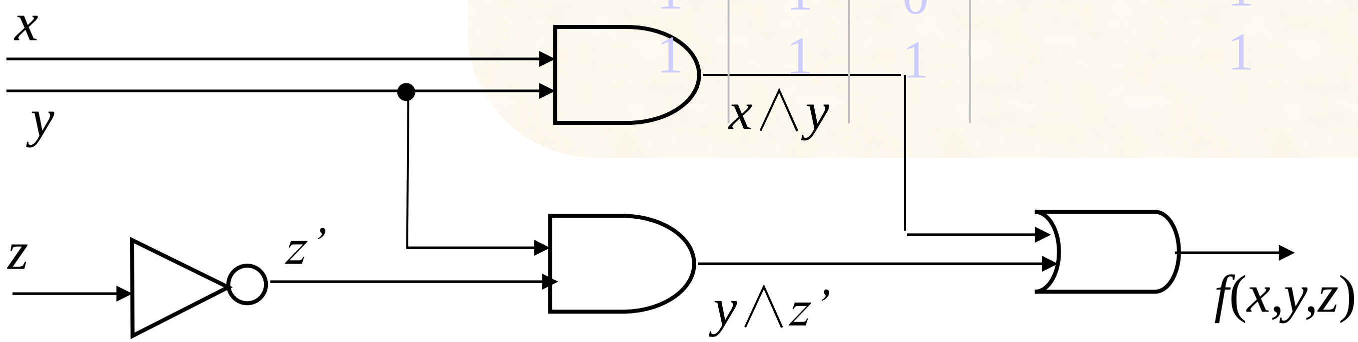


inverter

Logic Diagrams for Boolean Polynomials

x	y	z	$(x \wedge y) \vee (y \wedge z')$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$f(x,y,z) = (x \wedge y) \vee (y \wedge z')$



Subset of B_n Mapping to 1

- If f:$B_n \to B$, define $S(f)=\{b | b \in B_n$, and $f(b)=1\}$, then, for three functions from B_n to B, f, f_1, f_2, we have:
  - If $S(f)=S(f_1) \cup S(f_2)$, the $f(\mathrm{b})=f_1(b) \vee f_2(b)$ for all b in B_n.
  - If $S(f)=S(f_1) \cap S(f_2)$, the $f(\mathrm{b})=f_1(b) \wedge f_2(b)$ for all b in B_n.
- Proof:

the line in truth table with value 1

  - For any b in B_n, if $b \in S(f)$, then $f(b)=1$. Either b is in $S(f_1)$ or in $S(f_2)$, or both. In either cases $f_1(b) \vee f_2(b)=1$.
  - On the other hand, if $b \notin S(f)$, then $f(b)=0$. Since neither $b \in S(f_1)$ nor $b \in S(f_2)$, so, $f_1(b) \vee f_2(b)=0$.
  - Thus, for all $b \in B_n$, $f(\mathrm{b})=f_1(b) \vee f_2(b)$.
  - Same for the second part.

Minterm

x	y	$f(x,y)$
0	0	0
0	1	1
1	0	0
1	1	0

x'∧y

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

x'∧y∧z'

Minterm expression:

For $b = (c_1, c_2, \ldots, c_n) \in B_n$,

$E_b = \overline{x_1} \wedge \overline{x_2} \wedge \ldots \wedge \overline{x_n}$, where

$\overline{x_k} = x_k$ if $c_k = 1$, $\overline{x_k} = x_k'$ if $c_k = 0$

All Functions Expressible

- Any function f: $B_n \rightarrow B$ can be produced by a Boolean expression
  - **Union of minterms.**
  - Proof:
    - For any given boolean function $f : B_n \rightarrow B$, let S(f) = $\{b_1, b_2, \ldots, b_k\}$
    - For each i=1,2,...,k, define function f_i: $B_n \rightarrow B$, as, $f(b_i)$=1 and $f(b)$=0 for any other b.
    - Then $S(f_i)=\{b_i\}$, so, $S(f)=S(f_1)\cup\ldots\cup S(f_n)$.
    - So, $f = f_1 \vee f_2 \vee \ldots, \vee f_n$, which is produced by the union of all minterms E_{bi}

Karnaugh Map of *f* for *n*=2

$f: B_2 \to B$

Basic positions

00	01
10	11

	y'	y
x'	$x' \wedge y'$	$x' \wedge y$
x	$x \wedge y'$	$x \wedge y$

$f(x,y)=(x' \wedge y') \vee (x' \wedge y)$

x	y	$f(x,y)$
0	0	1
0	1	1
1	0	0
1	1	0

However,we know

f(x,y)=x'!

