**University of Virginia** **cs2102: Discrete Mathematics** 1 September 2016

# Class 4: Logical Formulas

### Schedule

**Problem Set 1** is due **Friday at 6:29pm**.

I will not be able to hold my normal office hours on Monday.

Next week, we will cover the rest of Chapter 3 (Satisfiability and Quantifiers).

## Notes and Questions

### Well-Ordering Principle Proof

**Odd Summation.** (Problem 2.12) Prove that for all $n > 0$, the sum of the first $n$ odd numbers is $n^2$.

### Notations

Mathematics and other domains often use many symbols to mean the same thing. Section 3.2 of the book gives some common notations, but there are others in common use.

| English | Logic | C, Java, Rust | Python |
|---|---|---|---|
| $P$ IMPLIES $Q$ | $P \implies Q$ *or* $P \longrightarrow Q$ | - | - |
| NOT$(P)$ | $\neg P$ *or* $\overline{P}$ | `!p` | `not p` |
| $P$ AND $Q$ | $P \wedge Q$ | `p && q` | `p and q` |
| $P$ OR $Q$ | $P \vee Q$ | `P \|\| Q` | `p or q` |
| $P$ XOR $Q$ | $P \oplus Q$ | `p ^ q` (bitwise) or `p != q` | `p ^ q` |

For what values in Java or C are `p ^ q` and `p != q` both valid, but have different meanings?

## Logical Formulas

| $P$ | NOT$(P)$ | ________ |
|---|---|---|
| **T** | **F** | |
| **F** | **T** | |

How many one-input Boolean operators are there? How many do we need to produce them all?

| $P$ | $Q$ | $P \land Q$ | $P \lor Q$ | $P \implies Q$ | ________ | $P \oplus Q$ |
|---|---|---|---|---|---|---|
| **T** | **T** | **T** | **T** | | **T** | **F** |
| **T** | **F** | | **T** | | **F** | **T** |
| **F** | **T** | | **T** | | **F** | **T** |
| **F** | **F** | | **F** | | **T** | **F** |

How many two-input Boolean operators are there?

**De Morgan's Laws:**

$$\neg(P \land Q) \equiv (\neg P) \lor (\neg Q) \qquad \neg(P \lor Q) \equiv (\neg P) \land (\neg Q)$$

How can these be written without the $\neg$ in front?

Prove that it is possible to make all two-input Boolean operators using just NOT and any *odd* two-input operator. (An operator is *odd*, if the number of outputs that are **True** are odd.)

**Definition: valid.** A logical formula is *valid* if there is no way to make it **false**. That is, no matter what truth values its variables have, it is always **true**. (Another name for this is a *tautology*.)

**Definition: satisfiable.** A logical formula is *satisfiable* if there is *some* way to make it **true**. That is, there is at least one assignment of truth value to its variables that makes the forumla true.

For each of the formulas below, determine if it is *valid* and if it is *satisfiable*.

1. $(P \lor \neg P)$
2. $(P \lor Q) \land (\neg P \lor Q)$
3. $((P \implies Q) \land (Q \implies P)) \lor (P \text{ XOR } Q)$