Proofs - A Brief Introduction

• When we ask you to prove or disprove something on a problem set or test, we expect you to supply us with a proof. A proof is a logical argument, based on a set of known truths or assumptions, that leads unambiguously to the stated conclusion.

• Common mistakes:
  
  • Known truths are false
    
    \[ 2 + 2 = 5 \]
  
  • Supporting truths are invoked without proof.
    
    *Theorem A is true because it’s just a particular case of theorem B.*
  
  • Assumptions which weaken or undermine the original statement are added.
    
    *Theorem A is true for every case I could think of.*
  
• If you cannot complete a proof, do as much as you can and then acknowledge where you got stuck!

  "I couldn’t prove case X” is usually worth more points than ignoring case X or stating that it is obviously true.

  "I couldn’t find a way to prove X without assuming Y” is usually worth more points than just assuming Y without defending this assumption."
Common Proof Techniques

There are too many to list, but here are some of the most frequently used:

- Deductive Proofs - proofs based purely on logical deductions. Most proofs have several deductive steps, but few are completely deductive.

**Thm**: the product of two odd integers is odd.

**Proof**: Let \( x \) and \( y \) be odd integers. From the definition of an odd number, we know that when an odd number is divided by 2, using integer division, the remainder is 1.

Therefore, we can we know that there exist some integers \( i, j \) such that \( x = 2i + 1 \), and \( y = 2j + 1 \).

Therefore, the product of \( xy \) can be written as

\[
(2i +1)(2j + 1) = (4ij + 2i + 2j + 1)
\]

\[
= 2(2ij + i + j) + 1
\]

The result of dividing this result by 2 is \( 2ij + i + j \), remainder 1, and therefore this product is odd.
Inductive Proofs

Perhaps the most useful technique, at least for this class.

An inductive proof is useful to show that a statement is true over all possible values (or all values of interest), without actually proving each value individually (which for infinite problems is not even an option).

In general, the inductive proofs we will do in this course will be over integers, and so the statements you are seeking to prove can be restated as

\[ \text{For all } N \Rightarrow 0, \text{ statement } S \text{ is true for } N \]

An inductive proof has two main components:

1. A Base Case

   The base case is a simple case, which can be proven directly. For our purposes, the base case is almost always \( N = 0 \).

1. The Inductive Case

   First, state the inductive hypothesis: the hypothesis that if the statement is true for \( N \), it is also true for \( N+1 \). Then prove that the inductive hypothesis is true.

   Once we have shown that the statement is true for \( N=0 \) and if true for \( N \), true for \( N+1 \), we’ve proven it for all \( N \Rightarrow 0 \).
An Example of Induction

Thm: if there are N people in a room, and every person shakes hands with every other person exactly once, then exactly $n(n-1)/2$ handshakes will occur.

Proof:
Other Principles

• The pigeonhole principle: if you have x pigeons and y pigeonholes, and you place each of the pigeons into the holes, then no matter how you arrange the pigeons:
  • If $x > y$, then at least one pigeonhole contains more than one pigeon.
  • If $x < y$, then at least one pigeonhole contains zero pigeons.

These principles, although seemingly simple, are actually quite useful and can be employed in a number of situations.

• Proof by contradiction: assume that the statement to be proven is FALSE, and then show how this contradicts one of the known truths-- if you can’t prove that something is true, try proving that it can’t possibly NOT be true.

• Disproof by counterexample. Finding a counterexample, or proving that a counterexample must exist (even without identifying it) immediately disproves a statement.
An Example Analysis

We know from earlier that the handshake problem requires \( \frac{N(N-1)}{2} \) handshakes, if there are \( N \) people in the room. If we allow one handshake per person per minute, and assume that people can instantly move from one part of the room to another, how many minutes will it take for everyone to shake everyone else’s hand?

Let us also assume that with each passing minute, at least one handshake takes place - our algorithm makes constant progress towards its goal.

- What is the upper bound on the number of minutes required to perform all of the handshakes?
- What is the lower bound on the number of minutes required?
- Can we find an algorithm that approaches the lower bound?
- How close can we get?