Boolean Algebra

- **Boolean algebra**
  - Boolean operations
  - Laws and identities of Boolean algebra
  - Simplifying Boolean expressions
- **Switching Algebra**
  - Logic gates
  - Designing logic circuits

Boolean Algebra

- A **Boolean algebra** is a **set B** = {a, b, …} and the **binary operators +** and **·** , together with the **unary operator '**, that satisfies the following:

  - *Axiom 1* **Closure**
    - a+b, a·b, a' ∈ B
  - *Axiom 2* **Commutative Law**
    - a+b = b+a
    - a·b = b·a
  - *Axiom 3* **Distributive Law**
    - a·(b + c) = a·b + a·c
    - a+(b·c) = (a + b)·(a + c)

Boolean Algebra

- A **Boolean algebra** is a **set B** = {a, b, …} and the **binary operators +** and **·** , together with the **unary operator '**, that satisfies the following:

  - *Axiom 4* **Identity Law**
    - a+0 = a
    - a·1 = a where 0 and 1 are distinct and unique
  - *Axiom 5* **Complementary Law**
    - a+a' = 1 where a' is unique
    - a·a' = 0

Boolean Algebra

- **Boolean operations**
  - **Boolean complement (')**
    - 0' = 1
    - 1' = 0
  - **Boolean sum (+)**
    - 0+0 = 0
    - 0+1 = 1
    - 1+0 = 1
    - 1+1 = 1
  - **Boolean product (·)**
    - 0·0 = 0
    - 0·1 = 0
    - 1·0 = 0
    - 1·1 = 1

Boolean Algebra

- **Laws of Boolean Algebra**
  - Theorem 1: **Associative**
    - a + (b + c) = (a + b) + c
    - a·(b·c) = (a·b)·c
  - Theorem 2: **Idempotency**
    - a+a = a
    - a·a = a
  - Theorem 3: **Dominance**
    - a+1=1
    - a·0 = 0

Boolean Algebra

- **Laws of Boolean Algebra**
  - Theorem 4: **Double Complement**
    - (a')' = a
  - Theorem 5: **Absorption**
    - a + a·b = a
    - a·(a + b) = a
  - Theorem 6: **De Morgan's**
    - (a+b)' = a'·b'
    - (a·b)' = (a' + b')

Boolean Algebra

- **Identities**:
  - *Identity 1:* a(a' + b)
  - *Identity 2:* a + a'b = a + b
  - *Identity 3:* ab(a + b) = ab
  - *Identity 4:* (ab)' (a + b) = ab' + a'b
  - *Identity 5:* (ab' + a'b')' = ab + a'b'
  - *Identity 6:* (a + b)(b+c)(a+c) = ab + bc + ac
  - *Identity 7:* (a + b)(a' + c) = ac + a'b
  - *Identity 8:* ac + ab + bc' = ac + bc'
  - *Identity 9:* (a + b)(b + c)(a' + c) = (a + b)(a' + c)

Boolean Algebra

- If an algebra A is a **Boolean algebra**, then we can summarize this as
  A = (B = { a, b, … }; +, ·, '; 0, 1)

Examples:
- Set algebra: **(P(U); ∪, ∩, '; ∅, U)**
- Logic: **(B, ∨, ∧, ', False, True)**
  where B is a set of propositions.

Boolean Algebra

- The **dual** of a statement involving Boolean expressions is obtained by replacing:
  - 0 with 1
  - \+ with ·
  - 1 with 0
  - · with +

Examples:
What is the dual of each of the following?
- **(a + b)'** = **a'b'**
- **(x+y)(x+1)** = **x + xy + y**

Boolean Algebra

- **The Duality Principle**
  The dual of a theorem in Boolean algebra is also a theorem.

