Chapter 7

Boolean Algebra

Solve problems using algebra

- 7.1 Finite Boolean Algebras
- 7.2 Function on Boolean Algebras
- 7.3 Laws of Boolean Algebras
- 7.4 Simplification of Boolean Expressions
- 7.5 Use of Karnaugh Map (*K*-map) up to 4 variables

7.1 Finite Boolean Algebras (cont)

- Boolean variables are the variables that can take only value 0 or 1.
F T
- The simplest Boolean algebra consists of set {0, 1} together with the operations of disjunction ($\vee$), conjunction ($\wedge$), and negation (’).
OR AND
NOT

$\bar{p} \vee q \wedge r$

Theorem 1

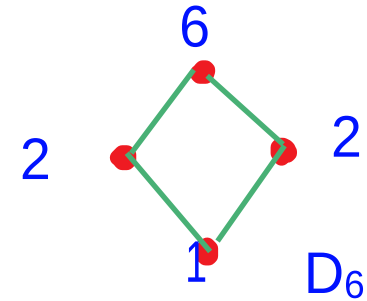
If $S_1 = \{x_1, x_2, \ldots, x_n\}$ and $S_2 = \{y_1, y_2, \ldots, y_n\}$ are any two finite sets with n elements, then the lattices $(P(S_1), \subseteq)$ and $(P(S_2), \subseteq)$ are isomorphic. Consequently, the Hasse diagrms of these lattices may be drawn identically.

Example 1:

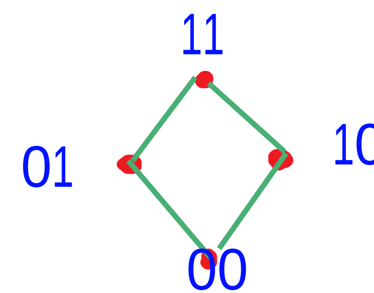
Let $A = \{1, 2, 3, 6\}$ and considered the lattice D_6 consisting of all positive integer divisors of 6 under the partial order of divisibility.

D_6 is isomorphic with B_2. In fact, $f : D_6 \to B_2$ is an isomorphism, where

$f(1) =$ 00', $f(2) =$ 01', $f(3) =$ 10', $f(6) =$ 11



D_6



B_2

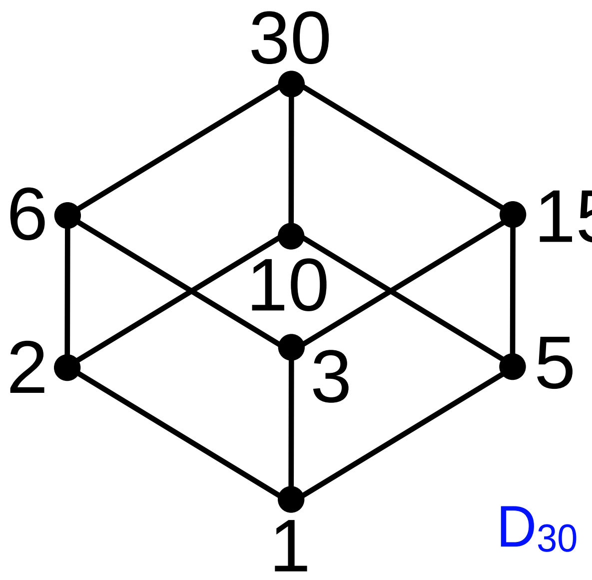
A finite lattice is called a Boolean algebra if it is isomorphic with B_n for some nonnegative integer n. Thus each B_n is a Boolean algebra and so is each lattice $(P(S), \subseteq)$, where S is a finite set.

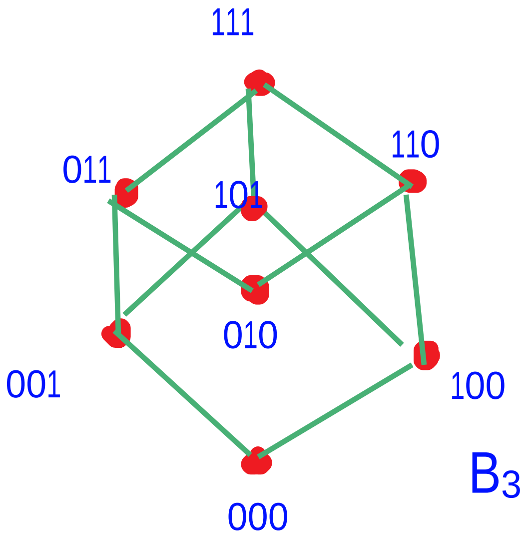
Example 2:

Consider the lattices D_{20} and D_{30} of all positive integer divisors of 20 and 30, respectively, under the partial order of divisibility.

