Math 55 Lecture 11 §5.2, 5.3, start §6.1

Recall: <u>Well-ordering property (WOP)</u>:

Can use WOP as tool in proofs.

<u>Ex</u>: Division Algorithm. Prove that if $a \in \mathbb{Z}$ and $d \in \mathbb{Z}^+$

<u>Proof</u>: Let S be the set of non-neg. integers of the form

S is nonempty because

By construction,

<u>Claim</u>:

Second part: show q & r are unique (exercise)

Rk: Can use WOP to show that induction is a

Theorem: Let $P(n)$ be a proposition for every $n \in \mathbb{Z}^+$. Suppose that $P(1)$ is true, and $P(k) \Rightarrow P(k+1)$ for every $k \in \mathbb{Z}^+$. Then

Proof:

§5.3 Recursive definitions

Sometimes it's hard to define a function or object explicitly, & is easier to define it

Ex:

The Fibonacci numbers $\{f_n\}$, defined by
(Base case)
(Recursive)

Def: A recursively defined function
is a
Base case
Recursive step

Ex: Give a recursive definition of the sequence $\{a_n\}$, $n = 1, 2, 3, \ldots$ if
(a)
(b)
For (a):
For (b):

Ex: Is the following a valid recursively defined function with domain $\mathbb{N}$?

Ex: Prove that if f_n are the Fibonacci numbers,
Proof:
Base Case:
Inductive Step

$P(k)$:

$P(k+1)$:

Can define sets recursively:

Ex: Consider subset S of set of integers defined by

Base case:

Recursive step:

Can define algorithms recursively:

"Def:" An algorithm is

Def: An algorithm is called recursive if

Ex: Give a recursive algorithm for computing

Sol: Based on recursive defn. of

Algorithm: if $n=0$, then

Ex: Give a recursive alg for computing the

Sol:

Algorithm:

Ex: Let $\{f_n\}$ be the Fibonacci numbers.
Show that

Pf: Base case:

Inductive step: Let $P(n)$ be

Suppose $P(j)$ true for
Consider $P(k+1)$.
Want to show:

Start §6.1

2 basic rules for counting: product rule & sum rule.

Product Rule: Suppose that a procedure can be broken down into a sequence of two tasks. If

<u>Ex 1:</u> Bob has

<u>Ex 2:</u> How many functions are there from a set S

<u>Ex 3:</u> How many one-to-one functions are there from

<u>Note:</u>

If $m \leq n$: