CS:4330 Theory of Computation

Spring 2018

Computability Theory

Undecidability

Haniel Barbosa

The University of Iowa

Readings for this lecture

Chapter 4 of [Sipser 1996], 3rd edition. Section 4.2.

Computing as we know it is limited in a fundamental way

- ▷ There are problems which are algorithmically unsolvable.
- ▷ We will cover several computationally unsolvable problems and how to prove unsolvability.
- ▷ We start with the halting problem.

Halting problem of TMs

- Whether an arbitrary TM will halt on an arbitrary input

- It is also the membership problem of TMs:

$$A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$$

- While A_{DFA} and A_{CFG} are decidable, A_{TM} **is not**.

Undecidability of Halting Problem

Theorem

A_{TM} is undecidable.

- ▷ Before we proceed to the proof, we first establish that A_{TM} is Turing-recognizable
- ▷ This proves that recognizers are more powerful than deciders
- ▷ Requiring that a TM halts on all inputs restricts its expressive power.

A recognizer for A_{TM}

The following TM U recognizes A_{TM}

$U =$"On input string $\langle M, w \rangle$, where M is a TM and w is a string:

1. Simulate M on input w
2. If M ever enters its accept state, *accept*; if M ever enters its reject state, *reject*."

U is an *universal Turing machine*, which can simulate any other Turing machine from its description.

A recognizer for A_{TM}

The following TM U recognizes A_{TM}

$U =$"On input string $\langle M, w \rangle$, where M is a TM and w is a string:

1. Simulate M on input w
2. If M ever enters its accept state, *accept*; if M ever enters its reject state, *reject*."

U is an *universal Turing machine*, which can simulate any other Turing machine from its description.

Halting

- ▷ Note that U is not a decider, since it possibly loops indefinitely.
- ▷ The name "halting problem" comes from the impossibility of U determining whether M ever halts.

How to prove undecidability

- ▷ The proof of undecidability of the TM membership problem uses Georg Cantor (1873) technique called *diagonalization*
- ▷ Cantor's problem was to measure the size of infinite sets
- ▷ The size of finite sets is measured by counting the number of their elements
- ▷ The size of infinite sets cannot be measured by counting their elements since this procedure does not halt

Examples of infinite sets

- ▷ The set of strings over $\{0, 1\}$ is infinite
- ▷ So is the set $\mathbb{N}$ of natural numbers[1]
- ▷ The set $\mathbb{E}$ of all even natural numbers is also infinite
- ▷ How can we compare these sets?

[1] Here we believe that 0 is a natural number.

Examples of infinite sets

- ▷ The set of strings over $\{0, 1\}$ is infinite
- ▷ So is the set $\mathbb{N}$ of natural numbers[1]
- ▷ The set $\mathbb{E}$ of all even natural numbers is also infinite
- ▷ How can we compare these sets?

Cantor's solution

- ▷ Two sets have the same size if their elements can be paired (i.e. you can establish a bijection, a one-to-one correspondence)
- ▷ Since this method does not rely on counting it serves both finite and infinite sets

[1] Here we believe that 0 is a natural number.

Correspondence

For two sets A and B and a function $f : A \to B$

- f is *one-to-one* if it never maps two different elements of A into the same element of B, i.e. f is injective, i.e. $f(a) \neq f(b)$ whenever $a \neq b$
- f is *onto* if it hits every element of B, i.e. f is surjective, i.e. $\forall y \in B. \exists x \in A. f(x) = y$
- f is called a *correspondence* if it is both *one-to-one* and *onto*

Two sets A and B have the same size if there is a correspondence $f : A \to B$

Example Correspondences

- ▷ Let $\mathbb{N}$ be the set of natural numbers and $\mathbb{E}$ the set of even natural numbers

- ▷ Intuitively one may believe that $\#(\mathbb{N}) > \#(\mathbb{E})$ since $\mathbb{E} \subset \mathbb{N}$. However, using Cantor's method we can show that $\mathbb{N}$ and $\mathbb{E}$ have the same size by constructing the correspondence $f : \mathbb{N} \to \mathbb{E}$ defined by $f(n) = 2n$

Example Correspondences

- ▷ Let $\mathbb{N}$ be the set of natural numbers and $\mathbb{E}$ the set of even natural numbers
- ▷ Intuitively one may believe that $\#(\mathbb{N}) > \#(\mathbb{E})$ since $\mathbb{E} \subset \mathbb{N}$. However, using Cantor's method we can show that $\mathbb{N}$ and $\mathbb{E}$ have the same size by constructing the correspondence $f : \mathbb{N} \to \mathbb{E}$ defined by $f(n) = 2n$

Definition

A set is *countable* if it is finite or it has the same size as $\mathbb{N}$.

