Math 582
Introduction to Set Theory

Kenneth Harris
kaharri@umich.edu

Department of Mathematics
University of Michigan

March 1, 2009

Bounded subsets

☞ We saw in Lecture 15 that $(\mathbb{N}, <)$ is a well-ordered set. We will see in the next topic, the Ordinals, that there are well-ordered sets which are not isomorphic to $\mathbb{N}$.

However the following is a distinguishing feature:

Theorem

If a nonempty subset of $\mathbb{N}$ has an upper bound in the $<$ ordering, then it has a $<$-greatest element.

Proof

☞ Let $X \subseteq \mathbb{N}$ be a nonempty set with an upper bound:
that is, there is an $n \in \mathbb{N}$ with

$$n \geq x \qquad \text{for all } x \in X.$$

☞ Define B as follows

$$B = \{n \in \mathbb{N} \mid n \text{ is an upper bound of } X\}.$$

Since $B \neq \emptyset$, it has a least element, b.

☞ It remains to show that $b \in X$.

Proof – continued

There are two cases.

☞ If $b = 0$, then $X = \{b\}$ since X is nonempty and $0 < x$ for all nonzero x.

☞ If $b \neq 0$, then $b = S(k)$ for some k. Suppose $b \notin X$.
If $x < b = S(k)$, then $x \leq k$, so k is also be an upper bound of X. $\lightning$
Thus, $b \in X$.

This completes the proof.

Characterizing ordering $(\mathbb{N}, <)$

☞ We now show that the upper bound property of the previous Theorem, together with well-ordering, characterizes $(\mathbb{N}, <)$.

Theorem

Let $(W, \prec)$ be a nonempty well-ordered set with the additional properties:

(a) *There is no largest element:*
for every $w \in W$, there is a $z \in W$ with $w \prec z$.

(b) *Every nonempty subset of W that has an upper bound has a $\prec$-greatest element.*

Then $(W, \prec)$ is isomorphic to $(\mathbb{N}, <)$.

Proof

☞ Let w_0 be least in W, and let $h : W \to W$ be given by

$$h(w) = \text{ least } z \succ w,$$

which is well defined by (a). Define by recursion $f : \mathbb{N} \to \mathbb{N}$ satisfying

$$\begin{aligned} f(0) &= w_0 \\ f(S(n)) &= h(f(n)). \end{aligned}$$

☞ We will prove two facts about f:

(i) $n < m$ implies $f(n) \prec f(m)$, and

(ii) $\text{ran}(f) = W$.

From (i) it follows that f is injective, and so together with (ii), that f is an order isomorphism. (See Slide 14 from Lecture 12).

Proof – continued

(i). The proof is by on m:

$$\forall n\,\big(n < m \;\rightarrow\; f(n) \prec f(m)\big).$$

Basis. Trivial.

Inductive. Suppose true for m. Then $f(S(m))$ is $\prec$-least in W greater than $f(m)$.

If $n < S(m)$ then either $n < m$ or $n = m$. In the first case, $f(n) \prec f(m) \prec f(S(m))$; in the second case, $f(n) = f(m) \prec f(S(m))$.

This completes the inductive step of (i).

☞ It follows that f is injective:

Suppose $f(m) = f(n)$, but $m \neq n$. Either $m < n$ or $n < m$. If $n < m$, then $f(n) \prec f(m)$ by (i), contradicting $f(m) = f(n)$. Similarly for $m < n$. Thus, $m = n$.

Proof – continued

(ii). We need to show that $\operatorname{ran}(f) = W$. Suppose not. Let $w \in W - \operatorname{ran}(f)$ be $\prec$-least. Consider the set

$$X = \{z \in W \,\big|\, z \prec w\}.$$

Since $w_0 \in \operatorname{ran}(f)$ and $w_0 \prec w$, it follows that X is nonempty and bounded. By (b), X has a $\prec$-greatest element z.

☞ We have $X \subseteq \operatorname{ran}(f)$ as w is least in $W - \operatorname{ran}(f)$. Let $f(n) = z$. Then

$$f(S(n)) = \text{ least in } W \text{ greater than } z\,,$$

so $f(S(n)) \notin X$, and thus $w \not\prec f(S(n))$.

Thus, $f(n) \prec w \prec f(S(n))$ contradicting definition of f.

Therefore, $W = \operatorname{ran}(f)$.

