

David Keil
Framingham State University

1. Diagonal proofs and uncountable sets

1. Countable sets
2. Uncountable sets
3. Incompleteness
4. Streams and coinduction

David Keil Theory of Computing 1/121

Inquiry

How many of the following exist?

- natural numbers
- rational numbers
- real numbers
- functions
- programs
- sets

David Keil Theory of Computing 1/122

Objectives

- 1a. To write a proof showing countability of a set
- 1b. To write a diagonal proof showing uncountability of a set
- 1c. To explain or write a coinductive definition

How Topic 1 relates to computing

- Our most important result in this course is that some problems are algorithmically unsolvable
- This sets a boundary on the power of computation
- Turing's uncomputability proof is *diagonal*
- Cantor's proof that $|\mathbb{N}| < |\mathbb{R}|$ and Gödel's proof of limits of logic are also diagonal
- Computation deals with finite, infinite, and uncountable objects or sets
- Computations are operations on sets of strings (languages)

1. Countable sets

- The natural numbers are *countable*, i.e., they can be arranged in a particular order
- A *bijection* is a one-to-one correspondence, i.e., a relation between two sets that is both surjective and injective
- A set is *countable* iff there exists a bijection between it and the natural numbers (\mathbb{N})
- Countable sets tend to be sets of finite objects, defined inductively

Peano's axioms: definition by induction

1. 0 is a natural number ($0 \in \mathbf{N}$)
2. Every natural number n has a unique successor, n' , also a natural number
($\forall n \in \mathbf{N}$) $n' \in \mathbf{N}$
3. All natural numbers follow (1) or (2)
($\forall n \in \mathbf{N}$) $n = 0 \vee (\exists m \in \mathbf{N}) n = m'$

- *Significance:* These axioms, or assumptions, provide a formal logical basis to work with counting numbers. *Note:* (2) is recursive
- Computation is a formal way to manipulate numbers and objects representable by them.

Theorem: \mathbb{N} is infinite

Proof by contradiction:

1. Suppose \mathbb{N} is finite
2. Then let n be the largest element of \mathbb{N}
3. Consider the number $(n + 1)$. It cannot be in \mathbb{N} , because it is larger than \mathbb{N} 's largest element
4. But by definition of \mathbb{N} , any successor of an element of \mathbb{N} is in \mathbb{N} .
5. Hence by the contradiction of (3) and (4), we must reject (1).
6. Hence \mathbb{N} is infinite.

Countability of strings

- To help prove equivalence of number-based and string-based models of computation, we show mappings of any string to a natural number and any natural number to a string
- Enumeration:

\mathbb{N}	$\{0,1\}^*$
0	λ
1	0
2	1
3	00
4	01

A bijection $\mathbb{N} \leftrightarrow \{0,1\}^*$

$$\text{Str}(n) = \begin{cases} \lambda & \text{if } n = 0 \\ \text{Str}(\lfloor n/2 \rfloor) + '0' & \text{if } \text{Odd}(n) \wedge n > 0 \\ \text{Str}(\lfloor n/2 \rfloor) + '1' & \text{otherwise} \end{cases}$$

$$\text{Num}(s) = \begin{cases} 0 & \text{if } s = \lambda \\ 2 \times \text{Num}(s[1 .. |s| - 1]) + s[|s|] + 1 & \text{otherwise} \end{cases}$$

Strings are defined inductively

- A *string* is a sequence of symbols
- *Notation:*
 - x^R is the string x reversed
 - $n_0(x)$ is the number of zeroes in string x
- *Induction* defines countable sets of finite objects
- *Example:* Strings over alphabet Σ
 $\Sigma^* = \{\lambda\} \cup \{ax \mid a \in \Sigma, x \in \Sigma^*\}$
- Sets of strings are *countable*, e.g., by numbering them according to bit representation: $\lambda, 0, 1, 00, 01, 10, 11, \dots$