	y'	y
x'	1	1
x	0	0

Simplifying Using Karnaugh Map

$f: B_2 \rightarrow B$

Basic positions

00	01
10	11

	y’	y
x’	x’∧y’	x’∧y
x	x∧y’	x∧y

$f(x,y)=(x'\wedge y')\vee(x'\wedge y)\vee(x\wedge y')$

x	y	$f(x,y)$
0	0	1
0	1	1
1	0	1
1	1	0

$f(x,y) = x' \vee y'$

	y’	y
x’	1	1
x	1	0

Karnaugh Map with *n*=3

	00	01	11	10
0	0 0 0	0 0 1	0 1 1	0 1 0
1	1 0 0	1 0 1	1 1 1	1 1 0

y' y

	$y' \wedge z'$	$y' \wedge z$	$y \wedge z$	$y \wedge z'$
x'	$x' \wedge y' \wedge z'$	$x' \wedge y' \wedge z$	$x' \wedge y \wedge z$	$x' \wedge y \wedge z'$
x	$x' \wedge y' \wedge z'$	$x \wedge y' \wedge z$	$x \wedge y \wedge z$	$x \wedge y \wedge z'$

z' z

Simplifying 3-Variable Expression

x	y	z	$f(x,y,z)$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

	$y'z'$	$y'z$	yz	yz'
x'	**1**	0	0	**1**
x	**1**	0	**1**	**1**

z'

the expression:

$(x' \wedge y' \wedge z') \vee (x' \wedge y \wedge z') \vee$
$(x \wedge y' \wedge z') \vee (x \wedge y \wedge z') \vee (x \wedge y \wedge z)$

So, $z' \vee (x \wedge y)$

Logic Circuit at Work

- For each try in a contest of weight lifting, it is assumed success only if at least 2 of 3 referees decide it a success. Design a logic circuit for use in the situation.

The function: $f(x,y,z)=1$ iff. there are at least 2 one's in x,y,z

the expression:

$(x'\wedge y\wedge z)\vee (x\wedge y'\wedge z)\vee$
$(x\wedge y\wedge z')\vee (x\wedge y\wedge z)$

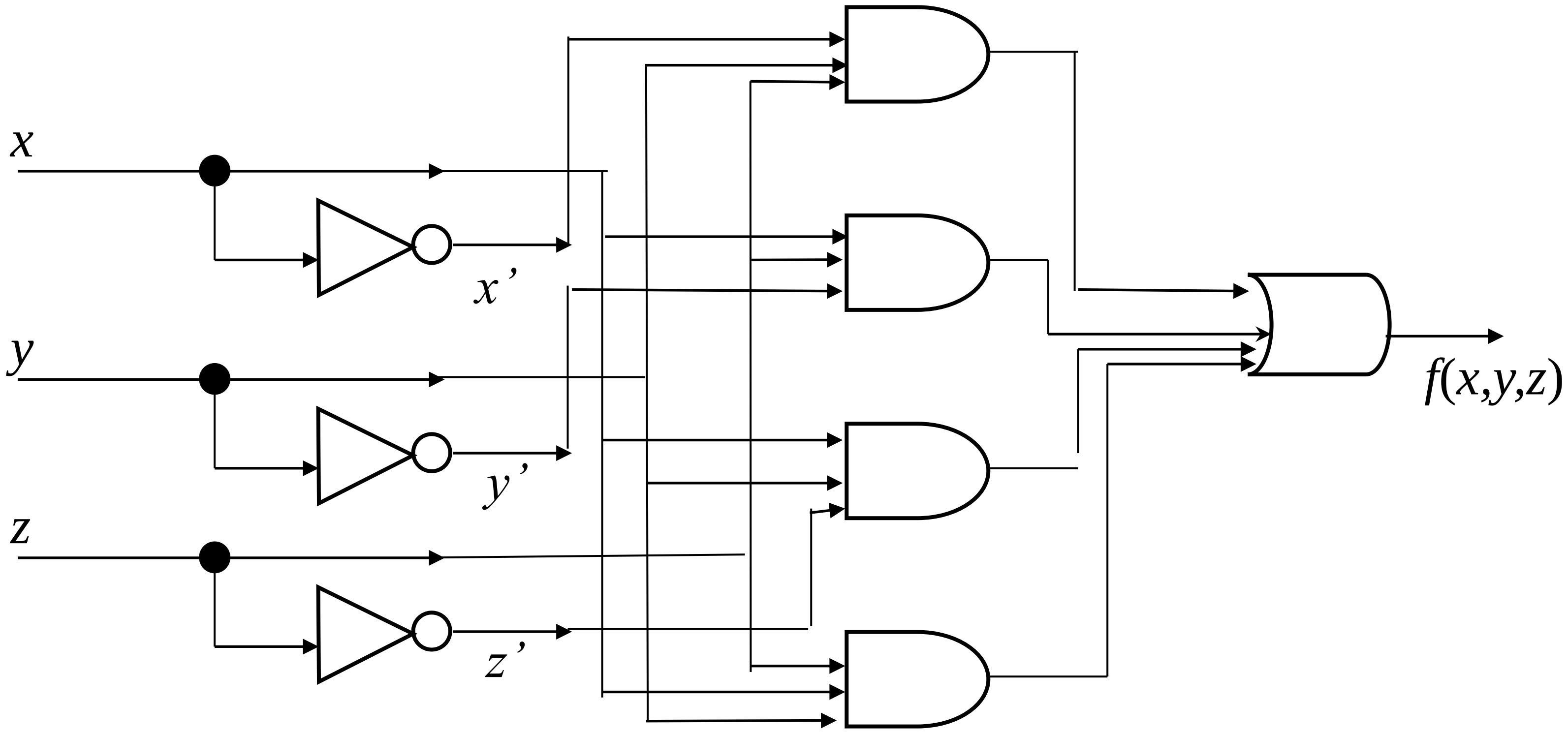
x	y	z	$f(x,y,z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