A complex correspondence

Let $\mathbb{Q}$ be the set of positive rational numbers, i.e. $\mathbb{Q} = \{\frac{m}{n} \mid m \in \mathbb{N},\ n \in \mathbb{N}^+\}$

- ▷ Intuitively, $\mathbb{Q}$ seems to be much larger than $\mathbb{N}$
- ▷ Yet we can show that these two sets have the same size by constructing a correspondence

Correspondence $\mathbb{Q} \to \mathbb{N}$

1. Put $\mathbb{N}$ on two axes
2. Line i contains all rationals that have numerator i, i.e. $\left\{\frac{i}{j} \mid i \in \mathbb{N} \text{ fixed}, j \in \mathbb{N}^+\right\}$
3. Column j contains all rationals that have denominator j, i.e. $\left\{\frac{i}{j} \mid i \in \mathbb{N}, j \in \mathbb{N}^+ \text{ fixed}\right\}$
4. Number $\frac{i}{j}$ occurs in i-th row and j-th column

Listing rational numbers

- *Bad idea*: list first elements of a line or a column.
  Lines and columns are labeled by natural numbers, therefore this would never end.
- *Good idea* (by Cantor): use the diagonals.
  1. First diagonal contains $\frac{0}{1}$
  2. Continue the list with the elements of the next diagonal skipping repetitions: $\frac{1}{1}, \frac{2}{1}, \frac{1}{2}, \ldots$
  3. Elements that may generate repetitions, such as $\frac{i}{i}$, which would generate a copy of $\frac{1}{1}$, or $\frac{0}{i}$, which would be a copy of $\frac{0}{1}$

Listing rational numbers

- ▷ *Bad idea*: list first elements of a line or a column.
  Lines and columns are labeled by natural numbers, therefore this would never end.
- ▷ *Good idea* (by Cantor): use the diagonals.
  1. First diagonal contains $\frac{0}{1}$
  2. Continue the list with the elements of the next diagonal skipping repetitions: $\frac{1}{1}, \frac{2}{1}, \frac{1}{2}, \ldots$
  3. Elements that may generate repetitions, such as $\frac{i}{i}$, which would generate a copy of $\frac{1}{1}$, or $\frac{0}{i}$, which would be a copy of $\frac{0}{1}$

More infinite countable sets

- ▷ $\mathbb{N} \times \mathbb{N}$
- ▷ $\mathbb{N}^k$, for any $k \in \mathbb{N}$
- ▷ Σ^*
- ▷ Any subset of a countable set is also countable

Uncountable sets

An infinite set for which no correspondence with $\mathbb{N}$ can be established is denoted *uncountable*.

Theorem

The set of real numbers is uncountable.

Proof idea

We can show this using Cantor's diagonalization method.

No correspondence exists between $\mathbb{N}$ and $\mathbb{R}$

- ▷ Suppose that such a correspondence $f : \mathbb{N} \to \mathbb{R}$ exists and deduce a contradiction showing that f fails to work properly.
- ▷ We construct an element $x \in \mathbb{R}$ that cannot be the image of an $n \in \mathbb{N}$
- ▷ We must show that $x \neq f(n)$ for every $n \in \mathbb{N}$

No correspondence exists between $\mathbb{N}$ and $\mathbb{R}$

- ▷ Suppose that such a correspondence $f : \mathbb{N} \to \mathbb{R}$ exists and deduce a contradiction showing that f fails to work properly.
- ▷ We construct an element $x \in \mathbb{R}$ that cannot be the image of an $n \in \mathbb{N}$
- ▷ We must show that $x \neq f(n)$ for every $n \in \mathbb{N}$

Building x

Construct $x \in (0, 1)$ by the following procedure:

$$x = 0.d_0 d_1 d_2 d_3 d_4 \ldots$$

such that $\forall i \in \mathbb{N}$, d_i is a digit different from the i-th digit of $f(i)$.

No correspondence exists between $\mathbb{N}$ and $\mathbb{R}$

- ▷ Suppose that such a correspondence $f : \mathbb{N} \to \mathbb{R}$ exists and deduce a contradiction showing that f fails to work properly.
- ▷ We construct an element $x \in \mathbb{R}$ that cannot be the image of an $n \in \mathbb{N}$
- ▷ We must show that $x \neq f(n)$ for every $n \in \mathbb{N}$

Building x

Construct $x \in (0,1)$ by the following procedure:

$$x = 0.d_0 d_1 d_2 d_3 d_4 \ldots$$

such that $\forall i \in \mathbb{N}$, d_i is a digit different from the i-th digit of $f(i)$.

- ▷ x is different from *all* real numbers in the image of f by at least one digit
- ▷ Therefore, since $x \in \mathbb{R}$ and $\forall n \in \mathbb{N}. x \neq f(n)$, the function f is not surjective, so it cannot be a correspondence
- ▷ The "diagonalization" comes from using the diagonal of the table with entries $(n, f(n)), n \in \mathbb{N}$ to build x.

