CAS 701 Fall 2008

03 Numbers, Sets, Functions, and Relations

William M. Farmer

Department of Computing and Software
McMaster University

23 September 2008

Number Systems

- The main number systems:
  - $\mathbf{N}$, the natural numbers.
  - $\mathbf{Z}$, the integers.
  - $\mathbf{Q}$, the rational numbers.
  - $\mathbf{R}$, the real numbers.
  - $\mathbf{C}$, the complex numbers.
- Other important number systems:
  - $\mathbf{Z}_n$, the integers modulo n.
  - $\mathbf{O}$, the ordinal numbers.
  - $\mathbf{H}$, the hyperreal numbers.
  - $\mathbf{S}$, the surreal numbers.

Foundational Mathematical Objects

- The three most common kinds of foundational objects:
  1. Sets.
  2. Functions.
  3. Relations.
- Each kind of object can be used to represent the other two kinds of objects.

Sets

- A set is a collection of objects.
- Some very large collections of objects cannot be sets.
  - For example, consider the the Russell set, the set of all sets that do not contain themselves.
  - A collection that is too large to be a set is called a proper class.
- Styles of set theories:
  - Naive set theory.
  - Having a universal set.
  - Having no universal set (e.g., ZF set theory).
  - Having a universal class (e.g., NBG set theory).

Set Concepts

- Basic properties: membership, subset, cardinality.
- Basic operations:
  - Union, intersection, complement, difference, symmetric-difference.
  - Cartesian product (product), disjoint union (sum).
  - Sum set, power set.
- Special sets: the emptyset, universal sets, functions, relations, ordinals, cardinals.
- Functions and relations can be represented as special kinds of sets (e.g., as sets of tuples).

(Unary) Functions

- Definition 1: A function is a rule $f : I \to O$ that associates members of I (inputs) with members of O (outputs).
  - Every input is associated with at most one output.
  - Some inputs may not be associated with an output. Example: $f : \mathbf{Z} \to \mathbf{Q}$ where $x \mapsto 1/x$.
- Definition 2: A function is a set $f \subseteq I \times O$ such that if $(x, y), (x, y') \in f$, then $y = y'$.
- Each function f has a domain $D \subseteq I$ and a range $R \subseteq O$.
  - f is total if $D = I$ and partial if $D \subset I$.
- A set or relation can be represented as a special kind of function (e.g., as a predicate, a characteristic function, or an indicator).

Lambda Notation

- Lambda notation is a precise, convenient way to specify functions.
- If B is an expression of type β,

  $$\lambda x : \alpha \,.\, B$$

  denotes a function $f : \alpha \rightarrow \beta$ such that $f(a) = B[x \mapsto a]$.
- Example: Let $f = \lambda x : \mathbf{R} \,.\, x * x$.
  - $f(2) = (\lambda x : \mathbf{R} \,.\, x * x)(2) = 2 * 2$.
  - f denotes the squaring function.
- Lambda notation is used in many languages to express ideas about functions.
- Examples:
  - Lambda Calculus (a model of computability).
  - Simple Type Theory (a higher-order predicate logic).
  - Lisp (a functional programming language).

n-Ary Functions

- Definition 1: For $n \geq 0$, an n-ary function is a rule $f : I_1, \ldots, I_n \rightarrow O$ that associates members of $I_1, \ldots, I_n$ (inputs) with members of O (outputs).
  - Every list of inputs is associated with at most one output.
  - Some lists of inputs may not be associated with an output.
- Definition 2: For $n \geq 0$, an n-ary function is a set $f \subseteq I_1 \times \cdots \times I_n \times O$ such that if $(x_1, \ldots, x_n, y), (x_1, \ldots, x_n, y') \in f$, then $y = y'$.
- Each function f has a domain $D \subseteq I_1 \times \cdots \times I_n$ and a range $R \subseteq O$.

Representing n-Ary Functions as Unary Functions

There are two ways of representing a n-ary function as a unary function:

1. As a function of tuples: $f : I_1, \ldots, I_n \longrightarrow O$ is represented as

$$f' : I_1 \times \cdots \times I_n \longrightarrow O$$

   where

$$f(x_1, \ldots, x_n) = f'((x_1, \ldots, x_n)).$$

2. As a curryed function: $f : I_1, \ldots, I_n \longrightarrow O$ is represented as

$$f'' : I_1 \longrightarrow (I_2 \longrightarrow (\cdots (I_n \longrightarrow O) \cdots))$$

   where

$$f(x_1, \ldots, x_n) = f''(x_1) \cdots (x_n).$$

Example

- Let $f = \lambda\, x, y : \mathbf{R} \,.\, x^2 + y^2$.
- $f' = \lambda\, p : \mathbf{R} \times \mathbf{R} \,.\, [\mathrm{fst}(p)]^2 + [\mathrm{snd}(p)]^2$.

