CSC236 fall 2016

Theory of computation

Danny Heap
heap@cs.toronto.edu / BA4270 (behind elevators)
http://www.cdf.toronto.edu/~csc236h/fall/
416-978-5899

use Introduction to the Theory of Computation, Section 1.2

Outline

Introduction

chaper 1, simple induction

Notes

why reason about computing?

- more than just hacking
- testing isn't enough
- careful, you might get to like it (?!*)

how to reason about computing

- it's messy...
- it's art...

how to do well

- read the course information sheet as a two-way promise
- question, answer, record, synthesize
- collaborate with respect, e.g. registered study group

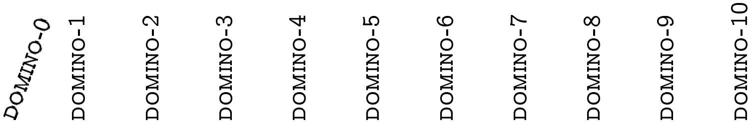
you should already know...

- Chapter 0 material from *Introduction to Theory of Computation*
- CSC165 material, especially the mathematical prerequisites (Chapter 1.5), proof techniques (Chapter 3), and the introduction to big-Oh (Chapter 4).
- But you can *relax* the structure (more on this later)
- recursion, efficiency material from CSC148

you'll know by December...

- understand, and use, several flavours of induction
- complexity and correctness of programs — both recursive and iterative
- formal languages, regular languages, regular expressions

domino fates foretold



$$[\,P(0) \;\wedge\; (\forall n \in \mathbb{N}, P(n) \Rightarrow P(n+1))\,] \Longrightarrow \forall n \in \mathbb{N}, P(n)$$

If the initial case works,
and each case that works implies its successor works,
then all cases work

simple induction outline

inductive step: state inductive hypothesis $H(n)$

derive conclusion $C(n)$: show that $C(n)$ follows from $H(n)$, indicating where you use $H(n)$ and why that is valid

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

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?

- list the subsets of $\{7\}$
- now list the subsets of $\{7, 13\}$
- is there a connection between the lists that helps you count them?

Every set with n elements has exactly 2^n subsets...

scratch work: check a few more sets...

Every set with n elements has exactly 2^n subsets...

use the simple induction outline

Every set with n elements has exactly 2^n subsets...

use the simple induction outline

$3^n \geq n^3$?

scratch work: check for a few values of n

$3^n \geq n^3$

use the simple induction outline

$3^n \geq n^3$

use the simple induction outline

For every $n \in \mathbb{N}, 12^n - 1$ is a multiple of 11

scratch work: substitute a few values for n

For every $n \in \mathbb{N}$, $12^n - 1$ is a multiple of 11

use the simple induction outline

For every $n \in \mathbb{N}$, $12^n - 1$ is a multiple of 11

use the simple induction outline

The units digit of 3^n is either 1, 3, 7, or 9

scratch work: substitute a few values for n

The units digit of 3^n is either 1, 3, 7, or 9

use the simple induction outline

The units digit of 3^n is either 1, 3, 7, or 9

use the simple induction outline

What about: the units digit of 3^n is either 1, 2, 3, 7, or 9

use the simple induction outline

is the claim still true? What happens if you add this other case to the inductive step?

Notes