Applying Computability to Study Mathematical Theorems

Carl Mummert

November 1, 2016

Marshall University Combinatorics Seminar

Note

This is the second of two 50-minute talks on computability theory for the Marshall University Combinatorics Seminar in Fall 2016.

These talks are meant to be a general introduction to the area for colleagues who work in other areas of combinatorics.

The material presented is well known in the field. Although I have not included detailed references, no material here is original or due to me.

A selection of general references is provided at the end of the talk.

Introduction

In general, **mathematical logic** has two senses:

1. Using mathematical methods to clarify “logical” concepts such as provability, truth, computability, set theory, etc.

2. Using these “logical concepts” to study mathematics.

The first talk in this series was about sense 1: using mathematical methods to clarify the nature of computability.

This talk is about sense 2: using that refined understanding of computability to study mathematical theorems.

Introduction

Last time, we talked about the basics of classical computability theory.

Recall that φ_e is the computable function with index (program) e.

By using oracle computations, we can use these to compute functions from $\mathbb{N}$ to $\mathbb{N}$, from $P(\mathbb{N})$ to $P(\mathbb{N})$, or from $\mathbb{N}^{\mathbb{N}}$ to $\mathbb{N}^{\mathbb{N}}$, or mixtures of the three.

The Church–Turing thesis states that these functions encompass all the functions on these sets that we would normally view as algorithmically computable by a human.

Introduction

We want to use computability theory to study mathematical theorems:

- When a theorem states that an object exists, can we compute that object?
- If not, can we say anything about how noncomputable the object is?

In practice, we usually ask these questions about ***specific*** theorems, rather than about theorems in general.

Many theorems are of the form “for all X there is a Y such that ...”. We can ask whether the function implicit in the theorem is computable in some sense.

Other mathematical objects

Most mathematical objects are not natural numbers, sets of natural numbers, or infinite sequences of natural numbers.

To handle these, we need to **represent** or **code** them into objects which can be handled by computable functions.

In general, a **representation** is a relation $R(x, c)$ between a mathematical object x and a code c for that object.

Requirement: each possible code can represent at most one object, and each object has at least one code.

Examples

A common representation of an integer z as a natural number:

$$R_{\mathbb{Z}}(z, 2^n 3^m) \longleftrightarrow z = (n - m)$$

A representation of $\mathbb{Q}$, using that representation of $\mathbb{Z}$:

$$R_{\mathbb{Q}}(q, 2^z 3^w) \longleftrightarrow z = 2^a 3^b \text{ and } w = 2^c 3^d \text{ and } q = \frac{a-b}{c-d}$$

As usual, not every natural number is actually a code, and each integer or rational has infinitely many codes.

Representing countable discrete structures

We can represent countable discrete structures such as graphs, groups, etc. by first representing their objects as natural numbers.

Then we provide an oracle to compute the relations and functions in the structure.

This is usually relatively routine.

Representing the real line

It is more challenging to represent real numbers.

There are several options, including:

- $R_{\text{Decimal}}(x, f) \subseteq \mathbb{R} \times \mathbb{N}^{\mathbb{N}}$ holds if
  - $f(0)$ codes an integer n,
  - $f(i) \in \{0, \ldots, 9\}$ for $i > 0$,
  - $\sum_{i=1}^{\infty} f(i) \cdot 10^{-i}$ converges to a real d,
  - and $x = n + d$.

- $R_{\text{Cauchy}}(x, f) \subseteq \mathbb{R} \times \mathbb{N}^{\mathbb{N}}$ holds if
  - $f(k)$ codes a rational q_k for each $k \in \mathbb{N}$,
  - $x = \lim_{k \to \infty} q_k$, and
  - (q_k) has a modulus of convergence given by $N(1/m) = m$:

$$\text{Whenever } k > m, \ |q_k - x| < 1/m.$$

Computing more general functions

Suppose A and B are sets of mathematical objects and we have a function $F\colon A \to B$.

We want to give a notion of what it could mean for F to be "computable".

