CS 225: Switching Theory

S. Tripathy
IIT Patna

Previous Class

Switching Algebra

This Class

- Switching Algebra
-
  - Switching circuit
  - Propositional calculus

Functional Properties

Let binary constant a_i be the value of function f(x1, x2, …, xn) for the combination of variables whose decimal code is i. Thus,

$$f(x1, x2, \ldots, xn)= a_0 x_1' x_2' \ldots x_n' + a_1 x_1' x_2' \ldots x_n + \ldots + a_r x_1 x_2 \ldots x_n$$

The coefficient a_i is set to 1 (0) if the corresponding minterm is (is not) in the canonical form

Since there are 2^n coefficients, each of which can have two values, 0 and 1, there are 2^2^n possible switching functions of n variables

Example: Canonical sum-of-products form for two variables

$$f(x,y) = a_0 x'y' + a_1 x'y + a_2 xy' + a_3 xy$$

Thus 2^2^2 =16 functions corresponding to the 16 possible assignments of 0's and 1's to a_0, a_1, a_2, and a_3

List of Functions of Two Variables

a3	a2	a1	a0	f(x,y)	Name of Function	Symbol
0	0	0	0	0	Inconsistency	
0	0	0	1	x'y'	NOR	x↓y
0	0	1	0	x'y		
0	0	1	1	x'	NOT	x'
0	1	0	0	xy'		
0	1	0	1	y'		
0	1	1	0	x'y+xy'	Exclusive OR	x⊕ y
0	1	1	1	x'+y'	NAND	x|y

a3	a2	a1	a0	f(x,y)	Name of Function	Symbol
1	0	0	0	xy	AND	x.y
1	0	0	1	xy+x'y'	Equivalence	x ≡ y
1	0	1	0	y		
1	0	1	1	x'+y	Implication	x⟶y
1	1	0	0	x		
1	1	0	1	x+y'	Implication	y⟶x
1	1	1	0	x+y	OR	x + y
1	1	1	1	1	Tautology	

Functionally Complete Operations

Every switching function can be expressed in canonical form consisting of a finite number of switching variables, constants and operations +, ., '

A set of operations is functionally complete (or universal) if and only if every switching function can be expressed by operations from this set

Example: Set {+, ., '}

Set {+, '} ?

Yes, since using De Morgan's theorem, $x . y = (x' + y')'$.

Thus, + and ' can replace the . in any switching function

Set {., '} Yes for similar reasons

NAND:? Yes, $NAND(x,x) = x'$ and $NAND[NAND(x,y),NAND(x,y)] = xy$

NOR:? Yes, $NOR(x,x) = x'$ and $NOR[NOR(x,y), NOR(x,y)] = x + y$

Isomorphic Systems

Isomorphism: Two algebraic systems are isomorphic if

- For every operation in one system, there exists a corresponding operation in the second system
- To each element xi in one system, there corresponds a unique element yi in the other system, and vice versa
- If each operation and element in every postulate of one system is replaced by the corresponding operation and element in the other system, then the resulting postulate is valid in the second system

Thus, two algebraic systems are isomorphic if and only if they are identical except the labels and symbols used to represent the operations and elements

Transmission Function

Transmission function for a circuit: assumes value 1 (0) when there is (there is not) a path from one terminal of the circuit to the other through which information flows

Definition of transmission functions

x y	x + y	xy
0 0	0	0
0 1	1	0
1 0	1	0
1 1	1	1

Analogy: OR <-> parallel; AND <-> series

Complement of a given circuit: one that blocks all paths of information flow whenever the given circuit permits any

Thus, algebraic system for switching circuits isomorphic to switching algebra

Propositional Calculus

Proposition: declarative statement which may be either true or false

Example: temperature is 100 degree Celsius
turtle runs faster than the hare
sum of 2 and 3 equals 5

Proposition variable: 1 (0) if proposition is true (false)

Negation p' of proposition p: 1 (0) if p is 0 (1)

Conjunction of propositions p and q is pq: true when both p and q are true and false whenever either one or both p and q are false

Disjunction of propositions p and q is p + q: true when either p or q or both are true and false whenever both p and q are false

Propositional Calculus (Contd.)

Definition of conjunction and disjunction of p and q

p q	pq	p + q
0 0	0	0
0 1	0	1
1 0	0	1
1 1	1	1

- **Analogy**: OR <-> disjunction; AND <-> conjunction

- Thus, algebraic system for switching circuits is isomorphic to propositional calculus

Propositional Calculus Example

Example: Air-conditioning of a storage warehouse to be turned on if one or more of the following three conditions occurs:

1. Weight of stored material is less than 100 kg, relative humidity is at least 60%, and temperature is above 60 degrees Celsius
2. Weight of stored material is 100 kg or more and temperature is above 60 degrees Celsius
3. Weight of the stored material is less than 100 kg and the barometer stands at 30 inches of mercury or over

Propositional Calculus Example (Contd.)