Since D_{20} has six elements and $6 \neq 2^n$ for any integer $n \geq 0$, we conclude that D_{20} is not a Boolean algebra.

The poset D_{30} has eight elements, and since $8 = 2^3$, it could be a Boolean algebra.





By comparing the diagrams above, we see that D_{30} is isomorphic with B_3. In fact, $f : D_{30} \to B_3$ is an isomorphism, where

$f(1) = 000$, $f(2) = 001$, $f(3) = 010$, $f(5) = 100$
$f(6) = 011$, $f(10) = 101$, $f(15) = 110$, $f(30) = 111$
Thus D_{30} is a Boolean algebra.

7.2 Function on Boolean Algebras

- A Boolean function of the n Boolean variables, $x_1, x_2, \ldots, x_n$, is a function $f : B^n \rightarrow B$ such that $f(x_1, x_2, \ldots, x_n)$ is a Boolean expression.
- Boolean function can be expressed in an equivalent standard form, called the disjunctive normal form (sum of products).

$B=\{0,1\}$

Example 3:

Construct a truth table for the Boolean function $f : B^3 \rightarrow B$ where

$f(\overbrace{x, y, z}^{3}) = (x \wedge y) \vee (x' \wedge z)$.

Then write the disjunctive normal form for $f(x, y, z)$.

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

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

7.3 Laws of Boolean Algebras

(Will be given in exams)

1. $(x')' = x$	Involution Property
2. $(x \wedge y)' = x' \vee y'$ $(x \vee y)' = x' \wedge y'$	De Morgan's Laws
3. $x \wedge y = y \wedge x$ $x \vee y = y \vee x$	Commutative Laws

7.3 Laws of Boolean Algebras

4. $(x \wedge y) \wedge z = x \wedge (y \wedge z)$ $(x \vee y) \vee z = x \vee (y \vee z)$	Associative Laws
5. $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$	Distributive Laws
6. $x \wedge x = x$ $x \vee x = x$	Idempotent Laws

7.3 Laws of Boolean Algebras

7. $x \wedge (x \vee y) \equiv x$ $x \vee (x \wedge y) = x$	Absorption Laws
8. $x \wedge x' = 0$ $x \vee x' = 1$	Inverse Laws
9. $x \wedge 0 = 0$ $x \vee 1 = 1$	Dominance Laws
10. $x \wedge 1 = x$ $x \vee 0 = x$	Identity Laws

- The following table summarises the correspondence between Boolean operations, the logical operators in propositional calculus and set operations:

Logical Operation	Set Operation	Boolean Operation
not, ~	—	'
or, $\vee$	$\cup$	$\vee$
and, $\wedge$	$\cap$	$\wedge$

7.4 Simplification of Boolean Expressions

Example 4:

LHS RHS

Show that $(x' \wedge y)' \wedge (x \vee y) = x$.

$$
\begin{aligned}
LHS &= (x' \wedge y)' \wedge (x \vee y) \\
&= (\boldsymbol{x} \vee y') \wedge (\boldsymbol{x} \vee y) \\
&= x \vee (y' \wedge y) \\
&= x \vee 0 \\
&= x = RHS
\end{aligned}
$$

- To find a ‘simpler’ equivalent equation, i.e. use fewer symbols than the original expression.
- Applied to disjunctive normal form of the given expression.

- **Example 5:** Usually in Question 1

Simplify the expression below.

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

$$
\begin{aligned}
&= z \wedge [(\boldsymbol{x'} \wedge y') \vee (\boldsymbol{x'} \wedge y) \vee (x \wedge y')] \\
&= z \wedge [x' \wedge (y' \vee y) \vee (x \wedge y')] \\
&= z \wedge [(x' \wedge 1) \vee (x \wedge y')] \\
&= z \wedge [x' \vee (x \wedge y')] \\
&= z \wedge [(\boldsymbol{x'} \vee x)\,(\boldsymbol{x'} \vee y')] \\
&= z \wedge [1\,(x' \vee y')] \\
&= z \wedge (x' \vee y')
\end{aligned}
$$

is there
a shortcut?

7.5 Use of Karnaugh Map (*K*-map) up to 4 variables Usually in last Q

- A Karnaugh map is a device invented as an aid to logical circuit design that uses a visual display to simplify a sum-of-product Boolean expression by indicating which pairs of minterms can be brought together and merged into a single simpler expression.