First we need to represent A and B in terms of more basic computable objects like $\mathbb{N}$ or $\mathbb{N}^{\mathbb{N}}$.

After that, there are several possible definitions.
We will see two in this talk.

Weihrauch style definition

Suppose that R_A is a representation of A and R_B is a representation of B.

We say that a function $F\colon A \to B$ is **W computable** (relative to R_A and R_B) if there is a computable function Φ such that:

Whenever Φ is given an R_A code for some $x \in A$, it produces an R_B code for $F(x) \in B$.

$$\begin{tikzcd} A \arrow[r, "F"] & B \\ \mathbb{N}^{\mathbb{N}} \arrow[u, leftrightsquigarrow, "R_A"] \arrow[r, "\Phi"'] & \mathbb{N}^{\mathbb{N}} \arrow[u, leftrightsquigarrow, "R_B"'] \end{tikzcd}$$

A more Russian definition

An alternative definition is related to
“Russian-style computable mathematics.”

We say that $F\colon A \to B$ is **R computable** (relative to some representations) if, whenever c is a computable code for an object x, there is at least one computable code for $F(x)$.

W computability differs by asking about all codes,
not just computable ones.

W computability also requires a single computing procedure that works for all inputs. R computability allows the procedure to change from one code to another.

The representation can matter

The representation we use can affect computability of a function.

Theorem

Let $P(x, y)$ be the addition function on $\mathbb{R}$.

- *With the representation via Cauchy sequences, P is both W computable and R computable.*
- *With the representation via decimal expansions, P is R computable but not W computable.*

Mathematical theorems

Consider a mathematical theorem such as
"Every countable commutative ring has a maximal ideal".

We can consider this as a kind of computational problem: given a code for a countable commutative ring, produce a code for a maximal ideal.

So we are trying to compute a ***section*** of the relation between countable rings and maximal ideals.

In this case, it turns out that there is no computable section for the maximal ideal relation (this is not obvious).

Mathematical theorems as problems

Many “for all ... there exists ...” theorems produce problems analogous to the maximal ideal theorem.

Some of these problems have a computable solution (relative to particular representations) while others do not.

The next step in the spirit of computability theory is to ask: how can we compare the ***amount*** of uncomputability in the problems that don’t have a computable solution?

There is a joke that computability theory should be called “uncomputability theory”, because most effort is spent on uncomputable problems.

Weihrauch reducibility

We will view a “for all ... there exists ...” theorem as a **Weihrauch problem** P, as follows:

- There is a set $I(P)$ of “instances”,
  e.g. countable commutative rings
- There is a set $S(P)$ of “solutions”,
  e.g. maximal ideals
- There is a relation $P(x, y)$ that tells when y is a solution to instance x.

The “problem” is: produce a function F so that $P(x, F(x))$ holds for all $x \in I(P)$.

We can call this function F a “solution method for P”.

Weihrauch reducibility

We can compare two Weihrauch problems P and Q:

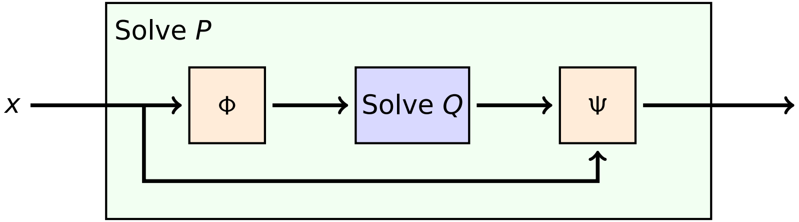
> *P is **Weihrauch reducible** to Q if there are computable functions $\Phi(x)$ and $\Psi(y, x)$ so that, for every solution method G of Q, $F(x) = \Psi(G(\Phi(x)), x)$ is a solution method for P.*

$$\begin{tikzcd}
I(P) \arrow[r, "F"] \arrow[d, "\Phi"'] \arrow[dr, dashed] & S(P) \\
I(Q) \arrow[r, "G"'] & S(Q) \arrow[u, "\Psi"']
\end{tikzcd}$$

Digging deeper, there will also be representations for $I(P)$, $I(Q)$, $S(P)$, and $S(Q)$. These are usually clear from context.

