<u>Discrete Mathematics Scribed Lecture 4 Notes</u>

Date: 08-08-2022

Student ID:

202212016

202212017

202212018

202212019

202212020

Basic Connectives :

There are 3 basic connectives out of which other connectives can be derived. These connectives are ‘OR(∨)’, ‘AND(∧)’ and ‘NOT’(¬, ~). Other derived connectives include XOR, NAND, NOR.

Truth Tables:

A truth table is a mathematical table used to determine if a compound statement is true or false.

NOT GATE Truth Table:

P	**¬P**
0	1
1	0

OR, AND, Implication, Bi-Implication and XOR GATE Truth Table:

P	**Q**	**P ∨ Q**	**P ∧ Q**	**P => Q**	**P <=> Q**	**P ⊕ Q**
0	0	0	0	1	1	0
0	1	1	0	1	0	1
1	0	1	0	0	0	1
1	1	1	1	1	1	0

Parity:

Parity in mathematics works on the principle if the numbers are either odd or even.

1. **$1 \oplus 1 \oplus 1 \oplus 1 = 0$**
   $(1 \oplus 1)$ evaluates to 0, thus the above statement can be simplified to $0 \oplus 0$ which evaluates to 0.
2. **$1 \oplus 1 \oplus 1 = 1$**
   $(1 \oplus 1)$ evaluates to 0, thus the above statement can be simplified to $0 \oplus 1$ which evaluates to 1.

Mathematical Induction:

It is a mathematical proof technique. Here we assume that a statement holds true for P(n), then we prove the statement for P(n+1). It is essentially used to prove that a statement holds true for all natural numbers N >= n.

For Example –

$(P_0 \oplus P_1 \oplus P_2 \oplus P_3 \oplus P_4 \oplus P_5)$

Truth table for this proposition will contain 2^6 = 64 assignments. Rather than finding all the cases we can prove using mathematical induction that half of the cases will evaluate to 1 and half to 0.

Using mathematical induction,

$(P_0 \oplus P_1)$ holds true if ($P_0 = 0$ and $P_1 = 1$) or ($P_0 = 1$ and $P_1 = 0$) i.e. for half of the cases according to it's truth table.

$((P_0 \oplus P_1) \oplus P_2)$ will become true if ($P_0 \oplus P_1 = 0$ and $P_2 = 1$) or ($P_0 \oplus P_1 = 1$ and $P_2 = 0$).

Thus, it holds true for $(P_0 \oplus P_1)$ and we proved that it holds true for $((P_0 \oplus P_1) \oplus P_2)$. Hence by principle of mathematical induction it will hold true for $P_0 \oplus P_1 \oplus \ldots\ldots P(n)$.

Thus, $P_0 \oplus P_1 \ldots\ldots P^{k-1}$ has exactly 2^{k-1} satisfying assignments.

All possible logic gates for propositions can be derived from ‘OR(∨)’, ‘AND(∧)’ and ‘NOT’(¬, ~) gates using these 3 equations –

1. $(P \Rightarrow Q) \equiv ((\neg P) \lor (Q))$.
2. $(P \Leftrightarrow Q) \equiv (P \Rightarrow Q) \land (Q \Rightarrow P)$.
   $\equiv (\neg P \lor Q) \land (\neg Q \lor P)$.
3. $(P \oplus Q) \equiv \neg (P \Leftrightarrow Q)$

Thus, in proposition logic 3 can be replaced by 2 and 2 can be replaced by 1 to get the logic consisting of only 3 basic connectives.

De Morgan’s Law :

1. $\neg(P \lor Q) \equiv (\neg P) \land (\neg Q)$
2. $\neg(P \land Q) \equiv (\neg P) \lor (\neg Q)$

Conjunction Normal Form(CNF) :

$C_1 \land C_2 \land \ldots \land C_k$

where $C_i = l_1 \lor l_2 \ldots \lor l_t$

where $l_j = P_i$ or $\neg P_i$

Disjunction Normal Form (DNF) :

$C_1 \lor C_2 \lor \ldots \lor C_k$

Where, $C_i = l_1 \land l_2 \ldots \land l_t$

Where, $l_j = P_i$ or $\neg P_i$

here l is a literal and P is a proposition.