Once we get logical expression:

1. Work out truth table
2. Find PDNF
3. Get Karnaugh Map
4. Get simplest form

7.5 Use of Karnaugh Map (*K*-map) up to 4 variables

- Given a sum-of-product Boolean expression.
  - If two variables *x* and *y* are involved, a Karnaugh map consists of a table with two rows and two columns. Each cell corresponds to a minterm. During exam, empty table given to fill in

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

00	01
10	11

7.5 Use of Karnaugh Map (*K*-map) up to 4 variables

- A rectangle of size 2 × 4 is used for three variables *x*, *y*, and *z* in a Boolean expression.

	y'	y'	y	y
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	z	z'

	00	01	11	10
0	000	001	011	010
1	100	101	111	110

7.5 Use of Karnaugh Map (*K*-map) up to 4 variables

- *K*-map with four variables, *w*, *x*, *y*, and *z*.

	z'	z'	z	z	
x'					y'
x'					y
x					y
x					y'
	w'	w	w	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

7.5 Use of Karnaugh Map (*K*-map) up to 4 variables

- For a given Boolean expression in disjunctive normal form we write a one in each of the box representing minterms which appear.
- In a *K*-map, two cells are adjacent if their minterms differ in only one variable.
- The required simplification is to group the 1's in adjacent cells and the variable that appears in pair will be eliminated.

7.5 Use of Karnaugh Map (*K*-map) up to 4 variables

- Steps:

1) Make rectangles around groups of one, two, four or eight 1s.

  - All of the 1s in the map should be included in at least one rectangle.
  - Do not include any of the 0s.

7.5 Use of Karnaugh Map (*K*-map) up to 4 variables

2) Each group corresponds to one product term. For the simplest result:

- Make as few rectangles as possible, to minimize the number of products in the final expression.
- Make each rectangle as large as possible, to minimize the number of literals in each term.
- It's all right for rectangles to overlap, if that makes them larger.

7.5 Use of Karnaugh Map (*K*-map) up to 4 variables

- May have more than one valid solution.

Example 6:

Simplify the following Boolean expressions using Karnaugh map.

i. $(x \wedge y) \vee (x' \wedge y) \vee (x' \wedge y')$ =y + x'

	y	y'
x	1	0
x'	1	1

Note: overlap to make them larger

Check

$$
\begin{aligned}
(x \wedge y) \vee (x' \wedge y) \vee (x \wedge y') &= [y \wedge (x \vee x')] \vee (x' \wedge y') \\
&= [y \wedge 1] \vee (x' \wedge y') \\
&= y \vee (x' \wedge y') \\
&= (y \vee x')\,(y \vee y') \\
&= (y \vee x') \wedge 1 \\
&= y \vee x'
\end{aligned}
$$

Example 6:(cont)

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

	y'	y'	y	y
x'	0	0	0	1
x	0	1	1	0
	Z'	Z	Z	Z'

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

Example 6: (cont)

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

	y'	y'	y	y
x'	1	0	1	1
x	0	0	1	1
	Z'	Z	Z	Z'

Typical steps:

1. Expression

Truth Table

PDNF

KMap

Simplest form

Example 7:

Simplify the Boolean expression

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

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

$$
\begin{aligned}
PDNF &= (x' \wedge y' \wedge z') \vee (x' \wedge y' \wedge z) \vee (x \wedge y' \wedge z') \\
&= x'y'z' + x'y'z + xy'z'
\end{aligned}
$$

$f(x, y, z) =$ $= x'y'z' + x'y'z + xy'z'$

$$\equiv x'y' + y'z'$$

Note: cannot be simplified further.

You can overlap, but you cannot include 0

	y'	y'	y	y
x'	1	1	0	0
x	1	0	0	0
	z'	z	z	z'

Example 8:

i) Refer to the given Karnaugh maps, simplify the Boolean expressions.

	z'	z'	z	z	
x'	1	0	0	1	y'
x'	0	1	1	0	y
x	0	1	1	0	y
x	1	0	0	1	y'
	w'	w	w	w'	

yw+y'w'

Example 8:(cont)

ii)

	z'	z'	z	z	
x'	1	1	1	1	y'
x'	0	0	0	0	y
x	0	0	1	0	y
x	1	1	0	0	y'
	w '	w	w	w '	

$$\boxed{x'y'} + \boxed{y'z'} + \boxed{xyzw}$$