The Circuit

Too complecated!

the expression:

$(x' \wedge y \wedge z) \vee (x \wedge y' \wedge z) \vee (x \wedge y \wedge z') \vee (x \wedge y \wedge z)$



Make it Simpler

x	y	z	$f(x,y,z)$
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	1
1	1	1	1

	$y'z'$	$y'z$	yz	yz'
x'	0	0	**1**	0
x	0	**1**	**1**	**1**

z'

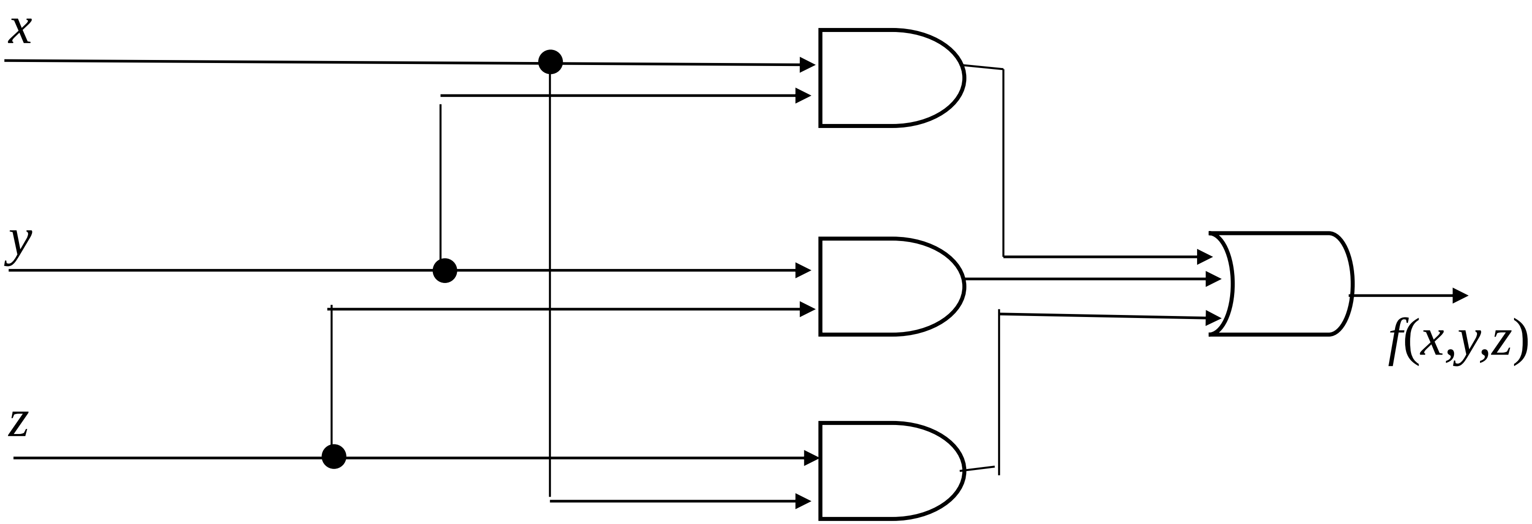
the expression:

