Defining % (remainder)?

- '%': **N** × **N** → **N**
- '%' = {((0, 1), 0), ((1, 1), 0), …, ((0, 2), 0), ((1, 2), 1), …} infinitely many members!
- '%' = {((*x*, *y*), *z*) | *z* = *x* % *y*} circular!
- '%' = {((*x*, *y*), *z*) | $x = y \times n + z$ for some *n*, where $x, y, n, z \in$ **N** and $z < y$}

Then, how to define '+'?

How to define these?

- Addition
- Natural numbers
- Strings
- Cities which can be traveled from Trenton by driving
- All of your ancestors

Are they all human?

Review: Set Definition

- Finite sets
  - Can be defined using the list notation
- Infinite sets
  - Cannot be defined using the list notation
  - Could be defined using the predicate notation
  - Must avoid circular definitions
  - Could be defined in terms of more primitive sets

Does infinity exist? Useful?

Most primitive infinite set?

Hypothesis

- If we can define the set of natural numbers, we can define all sorts of infinite sets.
- Consequence: You can define even **PingPong** (including '%') completely formally.

Module D

- Foundation
- Most abstract
- Intellectual challenge ~ problem solving skills ~ mathematical reasoning skills
- Tools to convince others (who understand logic and mathematics, e.g., people in science and engineering)

Moved toward the end

Unit D1: Sets

Today

- Defining natural numbers
- Understand and use inductive definition of a set
- Take-home exercises
  - Web pages, Foreign policy

Section 1

Defining Natural Numbers

- Approach: Specification using logical statements
- To specify a structure that would contain **N** (the set of natural numbers) as the primary set (called *carrier*)

Will make several attempts

Group Exercise 1.1

- Consider a collection of structures $\mathbf{Nat_1} = (A, \leq, 0)$ specified by the following:
  1. $\exists x\ (x = 0)$
  2. $\forall x \exists y\ (x < y)$

  $x < y$ is abbreviation for $x \leq y \wedge x \neq y$
- Identify multiple instances of $\mathbf{Nat_1}$ that would satisfy the conditions
  - Number of elements?
  - Represent schematically

Group Exercise 1.2

- Consider a collection of structures $\mathbf{Obj_1} = (O, \mathit{noHigherThan}, \mathit{Bottom})$ specified by the following:
  1. $\exists x\ (x = \mathit{Bottom})$
  2. $\forall x \exists y\ (x\ \mathit{below}\ y) \Leftrightarrow \forall x \exists y\ (y\ \mathit{above}\ x)$

  $x\ \mathit{below}\ y$ is abbreviation for $x\ \mathit{noHigherThan}\ y \wedge x \neq y$
- Identify multiple instances of $\mathbf{Obj_1}$ that would satisfy the conditions
  - Number of elements?
  - Represent schematically

Group Exercise 1.3

- Consider a collection of structures $\mathbf{Plot_1}$ = (*O*, ✂, 💣) specified by the following:
  1. ∃*x* (*x* = 💣)
  2. ∀*x*∃*y* (*x* ✁ *y*)

  x ✁ *y* is abbreviation for *x* ✂ *y* ∧ *x* ≠ *y*
- Identify multiple instances of $\mathbf{Plot_1}$ that would satisfy the conditions
  - Number of elements?
  - Represent schematically

More Things to Check

- Linear?
- Least element?
- Hasse diagram? I.e., poset?
- Smallest structure?

Group Exercise 2

- Consider a collection of structures $\mathbf{Nat_2} = (A, \mathit{succ}, 0)$ specified by the following:
  1. $\exists x\ (x = 0)$
  2. $\forall x \exists y\ (y = \mathit{succ}(x))$

  succ for successor, but do not be trapped by assumptions
- Identify multiple instances of $\mathbf{Nat_2}$ that would satisfy the conditions
  - Number of elements?
  - Represent schematically

Think flexibly

Conditions for Natural Numbers

- Linearity ~ special case of poset
- Least element
- Intended meaning for '≤' and *succ*

Specifying Natural Numbers

- $\mathbf{Nat_3} = (A, \leq, succ, 0)$
- Conditions
  - Poset
  - Linearity
  - Least element
  - Definition of *succ*

$\mathbf{Nat_3} = (A, \leq, succ, 0)$

succ for successor
$x < y$ is abbreviation for $x \leq y \wedge x \neq y$

- Poset
  - $\forall x \, (x \leq x)$
  - $\forall x \forall y \, ((x \leq y \wedge y \leq x) \rightarrow x = y)$
  - $\forall x \forall y \forall z \, ((x < y \wedge y < z) \rightarrow x < z)$
- Linearity: $\forall x \forall y \, (x \leq y \vee y \leq x)$
- Least element: $\forall x \, (0 \leq x)$
- Definition of *succ*
  - $\forall x \exists y \, (x \neq 0 \rightarrow x = succ(y))$
  - $\forall x \forall y \, (x < succ(y) \leftrightarrow x \leq y)$

Number Theory (Math)

Digression: Hidden Problem

- The conditions for $\mathbf{Nat_3}$ can be satisfied by many unintended structures.
- In fact, there are infinite many of them.

Model Theory (Mathematical Logic)

Section Summary

- It is possible to define the set of natural numbers using the logic-structure connection.
- Disadvantages
  - Complicated
  - Still unintended structures
- Need for a convenient pattern to define infinite sets

Strings

$\mathbf{String} = (S, +, \varepsilon)$

- S: set of all strings
- How to define A formally?

Section 2

Inductive Definition of **N**

- Base case: $0 \in$ **N**
- Induction step: If $x \in$ **N**, $succ(x) \in$ **N**
- Exclusion clause: Nothing else is in **N**.

Schematic

- Abbreviation/alternatives
  - Inductive definition: recursive definition
  - Base case: Base, basis
  - Exclusion clause: Exclusion

In FOL?

Essence of Inductive Definition

- Base case
  - Specification of starting members (finite number)
- Induction step
  - Specification of how to grow the set (potentially infinitely)
- Exclusion clause
  - Specification of how to stop the growth

Integers

- Define the set of integers: I
  - Assume: $prev = succ^{-1}$ (i.e., the inverse)
- Base:
- Induction step:
- Exclusion:

Strings (Group Exercise 3)

- Define the set of strings: S
  - $\Sigma = \{a, b, c, ..., z\}$
  - $attachChar: S \times \Sigma \rightarrow S$
- Base:
- Induction step:
- Exclusion:

Cf. definition of strings using a regular expression

Ancestors (1)

- Define the set of your ancestors: A
  - $isParentOf: A \times A$
- Base: $you \in A$
- Induction step:
- Exclusion: Nothing else is in A.

Are we all brothers/sisters?

Ancestors (2)

- Define the set of your ancestors: A
  - $parents: A \rightarrow$ ____ (returns all the parents)
- Base: $you \in A$
- Induction step:
- Exclusion: Nothing else is in A.

Cities

- Define the set of cities drivable from Trenton: C
  - $drivableFromTo$: $C \times C$
- Base: $Trenton \in C$
- Induction step:
- Exclusion: Nothing else is in C.

Infinite?
Undirected graph?

Section Summary

- Inductive definition is a convenient way to precisely and concisely define potentially infinite sets.
  - Cf. the definition of $\mathbf{Nat_3}$ in Section 1

Summary Exercise

- Can you define all the possible chess positions inductively? [no need to actually do this; just explain whether you can do it]
- Questions/Comments/Suggestions