- **Hierarchy of Boolean Operators**
  (highest to lowest) **'**, **·** , **+**

- **Simplifying Boolean Expressions**
  - Reduce a Boolean expression into one that uses the minimal number of operations
    - *Apply axioms, theorems and identities*
  - Important step in designing logic circuits

Boolean Algebra

- **Simplifying Boolean Expressions**
  Example:
  Simplify the expression **ab + a'b + ab'.**
  Solution:
  ab + a'b + ab'

  = b + a

Boolean Algebra

- **Simplifying Boolean Expressions**
  Example:
  Simplify the expression **ab + a'c + bcd.**
  Solution:
  ab + a'c + bcd

  = ab + a'c

Boolean Algebra

- **Other examples/exercises:**
  1) [(a + b’) + c(a + b’)]
  2) a’b’ + ab + a’b
  3) (a + b)’(a’ + b’)
  4) ab + (b’d)’ + (c + d’)
  5) ab + (a + b)c’ + b
  6) a + b (a’ + b + c)’
  7) cd + ab + d + [ad(bc + ad)]

Switching Algebra

- A **switching function** is a *binary-valued* function of one or more binary-valued variables.
- Given $x_1, x_2, \ldots, x_n$ where $x_i = 0$ or 1 then the switching function S is defined as
  S: $(x_1 \times x_2 \times \ldots \times x_n) \rightarrow \{0,1\}$

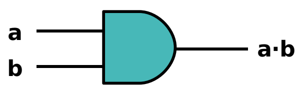
Example:
Define switching function **f(x,y,z) = xy + (y'z)'**.
What is f(x,y,z) when x, y = 0 and z = 1?
f(0,0,1)

Switching Algebra

- **Basic Gates**
  - **AND gate**

Truth table:

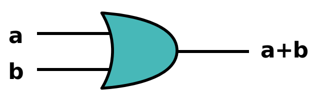
a	b	a·b
0	0	0
0	1	0
1	0	0
1	1	1



  - **OR gate**

Truth table:

a	b	a + b
0	0	0
0	1	1
1	0	1
1	1	1

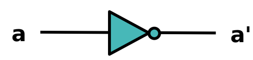


Switching Algebra

- **Basic Gates** (cont'd)
  - **NOT gate**

Truth table:

a	a'
0	0
1	0

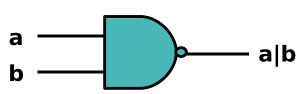


Switching Algebra

- **Other Gates**
  - **NAND (*not* AND) gate**

Truth table:

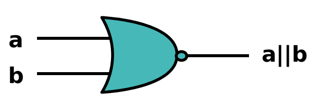
a	b	a|b
0	0	1
0	1	1
1	0	1
1	1	0



  - **NOR (*not* OR) gate**

Truth table:

a	b	a||b
0	0	1
0	1	0
1	0	0
1	1	0

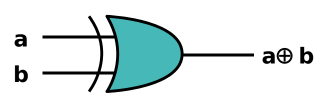


Switching Algebra

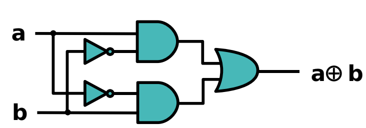
- **Other Gates** (cont'd)
  - **XOR (Exclusive OR) gate**

Truth table:

a	b	a ⊕ b
0	0	0
0	1	1
1	0	1
1	1	0



Note: **a ⊕ b = a'b + ab'**



Switching Algebra

- ***Examples***:
  - Draw the combinational circuit for
    - **f(a,b,c) = a + (bc)'**
    - **f(a,b) = a' + ab' + b**

Designing Logic Circuits

- **CASE 1: Given:** Switching function
  - *Step 1*: Simplify the function.
  - *Step 2*: Draw the corresponding logic diagram

Examples/Exercises:

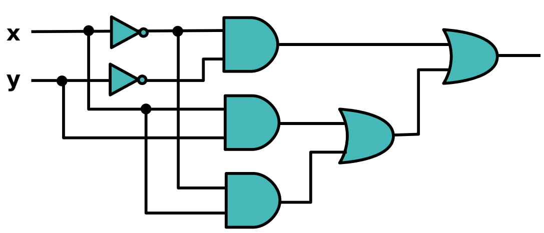
Design logic circuits for the following switching f'ns.