The box picture

We can also view Weihrauch reducibility by thinking of a possible procedure to solve *P* if we have a black box to solve *Q*:



In Weihrauch reducibility, we want to have specific computable functions Φ and Ψ which work no matter what solution method for *Q* is used.

Formalized comparisons

Weihrauch reducibility gives us a way to compare theorems by comparing the (non)computability of functions that solve them.

We can also compare theorems by asking which theorems imply which other theorems.

Of course, every true statement implies every other true statement.

So we restrict the proof methods that can be used, to break theorems into smaller equivalence classes based on mutual provability.

Reverse Mathematics

In **Reverse Mathematics**, we represent theorems as statements in a formal system called *second order arithmetic*. The only objects are natural numbers and sets of natural numbers.

We use a particular base system, RCA_0, which says:

- The natural numbers are a discrete ordered semiring
- The natural numbers satisfy a weak form of the axiom of mathematical induction
- If we have a set A, and a computable function φ_e^A is total, then the set $B = \{n : \varphi_e^A(n) = 1\}$ exists.

Reverse Mathematics

We compare theorems P and Q in Reverse Mathematics by asking whether P is provable from Q and RCA_0, and whether Q is provable from P and RCA_0.

Many theorems of undergraduate mathematics have been classified in both the Weihrauch reducibility framework and the Reverse Mathematics framework.

Comparison

Reverse Mathematics	**Weihrauch Reducibility**
Strengths: ▶ Most theorems fall into a small number of equivalence classes ▶ Restatements are usually equivalent to each other ▶ Also measures amount of induction needed	Strengths: ▶ Finer equivalence relation ▶ Focus on uniformity of computation ▶ Count “uses” of a theorem
Weaknesses: ▶ Unable to distinguish some natural principles ▶ Unable to “count uses” of theorems	Weaknesses: ▶ Insensitive to uses of induction ▶ Sensitive to precise statement of theorems

Motivations

Why do we want to know how uncomputable particular theorems are?

- Provides more information about the internal combinatorics of mathematical theorems
- Helps to determine whether we have “optimal” proofs
- Provides methodological information about (non)constructivity in mathematical practice

Surprising results

There are some results that are very surprising.

Ramsey's theorem for exponent n says that for every finite coloring of $[\mathbb{N}]^n$ there is an infinite set H so that $[H]^n$ is monochromatic.

Ramsey's theorem for exponent 1 is just the pigeonhole principle. Ramsey's theorem for exponents $3, 4, 5, \ldots$ is very well behaved.

But Ramsey's theorem for exponent 2 has much more unexpected and complex behavior.

This has led to a huge amount of research into combinatorial theorems related to Ramsey's theorem.

References (1 of 2)

Reverse Mathematics

- Stephen G. Simpson, *Subsystems of Second-Order Arithmetic*, 2nd ed., Cambridge Univ. Press, 2009
- Denis R. Hirschfeldt, *Slicing the Truth: On the Computable and Reverse Mathematics of Combinatorial Principles*, NUS IMS Lecture Notes Series 28, 2014.

References (2 of 2)

Weihrauch Computability

- Klaus Weihrauch, *Computable Analysis: An Introduction*, Springer, 2000.

Weihrauch Computability and Weihrauch Degrees

- Vasco Brattka and Guido Gherardi, “Weihrauch Degrees, Omniscience Principles and Weak Computability”, Proceedings of the Sixth International Conference on Computability and Complexity in Analysis, 2009, 83–94.
- François G. Dorais, Damir D. Dzhafarov, Jeffry L. Hirst, Joseph R. Mileti and Paul Shafer, “On uniform relationships between combinatorial problems”, *Trans. AMS* 368:2, 2016, 1321–1359.

See *http://cca-net.de/publications/weibib.php* for many more.