Formal languages

- *Alphabet*: a finite set of symbols
Examples: $\{0,1\}$, $\{0, 1, \dots,9\}$, $\{a, b, c, \dots, z\}$
- *String*: a finite sequence of symbols; $\lambda = \text{null}$
- *Language*: a set of strings over an alphabet
- Let alphabet Σ be $\{0,1\}$, let λ be the null string
 Then $\Sigma^0 = \{\lambda\}$, $\Sigma^1 = \{(0), (1)\}$
 $\Sigma^2 = \{00, 01, 10, 11\}$
 $\Sigma^k = \text{the set of strings of length } k$
 $\Sigma^* = \bigcup_{k \in \mathbb{N}} \Sigma^k$
- $0^n 1^n$ is the language of strings with n zeroes followed by n ones

Languages and countability

- A language may be represented as an infinite bit sequence, with a bit in a certain bit location denoting membership in the language: $\{0, 10, 11, 100\} = 10101100\dots$
- Computation operates on strings; instances of models of computation accept sets of strings

Countable set: the rational numbers

- The rationals may be enumerated as at right, where the numerator is the row number and the denominator is the column number

	1	2	3	4	5	6	7	8	...
1	$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{8}$...
2	$\frac{2}{1}$	$\frac{2}{2}$	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{2}{5}$	$\frac{2}{6}$	$\frac{2}{7}$	$\frac{2}{8}$...
3	$\frac{3}{1}$	$\frac{3}{2}$	$\frac{3}{3}$	$\frac{3}{4}$	$\frac{3}{5}$	$\frac{3}{6}$	$\frac{3}{7}$	$\frac{3}{8}$...
4	$\frac{4}{1}$	$\frac{4}{2}$	$\frac{4}{3}$	$\frac{4}{4}$	$\frac{4}{5}$	$\frac{4}{6}$	$\frac{4}{7}$	$\frac{4}{8}$...
5	$\frac{5}{1}$	$\frac{5}{2}$	$\frac{5}{3}$	$\frac{5}{4}$	$\frac{5}{5}$	$\frac{5}{6}$	$\frac{5}{7}$	$\frac{5}{8}$...
6	$\frac{6}{1}$	$\frac{6}{2}$	$\frac{6}{3}$	$\frac{6}{4}$	$\frac{6}{5}$	$\frac{6}{6}$	$\frac{6}{7}$	$\frac{6}{8}$...
7	$\frac{7}{1}$	$\frac{7}{2}$	$\frac{7}{3}$	$\frac{7}{4}$	$\frac{7}{5}$	$\frac{7}{6}$	$\frac{7}{7}$	$\frac{7}{8}$...
8	$\frac{8}{1}$	$\frac{8}{2}$	$\frac{8}{3}$	$\frac{8}{4}$	$\frac{8}{5}$	$\frac{8}{6}$	$\frac{8}{7}$	$\frac{8}{8}$...
...

- Reference: “The set of rational numbers is countable,” (<http://www.homeschoolmath.net/teaching/rational-numbers-countable.php>)

2. Uncountable sets

- Cardinalities of sets: informally, their sizes
- For infinite sets, we say that two sets have the same cardinality iff there is a *bijection* (1-to-1 correspondence) between their elements
- Example:* $\text{Card}(\text{EVEN}) = \text{Card}(\mathbb{N})$
Proof: $\text{EVEN}_0 = 0, \text{EVEN}_1 = 2, \text{etc.}$
- Example:* $\text{Card}(\mathbb{Q}) = \text{Card}(\mathbb{N})$
Proof: Zig-zag through a table of rationals

The set of real numbers

- Intuitively, the real numbers express analog or continuous quantities; they give all the possible weights or distances.
- We may consider the size of the set \mathbb{R} , of reals, as corresponding to all the points on a line segment, line, plane, space, or space/time
- *Another definition:* The reals are all the numbers that can be expressed using an infinite sequence of digits after the decimal or binary point

Theorem: $\text{Card}(\mathbb{R}) > \text{Card}(\mathbb{N})$ (Cantor)

Proof:

