CSC236 Summer 2017
Introduction to the Theory of Computation

Alex Palioudakis
alex.palioudakis@mail.utoronto.ca
www.cdf.toronto.edu/~csc236h/summer/index.shtml

Course Notes: Introduction to the Theory of Computation
Outline

Logistics and Introduction

Chapter 1: Simple Induction
CSC236 Logistics

- Course website - check this often!
- Tutorials begin next week
- Tutorial room assignments are BA 1180, BA 1230, and BA 1240. The assignment of each student to a tutorial will be posted on the course’s website.
Lecture Format

- Reading sections will be posted in advance.

- Tutorial problems will also be posted in advance.

- A summary of lecture (and any other electronic materials) will be posted as well.
Lecture Format (Continued)

- The concepts of the course will be covered in lectures. Given the time we will see some examples in lectures as well.

- At the start of each tutorial a quiz will be given - it will be for 20 min. (For the marking scheme of the course see the course’s syllabus.

- Each quiz will examine material covered in a previous week.

- The rest of the tutorial will be examples of the day’s topic.
Why Reason about Computing?

- More than just hacking
- Testing isn’t enough
What will we learn in this course?

- Reasoning to argue a claim is right or wrong.
- A math property holds or not.
- A computer program is correct or not.
- Systematic counting.
- Practise math proofs.
You should already know

- Chapter 0 material from *Introduction to Theory of Computation*

- CSC165 material, especially mathematical notation, proof techniques, and introduction to big-Oh.

- But you can *relax* the structure of your proofs

- Recursion, efficiency material from CSC148
What is CSC236 about?

- Part 1: Induction as a proof technique.
- Part 2: Analysis of Recursive Algorithms.
- Part 3: Introduction to formal languages.
What is CSC236 about? (continued)

- One of a sequence of courses on mathematical techniques for CS

- General goal - practice reading and writing mathematically dense statements

- More practice with proofs - in CSC263, your proofs will get more creative
Part 1: Induction
Simple Induction

- Some of you will have seen a bit of this in CSC165

- We’ll start here, and then move on to other types of induction.
The Principle of Simple Induction

If the initial case works, and each case that works implies its successor works, then all cases work (we don’t have to start at 0).

\[
[ \ P(0) \land (\forall n \in \mathbb{N}, P(n) \implies P(n + 1)) \] \implies \forall n \in \mathbb{N}, P(n)\]
Simple Induction Outline

**Inductive Hypothesis:** State inductive hypothesis $H(n)$

**Verify base case(s):** Verify that the claim is true for any cases not covered in the inductive step

**Inductive Step $C(n)$:** Show that $C(n)$ follows from $H(n)$, indicating where you use $H(n)$ and why that is valid

In simple induction $H(n)$ is the claim you intend to prove about $n$, and $C(n)$ is the same claim about $n + 1$ — “simple” because the reasoning moves from $n$ to $n + 1$. 
How many subsets of a set? (count them)

- \{\}\n- \{a\}\n- \{a, b\}\n- \{a, b, c\}
How many subsets of a set? (answer)

- $\{} - 1$
- $\{a\} - 2$
- $\{a, b\} - 4$
- $\{a, b, c\} - 8$
Every set with $n$ elements has exactly $2^n$ subsets...

Scratch work: check a few more sets
Use Simple Induction

Prove by simple induction that

\[ 1 + 2 + 3 + \cdots + n = \frac{n \cdot (n + 1)}{2} \]

(Proposition 1.2 of notes)
Why Simple Induction Works?

See Theorem 1.1 (proof a) of notes.
Every set with $n$ elements has exactly $2^n$ subsets

We can prove it formally using simple induction (Proposition 1.3 of notes)
$7^n - 1$ is a multiple of 6

Scratch work: substitute a few values for $n$
The units digit of $7^n$ is either 1, 3, 7, or 9

Scratch work: substitute a few values for $n$
$4^n \geq n^4$?

Scratch work: check for a few values of $n$ ($n \geq 4$)
Represent any amount greater than 18 with any coins of value 4 or 7

Tip: Take cases for whether you use or not a coin (at least one) of value 7.