1) f(x,y) = x'y' + xy + x'y.
2) f(x,y) = x' + y'.
3) f(x,y,z) = (xy + z)'.
4) f(a,b) = (a + b)'(a' + b').
5) f(a,b,c) = ab + bc(b + c).
6) f(a,b,c) = a + b(a + c) + ac.

Designing Logic Circuits

- ***Example:***
  Design a logic circuit for **f(x,y) = x'y' + xy + x'y**

Note that this switching function has the following corresponding logic diagram:



Designing Logic Circuits

- ***Example:***
  Design a logic circuit for **f(x,y) = x'y' + xy + x'y**

Solution:

Step 1: Simplify switching function

x'y' + xy + x'y

= x' + y

Designing Logic Circuits

- ***Example:***
  Design a logic circuit for **f(x,y) = x'y' + xy + x'y**

Solution:

Step 2: Draw logic diagram

Since **x'y' + xy + x'y = x' + y**

Designing Logic Circuits

- **CASE 2: Given:** general description of the switching function
  - *Step 1*:**Tabulate** the function for all possible input values based on the description of the function.
    - For each output =1, write a conjunct of all input variables x_i such that
      - if $\mathbf{x_i = 0}$, use $\mathbf{x_i'}$
      - if $\mathbf{x_i = 1}$, use $\mathbf{x_i}$
  - *Step 2*: **Connect** all the conjuncts obtained in the previous step as disjuncts.
  - *Step 3*: **Simplify** the expression if possible.
  - *Step 4*: **Draw** the corresponding logic diagram.

Designing Logic Circuits

- ***Example:***
  - Design a logic circuit for the **majority function**
    - *Input*: accepts three bits
    - *Output*: if majority of the input bits are equal to 1, then the function returns a 1.
  - We may define the majority function M thus:

$$M(a,b,c)=\begin{cases} 1 & \text{if at least two of } a,b,c \text{ are } 1 \\ 0 & \text{otherwise} \end{cases}$$

Designing Logic Circuits

- ***Example:***
  - Design a logic circuit for the **majority function**

$$M(a,b,c)=\begin{cases} 1 & \text{if at least two of } a,b,c \text{ are } 1 \\ 0 & \text{otherwise} \end{cases}$$

Step 1: Tabulate the function …

a	b	c	M(a,b,c)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Designing Logic Circuits

- ***Example:***
  - Design a logic circuit for the **majority function**

$$M(a,b,c)=\begin{cases} 1 & \text{if at least two of } a,b,c \text{ are } 1 \\ 0 & \text{otherwise} \end{cases}$$

Step 1: … and write a conjunct whenever output = 1

a	b	c	M(a,b,c)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Designing Logic Circuits

- ***Example:***
  - Design a logic circuit for the **majority function**

$$M(a,b,c)=\begin{cases} 1 & \text{if at least two of } a,b,c \text{ are } 1 \\ 0 & \text{otherwise} \end{cases}$$

Step 2: Connect the conjuncts as disjuncts

a	b	c	M(a,b,c)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Designing Logic Circuits

- ***Example:***
  - Design a logic circuit for the **majority function**

$$M(a,b,c)=\begin{cases} 1 & \text{if at least two of } a,b,c \text{ are } 1 \\ 0 & \text{otherwise} \end{cases}$$

Step 3: Simplify the expression …

a'bc + ab'c + abc' + abc

= bc + a(b ⊕ c)

Designing Logic Circuits

- ***Example:***
  - Design a logic circuit for the **majority function**

$$M(a,b,c)=\begin{cases} 1 & \text{if at least two of } a,b,c \text{ are } 1 \\ 0 & \text{otherwise} \end{cases}$$

Step 4: Draw the corresponding logic diagram.

bc + a(b ⊕ c)

UP NEXT on CMSC 56 ...

- **Read up on ...**
  - **Matrix/Linear Algebra**
    - What is a **matrix**?
    - What is a **square matrix**?
    - What is a **diagonal matrix**?
    - What is a **scalar matrix**?
    - How do you ...
      - **add** one matrix to another?
      - **multiply** a matrix by another?
      - get the **transpose** of a matrix
      - multiply a matrix by a **scalar**?