Primitive Recursion on the natural numbers

✎ Primitive Recursion on the natural numbers is the priniciple for defining a function f such that evaluation of $f(x)$ may require the evaluation of one or more values $y < x$.

✎ All instances of primitive recursion we have required have defined $f(S(x))$ using only $f(x)$; but, there is little additional complication in allowing access to all smaller values.

✎ Consider the following definition of the Fibonacci numbers

$$f(0) = f(1) = 1 \qquad f(x) = f(x-1) + f(x-2) \text{ for } x > 1.$$

It is easy to compute f by filling-out a table of values working left-to-right

x	0	1	2	3	4	5	...
$f(x)$	1	1	2	3	5	8	...

Finite sequences

☞ In Lecture 3 we defined the n-fold Cartesian product A^n to represent sequences of length n from a given set A. This is not convenient when we want to study finite sequences on A.

Definition

A finite sequence from A is a function whose domain is $\{0, \ldots, n-1\}$ for some natural number n. We write $f \in \textbf{Seq}(A)$ to mean

- $f \subseteq \mathbb{N} \times A$ is a function, and
- $\mathrm{dom}(f) = \{i \mid i < n\}$ for some $n \in \mathbb{N}$.

If f is a finite sequence, we define its length as the largest n such that for all $i \in \mathrm{dom}(f)$ for all $i < n$.

If f is a finite sequence of length n we will write f as

$$\langle a_i \mid i < n \rangle \quad \text{or} \quad \langle a_0, \ldots, a_{n-1} \rangle.$$

Finite sequences

☞ There is a unique empty sequence of length 0,
which we write as $\langle\rangle = \emptyset$.

☞ If $f : \mathbb{N} \to A$, we write $f \restriction n = f \restriction \{i \mid i < n\}$.

☞ If f is a finite sequence on A, then $f \subseteq \mathbb{N} \times A$, so $f \in \mathcal{P}(\mathbb{N} \times A)$.
We have not yet defined the "powerset" yet, so we cannot be sure $\mathcal{P}(\mathbb{N} \times A)$ exists.

Once we do have the Powerset Axiom, we can define the set of finite sequences on A:

$$Seq(A) = \{f \in \mathcal{P}(\mathbb{N} \times A) \mid f \in \mathbf{Seq}(A)\}$$

$Seq(A)$ is a set, which we cannot yet prove exists, and $\mathbf{Seq}(A)$ is a predicate that we have defined in the language of set theory.

Defining Fibonacci on natural numbers

The idea for defining the Fibonacci numbers is to define

$$f(x) = G(f \restriction x)$$

where $G : Seq(\mathbb{N}) \to \mathbb{N}$ is defined by the two conditions:
for any $s \in Seq(\mathbb{N})$

(i) $G(s) = s(x-1) + s(x-2)$: where x is the length of s and $x \geq 2$, and

(ii) 1 : otherwise.

✉ We will have to work around the fact that we may not be able to prove such a set function G exists.

✉ G is defined on alot of garbage values that will never arise (when s has the wrong domain); this is no problem, since the proof that this definition works shows that the garbage values never arise in constructing f, so never cause a problem.

A word on functions

There are two notions of function at play in Complete Recursion:

① Set functions: these are functions as sets-of-ordered-pairs and are objects of set theory. We write $f : X \to Y$ to mean f is a set-of-ordered-pairs satisfying $(x, y), (x, z) \in f \;\to y = z$ with set domain X and set range Y.

② Class functions: these are functions as rules-associating-argument-to-value, and are **not actual sets**, but are statements in the language of set theory.
We write $\mathbf{G} : \mathbf{V} \to \mathbf{V}$ to mean that there is a formula $\varphi(x, y)$ satisfying the following two conditions:

$$\forall x \exists! y\, \varphi(x, y)$$

$$\mathbf{G}(x) = y \quad \leftrightarrow \quad \varphi(x, y)$$

Complete Recursion

Theorem (Complete Recursion)

If $\mathbf{G} : \boldsymbol{V} \to \boldsymbol{V}$ then there is a unique function f whose domain is $\mathbb{N}$ such that

$$f(n) = \mathbf{G}(f \restriction n) \qquad \text{for all } n \in \mathbb{N}.$$

Note. $\mathbf{G}(x) = y$ is really an abbreviation of some formula $\varphi(x, y)$; and the formula $\varphi(x, y)$ may have other free variables, parameters, which play no role in the proof. These parameters are useful for defining functions, but are otherwise inert.