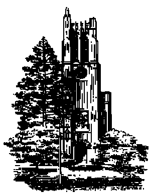
Thursday June 2, 2016
Lecture 11

Sequences and Cardinality



Notables

- Homework#7
  - Page 168, Problem 12
  - Page 169, Problems 26((a), (c), and (g), 32, and 34
  - Page 176, Problem 4
  - Due Tuesday June 7, 2016
- Tentative Schedule for the week

Week	M	T	W	R	Topic	Section
1						
		5-31			functions	2.3
			6-1		Sequences and summations	2.4
				6-2	Cardinality of sets	2.5

Summations…

1. Distributive Law:

$$\sum_{k\in A} c\cdot a_k = c\cdot \sum_{k\in A} a_k$$

2. Associative Law:

$$\sum_{k\in A}(a_k + b_k) = \left(\sum_{k\in A} a_k\right) + \left(\sum_{k\in A} b_k\right)$$

3. Commutative Law:

$$\sum_{k\in A} a_k = \sum_{\pi(k)\in A} a_{\pi(k)} \quad \text{where } \pi(k) \text{ is}$$

any permutation of the set of natural numbers. Example:

$$\sum_{0\le k\le n} (a + b\cdot k) = \sum_{0\le n-k\le n} (a + b\cdot(n-k))$$

Arithmetic Series

Prove that

$$S = \sum_{0\le k\le n} (a + b\cdot k) = \frac{(n+1)(2a + b\cdot n)}{2}$$

Proof: Recall that

$$S = \sum_{0\le k\le n} (a + b\cdot k) = \sum_{0\le n-k\le n} (a + b\cdot(n-k))$$

$$2S = \sum_{0\le k\le n} (a + b\cdot k + a + b\cdot(n-k)) = \sum_{0\le k\le n} (2a + b\cdot n)$$

$$2S = (2a + b\cdot n)\cdot \sum_{0\le k\le n} 1 = (n+1)(2a + b\cdot n)$$

$$S = \frac{(n+1)(2a + b\cdot n)}{2}$$

Geometric Series

Prove that if a and r are real numbers and $r \neq 0$, then

$$S = \sum_{j=0}^{n} a\cdot r^j = \begin{cases} \dfrac{a\cdot r^{n+1} - a}{r-1} & \text{if } r \neq 1 \\ (n+1)a & \text{if } r = 1 \end{cases}$$

Proof:

$$S = ar^0 + ar^1 + ar^2 + \cdots + ar^{n-1} + ar^n$$

$$rS = ar^1 + ar^2 + \cdots + ar^n + ar^{n+1}$$

$$rS - S = (r-1)S = ar^{n+1} - ar^0 = ar^{n+1} - a$$

$$S = \frac{ar^{n+1} - a}{r-1}$$

Geometric Series, special case

Given that $|r| < 1$, we have

$$\sum_{j=0}^{\infty} r^j = \lim_{n\to\infty} \frac{r^{n+1} - 1}{r-1} = \frac{1}{1-r}$$

Also, by differentiation, we have

$$\sum_{j=0}^{\infty} jr^{j-1} = \frac{1}{(1-r)^2}$$

Some Useful Summations

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{k=1}^{n} k^3 = \frac{n^2(n+1)^2}{4}$$

$$\sum_{k=1}^{n} ar^k = \frac{(ar^{n+1}-a)}{r-1} \qquad r \neq 0,\ r \neq 1$$

355

Double Summations

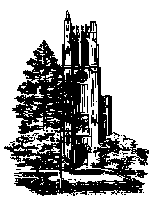
- To evaluate the double sum, first expand the inner summation and then then compute the outer summation (like double integrals).

$$\sum_{i=1}^{4}\sum_{j=1}^{3} i \cdot j = \sum_{i=1}^{4}(i+2i+3i) = \sum_{i=1}^{4} 6i = 6+12+18+24 = 60$$

356

Cantor's Legacy: Countable and Uncountable sets

Courtesy of Professor Steven Rudich
Carnegie Mellon University



∞

357

Ideal Computer

- *Ideal Computer* is defined as a computer with infinite RAM. Ideal computers differ from *real computers* in that they have no bound on the amount of memory they can use.
- If the notion of a computer with an infinite amount of memory seems unsatisfactory, think of it this way:
  - Your programs ask for as much memory as they wish. Whenever your computer runs out of memory, the computer automatically contacts the manufacturer, and a technician arrives with additional ram chips for your computer. Your computations continue where they left off, as if nothing had happened.

358

Ideal Computers…

- What can these ideal computers do that real computers can't?
- Well, real computers have a finite amount of memory. Add up all the memory they have – hard disk space, RAM, registers, etc. , and they still have only k bits of storage. This means, a real computer can be in only 2^k different states.
- Suppose you want to compute the ith digit of π, for any i. On a real computer, as your computation runs the computer changes from state to state. Eventually, it will run out of new states and must return to a state indistinguishable from a previous one. Thus, a deterministic program on a real computer can output only repeating decimals.
- So, real computers can output some digits of π, but there will always be some ith digit which cannot be computed.
- Because ideal computers have an unlimited amount of memory, they can be in an unlimited number of different states. Thus, we can write a program on an ideal computer that can print the ith digit of π, for any i. Any digit of π which interests us will eventually appear in the output.