1. Suppose \mathbb{R} were countable
2. Then the interval $(0, 1]$ could be enumerated as:

$$r_1 = 0 . b_{1,1} b_{1,2} b_{1,3} \dots$$

$$r_2 = 0 . b_{2,1} b_{2,2} b_{2,3} \dots$$

$$\dots$$
3. Now, consider the real number $s =$

$$0 . \neg b_{1,1} \neg b_{2,2} \neg b_{3,3} \dots$$
 That is, for all n , bit n is $\neg b_{n,n}$ (bitwise negation of the diagonal of r)

Cantor's thm. (cont'd)

4. Now, since for every i , the i^{th} bit of s is different from the i^{th} bit of r_i , so s differs from every element of r , so there is no element of sequence r that matches s
5. Hence r does not contain s , which is clearly a real number
6. Hence we have a contradiction and must reject the supposition
7. Hence \mathbb{R} is not countable

The proof, illustrated

Supposed enumeration of \mathbb{R} :

$r_1 = (\underline{0}, 0, 0, 0, 0, 0, 0, \dots)$

$r_2 = (1, \underline{1}, 1, 1, 1, 1, 1, \dots)$

$r_3 = (0, 1, \underline{0}, 1, 0, 1, 0, \dots)$

$r_4 = (1, 0, 1, \underline{0}, 1, 0, 1, \dots)$

$r_5 = (1, 1, 0, 1, \underline{0}, 1, 1, \dots)$

$r_6 = (0, 0, 1, 1, 0, \underline{1}, 1, \dots)$

$r_7 = (1, 0, 0, 0, 1, 0, \underline{0}, \dots)$

...

Real number that differs with every element in enumeration in at least one bit:

$s = (\underline{1}, \underline{0}, \underline{1}, \underline{1}, \underline{1}, \underline{0}, \underline{1}, \dots)$

Summary of Cantor's proof

- “The proof shows that no matter how a list of real numbers were arranged, a real number could still be defined that would not be in the list”

T. DiRienzo, J. Bartlett, 2/09

- A diagonal proof is both *by construction* and *by contradiction*

A simple diagonal proof

- Let Ω be “the set of all sets”
- Hence $\Omega \in \Omega$
- *Theorem:* “Set of all sets” is a self-contradictory notion
- *Proof:* Consider $\Omega' = \{ A : A \notin A \}$
(sets not members of themselves)
- $(\Omega' \in \Omega') \Rightarrow (\Omega' \notin \Omega')$,
 $(\Omega \notin \Omega) \Rightarrow (\Omega \in \Omega)$, contradictions
- Note role of “Spoiler” instance Ω'
- We say that Ω, Ω' are *not well founded* (NWF)

How many functions exist?

- Start with the set of functions
 $\{f: \{0\} \rightarrow \{0,1\}\}$
- This is a set of two functions:
 $\{(0,1), (0,0)\}$
- The set $\{f: \{0, 1\} \rightarrow \{0,1\}\}$ has four elements,
 $\{f: \{0, 1, 2\} \rightarrow \{0,1\}\}$ has eight, etc.
- There are $2^{\mathbb{N}}$ predicates $f: \mathbb{N} \rightarrow \{0,1\}$
- *Question:* Is this the same as, or more than, the cardinality of the natural numbers?
- Compare with the cardinality of reals

Can all predicates on \mathbb{N} be computed in Java?

1. Enumerate all Java methods that take an integer parameter and return 0 or 1, as J_1, J_2, J_3, \dots in a bitwise ordering
2. Consider the predicate $f: \mathbb{N} \rightarrow \{0,1\}$ s.t. $f(n) = 1$ if $J_n(n) = 0$ or $J_n(n)$ hangs, $f(n) = 0$ if $J_n(n) = 1$
3. The predicate f differs in its behavior from each element of our enumeration of Java methods
4. Hence f is a predicate not computed by any Java method

Results

- \mathbb{N} is countable (cardinality \aleph_0 , aleph-null)
- \mathbb{R} is uncountable (cardinality \aleph_1)
- The set of Java functions is countable using their bit representations
- The set of predicates $f: \mathbb{N} \rightarrow \{0,1\}$ is uncountable
- There are $2^{\aleph_0} = \aleph_1$ predicates on \mathbb{N}
- The *sets* of natural numbers are uncountable (see handout)

3. Incompleteness

Gödel's Theorem: Limits of logic and computation

David Keil

*Based on a talk at the Framingham State College
Math/Computer Science Faculty Seminar, March 27, 2003*

Overview

- Kurt Gödel, 1931, at age 25, in Vienna, shook up the foundations of mathematical thought as developed in Russell and Whitehead's *Principia Mathematica*
- He showed that no formal axiomatic system of logic could be both *consistent* and *complete*
- Since predicate calculus was proven consistent, Gödel's theorem showed that some truths are unprovable in the system

General-purpose definitions

- **Consistency:** Attribute of a system in which no false assertion can be proven
 $(\forall p) ((\vdash p) \Rightarrow p)$
- **Completeness:** Attribute of a system in which every true assertion can be proven
 $(\forall p) (p \Rightarrow \vdash p)$
- **Gödel number:** A value that uniquely encodes a logical assertion or sequence of assertions; e.g., a proof

Intuition of Gödel's proof

- Gödel needed a counter-example to the claim that a system can be both complete and consistent
- He showed that *something like* the following is true but unprovable:
“*This assertion has no proof*”
- But the assertion actually used has a precise meaning

Definitions related to theorem

- $Proves(x, y, z)$: The assertion that the sequence of symbols with Gödel number x is a valid proof that $\phi_y(z)$ is true, where ϕ_y is the formula with Gödel number y , and z is a natural number.
- Let $G(y)$ be the assertion that $\neg(\exists x)Proves(x, y, y)$ i.e., $G(y)$ means that what the formula with Gödel number y asserts about y is unprovable
- *Example*: Where y encodes the predicate *even*, $G(y)$ is the assertion that one cannot prove that the encoding of the predicate *even* is an even number

Gödel's theorem

“ $\neg(\exists x) \textit{Proves}(x, \#(G), \#(G))$ ” is true in predicate calculus, but cannot be proven in the predicate calculus

- *Or*: What G asserts about predicate G is true but unprovable
- *Or*: There is no proof that $G(\#(G))$ is true, but $G(\#(G))$ is nevertheless true

Proof of Gödel's theorem

1. Suppose the theorem's claim is false and there exists a proof in predicate calculus that $\neg(\exists x) \textit{Proves}(x, \#(G), \#(G))$.
 2. Then the claim is true, by consistency of predicate calculus.
 3. Therefore by its own assertion it is unprovable in predicate calculus.
 4. This contradicts step 1, hence claim holds.
- Note*: The above proof goes outside the formal system of predicate calculus.

Another way to see the proof

- Think of an infinite truth table, with all formulas on one variable enumerated down the left side and arguments to the formulas as column headers
- There is a *true* in column c of row r iff $f_r(c)$ is true
- Let q be the Gödel number of the formula that asserts the unprovability of its argument
- Then there must be a *true* in column q of row q , because $f_q(q) = \text{false}$ would say that formula q is not unprovable, hence provable, hence true, a contradiction
- But if $f_q(q) = \text{true}$, then $f_q(q)$ is both true and (as it claims) unprovable

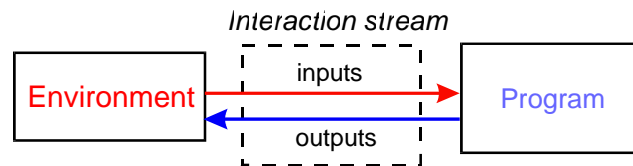
Incompleteness and computer science

- Using the same *diagonal proof* method, Turing proved a special case of Gödel's theorem
- Given a *computable* function f , executing an algorithm to compute $y = f(x)$ proves that $f(x) = y$
- But for some functions f , for some x , $y = f(x)$, but there is no proof that $f(x) = y$
- This is the same as saying that no algorithm computes f -- example: the Halting Problem
- Gödel's work in *recursive function theory* matches Turing's results with automata

Larger significance

- Gödel's theorem was part of a series of results establishing *limits of knowledge*, that also included
 - Einstein's relativity results
 - Heisenberg's Uncertainty Principle
 - Turing's theorem that some functions are uncomputable
- It marked the failure of Russell's, Whitehead's and Hilbert's effort to reduce mathematics to first-order logic and set theory
- A category of *valid assertions* is established that we cannot *know* are valid

