COT 3100

APPLICATIONS OF DISCRETE STRUCTURES

DR. ALPER UNGOR • SPRING 2016 • UNIVERSITY OF FLORIDA

Last Revision: January 5, 2016

Table of Contents

Abstract

These notes are intended as a resource for myself; past, present, or future students of this course, and anyone interested in the material. The goal is to provide an end-to-end resource that covers all material discussed in the course displayed in an organized manner. If you spot any errors or would like to contribute, please contact me directly.

1 Foundations of Logic: Overview

- Propositional Logic
  - Basic Definitions
  - Equivalence Rules & Derivations
- Predicate Logic
  - Predicates
  - Quantified Predicate Expressions
  - Equivalences & Derivations

1.1 Propositional Logic

Definition 1.1 (Propositional Logic)**.** The logic of compound statements built from simpler statements using *Boolean Connectives.*

Applications:

- Design of digital electronic circuits
- Expressing conditions in computer programs
- Queries to databses & search engines

1.1.1 Basic Definitions

Definition 1.2 (Proposition)**.** A ***proposition (p, q, r, ...)*** is simply a *statement i.e. a declarative sentence) with a definite meaning*, having a *truth value* that's either *true* or *false* (never both, neither, or somewhere in between).

In *probability theory*, we assign *degrees of certainty* to propositions. For now we will just use True/False.

Examples of Propositions

- "It is raining."

Not propositions:

- "Who's there?" (interrogative, question)

Boolean Operators

- Negation (NOT)
- Conjunction (AND)
- Disjunction (OR)
- Exclusive-Or (XOR)
- Implication (IF)
- Bi-conditional (IFF)

Operators & Connectives

- An *operator* or *connective* combines one or more *operand* expressions into a larger expression (e.g. "+" in numeric expressions).
- *Unary* operators take 1 operand (e.g. -3).
  - The unary *negation operator* "¬" (NOT) transforms a proposition into its logical *negation*
    - Example: If p = "I have brown hair.", then ¬ p = "I do not have brown hair."

p	¬ p
T	F
F	T

- *Binary* operators take 2 operands (e.g. 3 x 4).
- *Propositional or Boolean* operators operate on propositions or truth values instead of on numbers.

1.1.2 Equivalence Rules & Derivations

1.2 Predicate Logic

1.2.1 Predicates

1.2.2 Quantified Predicate Expressions

1.2.3 Equivalences & Derivations