Some languages are not Turing-recognizable

- ▷ There are uncountably many languages yet only countably many Turing machines

Some languages are not Turing-recognizable

- ▷ There are uncountably many languages yet only countably many Turing machines

Set of all languages is uncountable

1. The set $\mathscr{B}$ of all infinite binary strings is uncountable
2. There is a correspondence between the set of all languages $\mathscr{L}$ and $\mathscr{B}$

Set of all TMs is countable

1. Σ^* is countable
2. Each TM M has an encoding $\langle M \rangle$ into a string

- ▷ Since each Turing machine recognizes a single language and there are more languages than TMs, some languages are not recognized by any TM
- ▷ Such languages are not Turing recognizable

Proving the Halting problem is undecidable

We assume that $A_{\text{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$ is decidable:

- ▷ Let H be a decider for A_{TM}

$$H(\langle M, w\rangle) = \begin{cases} \textit{accept} & \text{if } M \text{ accepts } w \\ \textit{reject} & \text{if } M \text{ does not accept } w \end{cases}$$

- ▷ Let D be a TM that uses H as a subroutine: it calls H to determine how M behaves on the input $\langle M\rangle$ and outputs the opposite.
- ▷ $D =$"On input string $\langle M\rangle$, where M is a TM:
  1. Run H on input $\langle M, \langle M\rangle\rangle$
  2. Output the opposite of what H outputs, i.e. if H accepts, *reject*; if H rejects, *accept*."
- ▷ What about running D on $\langle D\rangle$?

Proving the Halting problem is undecidable

We assume that $A_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$ is decidable:

- ▷ Let H be a decider for A_{TM}

$$H(\langle M, w\rangle) = \begin{cases} \textit{accept} & \text{if } M \text{ accepts } w \\ \textit{reject} & \text{if } M \text{ does not accept } w \end{cases}$$

- ▷ Let D be a TM that uses H as a subroutine: it calls H to determine how M behaves on the input $\langle M\rangle$ and outputs the opposite.
- ▷ $D =$"On input string $\langle M\rangle$, where M is a TM:
  1. Run H on input $\langle M, \langle M\rangle\rangle$
  2. Output the opposite of what H outputs, i.e. if H accepts, *reject*; if H rejects, *accept*."
- ▷ What about running D on $\langle D\rangle$?

$$D(\langle D\rangle) = \begin{cases} \textit{accept} & \text{if } D \text{ does not accept } \langle D\rangle \\ \textit{reject} & \text{if } D \text{ accepts } \langle D\rangle \end{cases}$$

D cannot exist, so neither can H!

Where is the diagonalization?

- ▷ The use of diagonalization can be seen if we construct a table of all Turing Machines $M_0, \ldots, M_n$ (rows) running on encoded Turing Machines $\langle M_0 \rangle, \ldots, \langle M_n \rangle$ (columns) as inputs:

  Entries (i,j) are *accept* if M_i *accepts* $\langle M_j \rangle$, *reject* otherwise: $H(\langle M_i, \langle M_j \rangle \rangle)$

Where is the diagonalization?

▷ The use of diagonalization can be seen if we construct a table of all Turing Machines $M_0, \ldots, M_n$ (rows) running on encoded Turing Machines $\langle M_0 \rangle, \ldots, \langle M_n \rangle$ (columns) as inputs:

Entries (i,j) are *accept* if M_i *accepts* $\langle M_j \rangle$, *reject* otherwise: $H(\langle M_i, \langle M_j \rangle \rangle)$

▷ When we add D to the table, a contradiction occurs at $\langle D, \langle D \rangle \rangle$.

Where is the diagonalization?

- ▷ The use of diagonalization can be seen if we construct a table of all Turing Machines $M_0, \ldots, M_n$ (rows) running on encoded Turing Machines $\langle M_0 \rangle, \ldots, \langle M_n \rangle$ (columns) as inputs:

  Entries (i,j) are *accept* if M_i *accepts* $\langle M_j \rangle$, *reject* otherwise: $H(\langle M_i, \langle M_j \rangle \rangle)$

- ▷ When we add D to the table, a contradiction occurs at $\langle D, \langle D \rangle \rangle$.

- ▷ D computes the opposite of the diagonal entries, but on $\langle D, \langle D \rangle \rangle$ it must be the opposite of itself.

A Turing-unrecognizable language

Definition

A language is co-Turing-recognizable if it is the complement of a Turing-recognizable language.

Theorem

A language is decidable iff it is Turing-recognizable and co-Turing-recognizable.

Corollary

For any undecidable language, either the language or its complement is not Turing-recognizable.

Corollary

$\overline{A_{\mathsf{TM}}}$ *is not Turing-recognizable.*