## CS109B Notes for Lecture 5/10/95

### Why Logic?

- Models reasoning, mathematical proofs, human discourse.
- Its algebra allows us to model and optimize:
  1. Switching circuits, the stuff of which computers and built.
  2. Decision tables, complex choices made as a function of many variables, e.g., "should I buy this stock."
- Appears in programming languages such as Prolog (or even ML to an extent) that are designed to implement programs that reason, plan, strategize, or draw conclusions from data (e.g., automatic medical diagnosis).

### Propositional Logic

- *Constants*: `TRUE` and `FALSE`.
- *Propositional variable* = symbol that represents the truth or falsehood of a "proposition," i.e., a statement about something.
  - ☐ Examples are propositional variable $p$ standing for "it is raining" or variable $q$ standing for "$X < Y + Z$."

### Logical Expressions

Built from operands (constants and variables) and *logical operators*, which are functions with Boolean arguments and result.

Most common operators:

a) `AND`, `OR`,`NOT`: the usual stuff.

b) *Implies*, written $\rightarrow$. $p \rightarrow q$ has value `TRUE` unless $p$ is `TRUE` and $q$ is `FALSE`.

- ☐ Note "$p$ implies $q$" does not mean "$p$ causes $q$." For example, "if $2+2=5$ then the moon is made of cheese" is an expression with truth-value `TRUE`, even though there is nothing about arithmetic that causes the moon to be made of cheese, and in fact the moon is *not* made of cheese.

c) *Equivalence*, written $\equiv$. $p \equiv q$ is true iff $p$ and $q$ have the same truth value.

d) *Symmetric Difference*, written $\oplus$. It is the negation of $\equiv$; $p \oplus q$ is true iff exactly one of $p$ and $q$ is true.

e) *Nand*: $p\texttt{NAND}q$ is true whenever $p\texttt{AND}q$ is false, i.e., if at least one of $p$ and $q$ is false.

f) *Nor*: $p\texttt{NOR}q$ is true whenever $p\texttt{OR}q$ is false, i.e., if both $p$ and $q$ are false.

## Truth Tables

The *truth table* for an expression has one row for each combination of truth-values for its variables, i.e., $2^n$ rows if there are $n$ variables.

- Assignment of `TRUE` or `FALSE` to each variable of the expression is a *truth assignment.*

The value in each row is the value of the expression for that truth assignment.

- Note: 0 and 1 are used for `FALSE` and `TRUE`, respectively.
- Often, we evaluate an expression "bottom-up," with a column for each subexpression.
  - ☐ Apply an operator to two columns by applying the operator row-wise.

**Example:** $p\texttt{AND}\big((q\texttt{NOR}p) \equiv q\big)$.

| $p$ | $q$ | $q\texttt{NOR}p$ | $q\texttt{NOR}p \equiv q$ | whole |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |

## Functions of 2 Variables

There are 16 different logical functions of 2 variables $p$ and $q$.

- Why? Think of a truth table for such a function. There are 4 entries, each of which can be "colored" `TRUE` or `FALSE`.

| $p$ | $q$ | $f(p,q)$ |
|---|---|---|
| 0 | 0 | $w$ |
| 0 | 1 | $x$ |
| 1 | 0 | $y$ |
| 1 | 1 | $z$ |

- A function can be represented by listing the values of $wxyz$.

**Example:** `AND` is 0001, $\equiv$ is 1001.

- Some functions are "trivial," e.g., 0000 is `FALSE` regardless of its arguments,

- Others are "degenerate"; they look only at one argument. e.g., 0011 is the function "$p$."

## Class Problem

A function is *monotone* if when you "increase" one of its arguments, you do not "decrease" the result.

- For example, in arithmetic, $x+y$ is monotone; increasing $x$ or $y$ will not decrease the sum.

- In logic, "increase" means go from `FALSE` to `TRUE`, and "decrease" means go from `TRUE` to `FALSE`.

Question: How many of the 16 logical functions of 2 variables are monotone?

- Hint: What can you say about monotone functions if (a) the 00-row has value 1 or (b) the 11-row has value 0?