College of Computer & Information Science
Northeastern University
Fall 2018

CS5800
Algorithms
11 September 2018

Primer on Proofs

This very brief primer draws material from the text *Mathematics for Computer Science*, by E. Lehman, F. T. Leighton, and A. Meyer, and notes graciously shared by Prof. Neal Young.

1 Propositions, Predicates, Axioms, Deductions, and Proofs

A *proposition* is a statement that is either true or false. A *predicate* is a proposition that depends on one or more variables. For example, for a variable n that represents an integer, we could have a predicate $P(n)$ defined as "n is a prime". Then, $P(n)$ is true if and only if n is prime.

An *axiom* is a proposition that we assume to be true. Our proof system relies on a set of axioms. Indeed, almost all of mathematics that we have developed can be derived from a set of axioms called Zermelo-Fraenkel with Choice (ZFC). For our purposes in this class, we will assume all of the propositions we have proved in elementary and high school mathematics as our axioms.

A *deduction* is a rule for proving new propositions using previously proved ones. Here are three basic deductions.

$$\text{Modus Ponens:}\frac{PP \implies Q}{Q}$$

$$\text{Transitivity:}\frac{P \implies Q, Q \implies R}{P \implies R}$$

$$\text{Contrapositivity:}\frac{\text{not}(P) \implies \text{not}(Q)}{Q \implies P}$$

A *proof* of a proposition is a sequence of logical deductions from axioms and previously proved propositions leading to the desired proposition.

2 Basic Proof Principles

There are several simple proof methods that we will follow to develop our proofs: (a) proving implications ($P \implies Q$); (b) proving by cases; (c) proving by contradiction; and (d) principle of mathematical induction.

Many of our proofs will use induction, both the standard version and what is often referred to as strong induction. It is known, however, that both of these versions are equivalent. We will also often use the *well-ordering principle*, which states that every nonmepty set of nonnegative

Proof principle	How to prove	How to use
A and B	prove A; prove B	assume A, assume B, prove C
A or B	prove A, or prove B	prove by cases: assume A, then prove C; assume B, then prove C
$A \implies B$	assume A, prove B	if A holds, then you can deduce B
$\forall x \in S : P(x)$	let x be an arbitrary element of S; prove $P(x)$	given any $y \in S$, can deduce $P(y)$
$\exists x \in S : P(x)$	prove $P(x)$ for a particular element of S	let y be an element of S for which $P(y)$ holds; use $y \in S$ and $P(y)$ to derive other propositions
not A	proof by contradiction: assume A, derive a contradiction	prove A and not A to derive a contradiction

integers has a smallest element. Any proposition proved by induction can also be proved by the well-ordering principle.

The following table lists six universal logical operators that arise in propositions. The table lists, for each operator, a basic template for proving the associated proposition, and how to use such a proposition to prove other desired propositions.