$(y \wedge z) \vee (x \wedge z) \vee (x \wedge y)$

Looks Better

the expression:

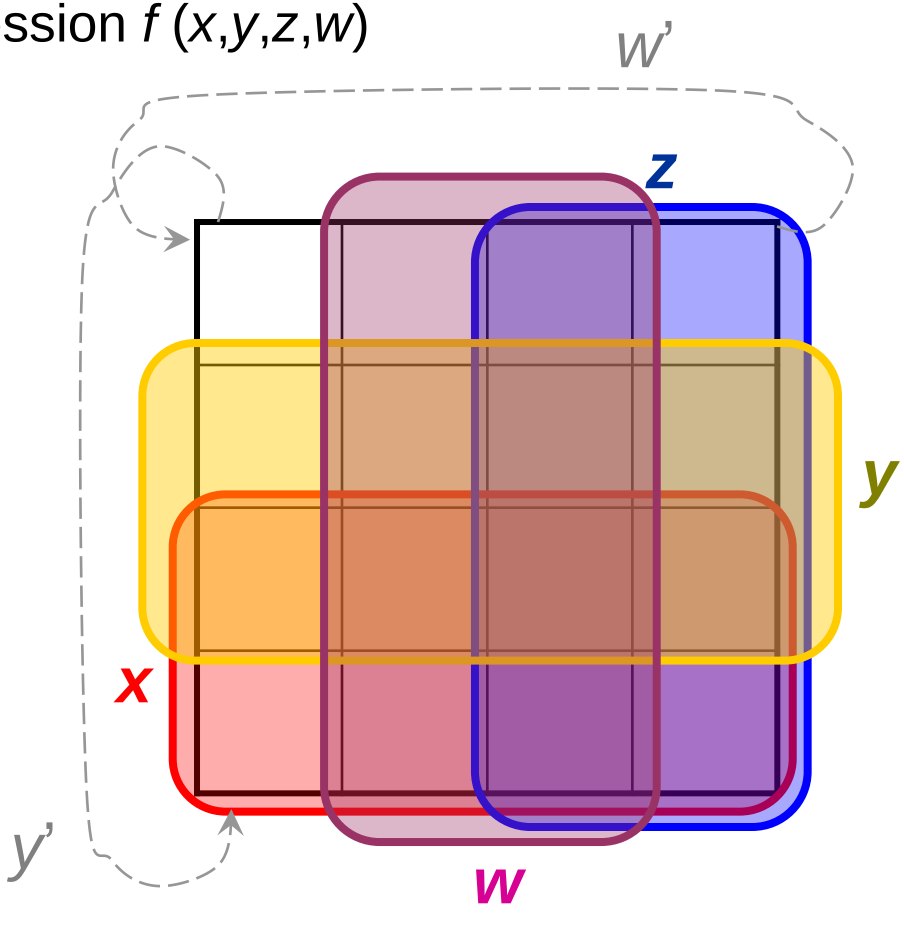
$(y \wedge z) \vee (x \wedge z) \vee (x \wedge y)$