359

An Ideal Computer Can Be Programmed To Print Out:

- π: 3.14159265358979323846264…
- 2: 2.00000000000000000000000…
- e: 2.71828182845590452353 36…
- 1/3: 0.33333333333333333333….

360

Printing Out An Infinite Sequence..

- We say *program P prints out* the infinite sequence $s(0)$, $s(1)$, $s(2)$, …; if when *P* is executed on an ideal computer then a sequence of symbols appears on the screen such that
  - The k^{th} symbol is $s(k)$
  - For every $k \in$ ***N***, *P* eventually prints the k^{th} symbol. That is, the delay between symbol *k* and symbol *k*+1 is not infinite.

361

Computable Real Numbers

- A real number *r* is *computable* if there is a program that prints out the decimal representation of *r* from left to right. Thus, each digit of *r* will eventually be printed as part of the output sequence. Example:
  - π is "computable," because we can program an ideal computer to output the *i*th digit for any *i*.
- In general, any real number *r* is computable, if we can program an ideal computer to print it digit by digit, such that any digit we are interested in will eventually be printed. For example, to find the 2^50th digit of *r*, we start the program and count the output digits. Eventually, the 2^50th digit will appear.
- The real numbers we're familiar with -- π, *e*, and the square root of two– are all computable. But in general, are all real numbers computable?

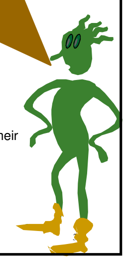
362

Describable Numbers

- A real number *r* is *describable* if it can be unambiguously denoted by a finite piece of English text. By unambiguous, we mean that the text must describe one number and only one number.
- Examples:
  - 2: "Two."
  - π: "The area of a circle of radius one."
- Our descriptions of numbers, such as π, do not necessarily tell us how to compute them. So, it is not obvious that all *describable* numbers are *computable*.

363

Is every computable real number, also a describable real number?



Computable *r* : some program outputs *r*

Describable *r* : some sentence denotes *r*

The moral here is that computer programs can be viewed as descriptions of their output.

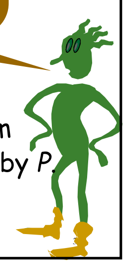
A program that computes π, for example, can be taken as an unambiguous description of the number π.

Theorem: Every computable real number is also describable

- Proof: Let *r* be a computable real that is output by a program *P*. The following is an unambiguous denotation:
  - The real number output by the following program: *P*

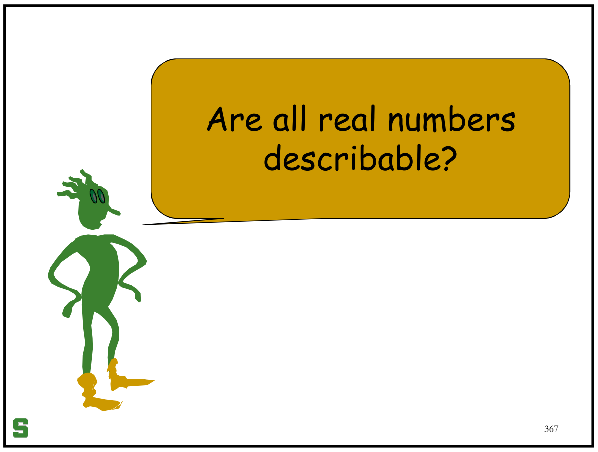
365

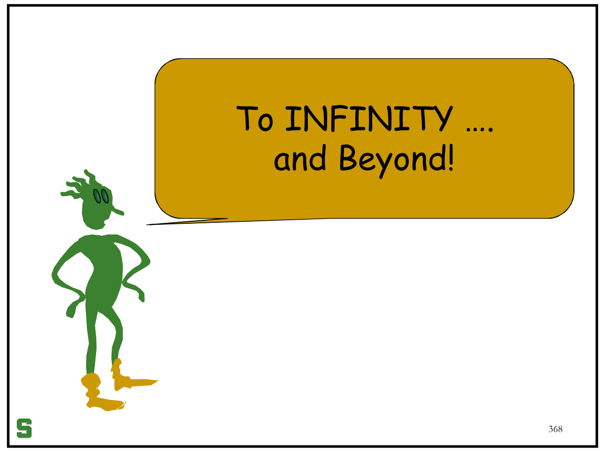
MORAL: A computer program can be viewed as a description of its output.



Syntax: The text of the program

Semantics: The real number output by P.


Are all real numbers
describable?
367


To INFINITY ….
and Beyond!
368