4. Streams and coinduction



- An infinite stream of inputs to a reactive system may be imagined, together with the corresponding infinite stream of outputs
- *Sets of streams* express interactive behavior
- *Coinduction* defines sets of infinite objects as induction defines sets of finite ones

Inductively defined sets

- (i) 0 is a natural number;
- (ii) Every $n \in \mathbb{N}$ has a unique successor, n' ;
- (iii) i and ii are the only ways to obtain a natural number (Peano)
- Set of expressions using numerals, +, and ():
 - $expression \rightarrow$
 - $numeral \mid$
 - $numeral + expression \mid$
 - $(expression)$
- The definition of such a set may *use* itself
- But such a set does not *contain* itself

Inductive and coinductive definitions of languages

- $\Sigma^* = \{\lambda\} \cup \{ax \mid a \in \Sigma, x \in \Sigma^*\}$
(note base case λ)
- “ L is a language over alphabet Σ^* ”
means that $L \subseteq \Sigma^*$
- A *stream* is an infinite string
- A *stream language* is a set of streams
- Example, defined coinductively:
 - $\Sigma^\infty = \{ax \mid a \in \Sigma, x \in \Sigma^\infty\}$
 - (note lack of base case)

Well-founded sets

- (B. Russell) Let the village hair stylist be the person who cuts the hair of everyone who doesn't cut own hair
- Question: Who cuts the hair-stylist's hair?
- Similar Questions: Is there a set of all sets? If so, does it contain itself? Is there a set of all sets that don't contain themselves?
- In classical (well founded) set theory, it is meaningless to say a set belongs or doesn't belong to itself (Foundation Axiom)

Streams and non-well-founded sets

- A non-well-founded set may contain itself, directly or indirectly
- *Example:* $A = \{ B, C \}; B = \{ A, D \}$
- Every stream of characters is a character, followed by a stream of characters
- Streams contain streams, which contain streams, ...
- *Ex.:* The set of all bit streams $\{0,1\}^\infty$ consists of any infinite sequence that consists of 0 or 1 followed by a stream

Streams and coinduction

- *Induction* defines countable sets of finite objects: $\Sigma^* = \{\lambda\} \cup \{ax \mid a \in \Sigma, x \in \Sigma^*\}$
- *Coinduction*
 - is a dual of induction (recursion), lacking a base case
 - is used to define sets of infinite objects
- The set of streams over an alphabet, expressing ongoing processes such as interaction:

$$\Sigma^\infty = \{ax \mid a \in \Sigma, x \in \Sigma^\infty\}$$

Least and greatest fixed points

- A *fixed point* x of function f is the value x (which may be a set), for which $f(x) = x$
- Whereas induction is characterized by *minimality* conditions (least fixed points), coinduction has maximality condition (greatest fixed point)
- *Minimality example*: Σ^+ is the *smallest* set that fits the spec $\{ax \mid a \in \Sigma, x \in \Sigma^*\}$
- *Maximality example*: Σ^∞ is the *largest* set that fits the spec $\{ax \mid a \in \Sigma, x \in \Sigma^\infty\}$

Questions

- What most stayed in your mind in discussing this topic?
- For you, what was the *least* clear concept that you encountered in this topic?

References (Gödel)

John W. Dawson. *Logical Dilemmas: The Life and Work of Kurt Gödel*. A. K. Peters, 1997.

Gary Flake. *The Computational Beauty of Nature*. MIT Press, 2001, Ch. ____

Kurt Gödel. *On Formally Undecidable Propositions of the Principia Mathematica and Related Systems*. 1931.

Ernest Nagel and James R. Newman. *Gödel's Proof*. New York University Press, 1958.

References (Coinduction)

Barwise-Moss. *Vicious Circles*, 1996

Finsler, 1920s (NWF sets)

Goldin-Wegner (at www.engr.uconn.edu/~dgg)

Peano, 1889 (induction)