Fitsum, Esther, Chong, Conner, Ngozi, Matt

Math 251, Wed 20-Oct-2021 -- Wed 20-Oct-2021
Discrete Mathematics
Fall 2021

Wednesday, October 20th 2021

Wk 8, We
Topic:: Euclidean algorithm
Read:: Rosen 5.2

Test 2: Wed. Oct. 27

Coverage: Late Ch. 2 (2.4-2.5)
Ch. 3 (focus on 3.2-3.3)
Early Ch. 5 (5.1-5.2)

We have two equivalent notions of induction. A third notion that looks different but is nonetheless equivalent is the **Well-Ordering principle**.

A is nonempty

Definition 2 (Well-Ordering Principle): Suppose $A \subseteq \mathbb{N}$. Then A has a *smallest element*. That is, $\exists a \in A$ such that $\forall b \in A$, $(a \leqslant b)$.

Note that the set {positive real numbers} does not have a smallest element, but that this does not violate the well-ordering principle.

skip 3. A simple polygon with $n \geqslant 3$ sides can be triangulated into $n-2$ triangles (use strong mathematical induction, and the fact that every simple polygon with at least four sides has an interior diagonal).

Division Algorithm

divisor quotient remainder

4. Given any integer n and any positive integer d, there exist integers q and r such that $n = dq + r$ and $0 \leqslant r < d$ (use the well-ordering principle).

$n = 311$
$d = 17$

$$\begin{array}{r} 18 \\ 17\,\overline{)\,311} \\ \underline{17} \\ 141 \\ \underline{136} \\ 5 \end{array}$$

d q r

Consider the set

$$A = \underbrace{\{n - dq : q \in \mathbb{Z}\}} \cap \mathbb{N}$$

only contains nonneg. ints.

If $n = 311$, $d = 17$ includes

$\{\ldots, 260, 277, 294, 311, 328, 345, \ldots\}$

So, the Well-Ordering Principle applies and tells us the set A has a smallest element.

$0 \leq r < d$

Claim: That smallest element serves the role of r (the remainder) and the q that produces r (via $n - qd$) serves as the quotient.

5.3 : Recursive definitions & Structural Induction

Examples of recursive definitions

basis step

1. arithmetic and geometric sequences

   $a_0 = 51$, $a_n = a_{n-1} + 23$

   $g_0 = -2$, $a_n = (a_{n-1})3$

   – recursive step

2. Fibonacci numbers

   $f_0 = 0$, $f_1 = 1$ (basis step)

   $f_n = f_{n-1} + f_{n-2}$ (recursive step)

3. $a_0 = 1$, and $a_n = n * a_{n-1}$ for $n \in \mathbb{Z}^+$

a_0	a_1	a_2	a_3	a_4	a_5	a_6	
1	1	2	6	24	120	720	...

```
int function fact(int n)
  if n==0 return 1
  else return n*fact(n-1)
```

4. Define the set $\mathcal{W}$ as follows: $\mathcal{W} = \{T, F, p, q, \ldots\}$

   basis step: T, F, and p, where p is a propositional variable, are in $\mathcal{W}$.
   recursive step: If $E, F \in \mathcal{W}$ then each of $(\neg E)$, $(E \wedge F)$, $(E \vee F)$, $(E \to F)$, and $(E \leftrightarrow F)$ are in $\mathcal{W}$.

   $p \wedge q$ $p \vee \neg q$

   Call $\mathcal{W}$ the set of **well-formed formulae** in propositional logic. Why are $\wedge pq$ and $\neg \wedge p \vee q$ not well-formed?

5. Consider set S of strings formed this way:

   basis step: the letter a is a string in S **recursive step**: for each $x \in S$, ax and xb are in S.

   basis step: $S = \{a\}$

   aa, ab admitted to S

   aaa, aab, abb

~~abba~~

~~abab~~

aaaaaaabbbb

6. Take Σ be some set of allowable characters. Define λ to be the empty string, the string containing no characters. We can think of the set Σ^* of **strings over the alphabet** Σ as defined inductively:

   basis step: $\lambda \in \Sigma^*$ $\Sigma^* = \{ \text{""} \}$

   recursive step: If $w \in \Sigma^*$ and $x \in \Sigma$, then $wx \in \Sigma^*$.

   $\Sigma^* : \{a, b, \ldots, z\}$ after one recursive step

   After many, Σ^* contains strings of any length w/ chars. from Σ.

7. Define the **length** function for inputs from Σ^* recursively:

   basis step: $\ell(\lambda) = 0$ (length of empty string)

   recursive step: For $w \in \Sigma^*$ and $x \in \Sigma$, $\ell(wx) = \ell(w) + 1$. (Here wx is the concatenation of w followed by x.)

8. Consider the Cantor middle-thirds set defined recursively as

   basis step: Start with the full interval of real numbers $C_0 = [0, 1] = \{x \mid 0 \leqslant x \leqslant 1\}$.

   recursive step: For each unbroken subinterval I still in C_n, divide I into three parts of equal length: $I = [a, b] \cup (b, c) \cup [c, d]$, and include only $[a, b] \cup [c, d]$ in C_{n+1}.

9. We define recursively various collections of **rooted trees**. Let the set $\mathcal{R}$ be defined as follows:

   basis step: A single vertex $r \in \mathcal{R}$.

   recursive step: Suppose $T_1, T_2, \ldots T_n \in \mathcal{R}$ are disjoint having roots $r_1, \ldots, r_n$, respectively. The graph formed by taking as root a vertex r not in any of $T_1, \ldots, T_n$, and adding an edge from r to each of the vertices $r_1, \ldots, r_n$ is also in $\mathcal{R}$. Call the resulting tree $T_1 \cdot T_2 \cdots T_n$.