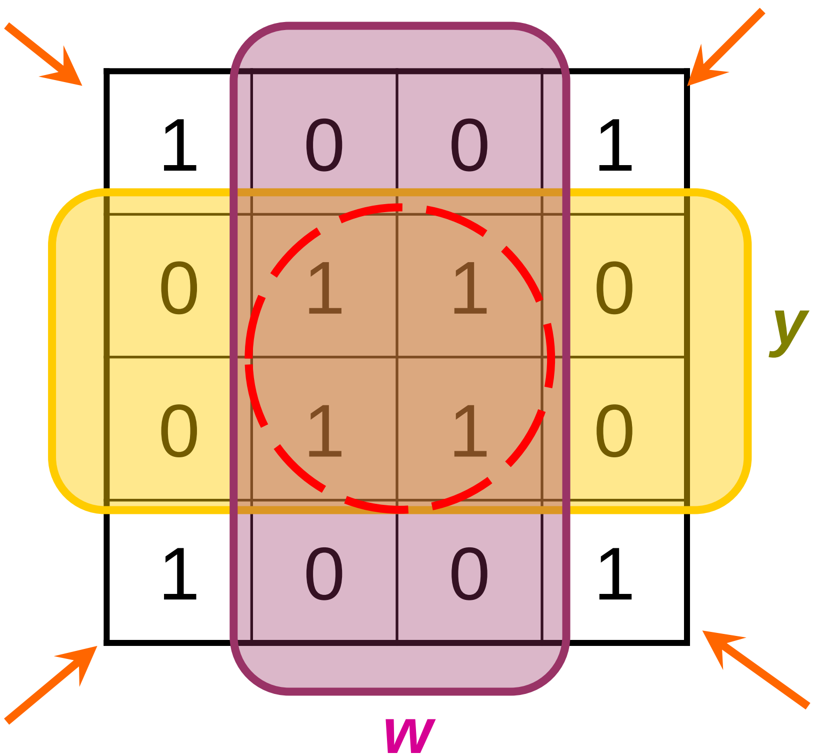
K-map of 4-Variable Expressions

For 4-variable Boolean expression *f* (*x*,*y*,*z*,*w*)

	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1011	1010



An Example

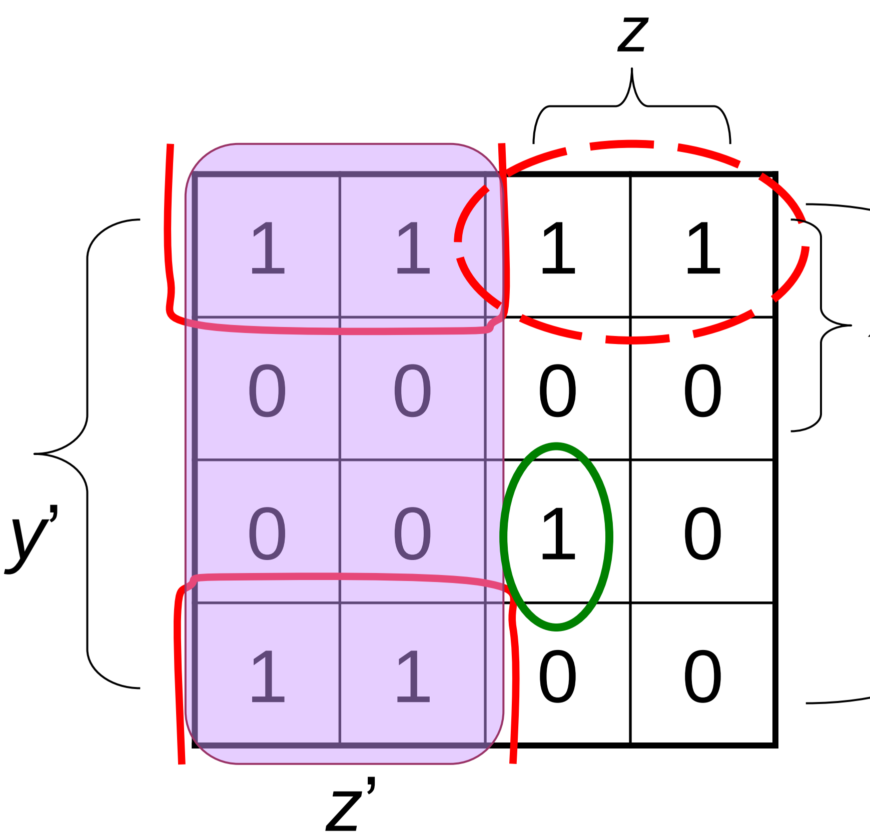


	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1011	1010

$(w \wedge y)$ $(w' \wedge y')$

So, $(w \wedge y) \vee (w' \wedge y')$

Another Example



2×2 square gives 2-literal minterm

$(z' \wedge y')$

1×1 square gives 4-literal minterm

$(x \wedge y \wedge z \wedge w)$

So: the Boolean sum of the three

$\mathbf{(z' \wedge y') \vee (x' \wedge y' \wedge z) \vee (x \wedge y \wedge z \wedge w)}$

1×2 square gives 3-literal minterm

$(x' \wedge y' \wedge z)$

Same, or Different

- The same Boolean function may take different forms, and,…
- The same circuit can implement different Boolean functions, maybe with some exchanges on inputs.

Home Assignments

- To be checked
  - Ex.6.4: 6, 8, 10, 16-21, 27, 29, 32
  - Ex.6.5: 11-14, 18-23
  - Ex.6.6: 8, 12, 14, 16, 24, 25-26

- Experiment 6