A: proposition that air-conditioning is turned on

W: weight of 100 kg or more

H: relative humidity of at least 60%

T: temperature above 60 degrees Celsius

P: barometric pressure is 30 inches of mercury or more

$$\begin{aligned} A &= W'HT + WT + W'P \\ &= HT + WT + W'P \\ &= T(H+W) + W'P \end{aligned}$$

Thus, air-conditioning is on if the temperature is above 60 degrees Celsius and either the weight is at least 100 kg or the humidity is at least 60%, or if the weight is less than 100 kg and the barometer stands at 30 inches or over

Electronic Gate Networks

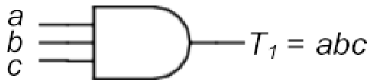
Electronic gates: generally receive voltages as inputs and produce output voltages

Precise values of voltages not significant: restricted to value ranges – high (value 1) and low (value 0)

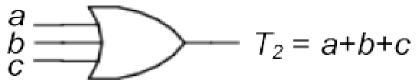
Electronic gates constructed with two-state switching devices: each capable of permitting the flow of current or blocking it

To implement arbitrary switching functions: gates must be able to implement a functionally complete set of operations

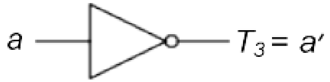
Functionally Complete Set



(*a*) AND gate.

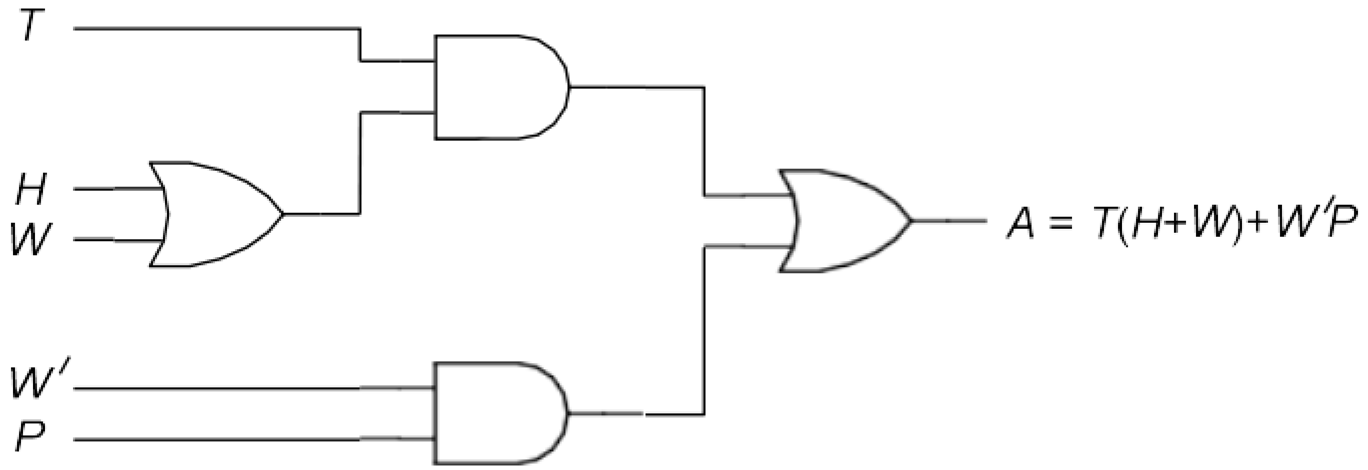


(*b*) OR gate.



(*c*) NOT gate.

Gate Network for Air-conditioning Function



Boolean Algebra

Boolean algebra B: A set of elements a, b, c, …, and binary operations + and . that satisfy the idempotent, commutative, absorption, and associative laws, and are mutually distributive

Complement a′ of any element a in B is unique, i.e., there exists only element a′ such that a + a′ = 1 and a . a′ = 0

- Suppose there exists element a which has two complements, b1 and b2,
  i.e., a + b1 = 1, a . b1 = 0, a + b2 = 1, a . b2 = 0
    - Then b1 = b1 . 1 = b1 . (a + b2) = b1 . a + b1 . b2 = 0 + b1 . b2 = b1 . b2
- Similar arguments show b2 = b1 . b2.
- Thus, b1 = b2, proving the uniqueness of the complement

Complements of elements 0 and 1: since by definition 0 + 0′ = 1, 0′ = 1. Similarly, 1′ = 0

De Morgan’s Theorem

Prove De Morgan’s theorem for two variables:

$$(a + b)' = a' \,.\, b'$$
$$(a \,.\, b)' = a' + b'$$

- We have to show that $(a + b)(a' \,.\, b') = 0$ and $(a + b) + a' \,.\, b' = 1$
- Applying the distributive law:
  - $(a + b) + a'b' = (a + b + a')(a + b + b') = (b + 1)(a + 1) = 1$
- Dual property proved similarly

Definition of a Boolean algebra isomorphic to switching algebra

+	0 1	.	0 1	
0	0 1	0	0 0	$0' = 1$
1	1 1	1	0 1	$1' = 0$

De Morgan’s Theorem

Example of Boolean algebra:

+	**0 1 a b**	**.**	**0 1 a b**	
0	0 1 a b	0	0 0 0 0	$0' = 1$
1	1 1 1 1	1	0 1 a b	$1' = 0$
a	a 1 a 1	a	0 a a 0	$a' = b$
b	b 1 1 b	b	0 b 0 b	$b' = a$

Boolean Algebra Operator Precedence

- Evaluate the following Boolean equations, assuming a=1, b=1, c=0, d=1.
  - Q2. F = ab + c.
    - Answer: first * (. AND) So on.
  - Q3. F = ab'.
    - Answer: we first evaluate b' because NOT has precedence over AND, resulting in F = 1 * (1') = 1 * (0) = 1 * 0 = 0.

a

Boolean algebra precedence, highest precedence first.

Symbol	Name	Description
( ) first	Parentheses	Evaluate expressions nested in parentheses
'	NOT	Evaluate from left to right
*	AND	Evaluate from left to right
+	OR	Evaluate from left to right

- Thanks
- Q2 on Next Week