  $$\begin{aligned} f'((a,b)) &= (\lambda\, p : \mathbf{R} \times \mathbf{R} \,.\, [\mathrm{fst}(p)]^2 + [\mathrm{snd}(p)]^2)((a,b)) \\ &= [\mathrm{fst}((a,b))]^2 + [\mathrm{snd}((a,b))]^2 \\ &= a^2 + b^2. \end{aligned}$$

- $f'' = \lambda\, x : \mathbf{R} \,.\, \lambda\, y : \mathbf{R} \,.\, x^2 + y^2$.

  $$\begin{aligned} f''(a)(b) &= (\lambda\, x : \mathbf{R} \,.\, \lambda\, y : \mathbf{R} \,.\, x^2 + y^2)(a)(b) \\ &= (\lambda\, y : \mathbf{R} \,.\, a^2 + y^2)(b) \\ &= a^2 + b^2. \end{aligned}$$

Function Concepts

- Basic properties:
  - Arity (0-ary, unary, n-ary with $n \geq 2$, multiary).
  - Total, injective, surjective, bijective.
  - Image, inverse image.
- Basic operations: composition, restriction, inverse.
- Special functions: the empty function, identity functions, choice functions.

Cardinality

- Two sets A and B are equipollent, written $A \approx B$, if there is a bijection $f : A \to B$ between them.
- $A \preceq B$ means $A \approx B'$ for some $B' \subseteq B$.
- A set is infinite if it is equipollent with a proper subset of itself.
- The cardinality of a set A is the cardinal number c such that A and c are equipollent.
- Theorem.
  1. $\mathbf{N} \approx \mathbf{Q}$.
  2. (Cantor) $\mathbf{N} \not\approx \mathbf{R}$.
- Theorem (Schröder-Bernstein). If $A \preceq B$ and $B \preceq A$, then $A \approx B$.

Relations

- For $n \geq 1$, an n-ary relation is a set $R \subseteq A_1 \times \cdots \times A_n$ $(n \geq 1)$.
  - Any set can be considered as a unary relation.
  - Any nonunary relation can be considered as a binary relation.
- Functions are considered as special relations.
  - An n-ary function $f : A_1, \ldots, A_n \rightarrow B$ is identified with the corresponding $(n+1)$-ary relation $R_f \subseteq A_1 \times \cdots \times A_n \times B$ called the graph of the function.
- An n-ary relation can be represented by an n-ary predicate.

Relation Concepts

- Basic binary relation properties:
  - Reflexive, symmetric, transitive.
- Basic binary relation operations:
  - Domain, range.
  - Composition, inverse.
- Special relations: the empty relation, universal relations, equivalence relations.
- Ways of representing relations:
  - Using zero-one matrices.
  - Using directed graphs.

Closures of Relations

- Reflexive closure.
- Symmetric closure.
- Transitive closure.
  - Equals the connectivity relation.

Equivalence Relations

- A binary relation on a set is an equivalence relation if it is reflexive, symmetric, and transitive.
- Given an equivalence relation R on a set S, the equivalence class of $a \in S$ is the set

$$\{b \in S \mid a \; R \; b\}.$$

- Theorem.

  1. The equivalence classes of an equivalence relation on a set S form a partition of S.
  2. Given a partition of a set S, there is an equivalence relation on S whose equivalence classes are the members of the partition.

Algebras

- A mathematical structure is a structured collection of sets, functions, and relations.
- An algebra is a mathematical structure consisting of:
  1. A set of elements called the domain.
  2. A set of distinguished elements, functions, and relations called the signature that impose a structure on the domain of elements.
- The signature usually determines a language for describing and making assertions about the elements of the domain.
- Algebras are often described by tuples of the form

$$(D, e_1, \ldots, e_k, f_1, \ldots, f_m, r_1, \ldots .r_n).$$

Examples of Algebras

- Number systems (e.g, natural, integer, rational, real, complex, ordinal, hyperreal, surreal).
- Algebraic structures (e.g, monoids, groups, rings, fields).
- Orders (e.g., pre, partial, total, well).
- Lattices and boolean algebras.
- Graphs and trees.
- Abstract data types (ADTs) used in Computer Science (e.g, ADTs for strings, lists, streams, arrays, records, stacks, queues).