CS:4330 Theory of Computation

Spring 2018

Computability Theory

Algorithm as Turing Machine

Haniel Barbosa

The University of Iowa

Readings for this lecture

Chapter 3 of [Sipser 1996], 3rd edition. Section 3.3.

Why TMs?

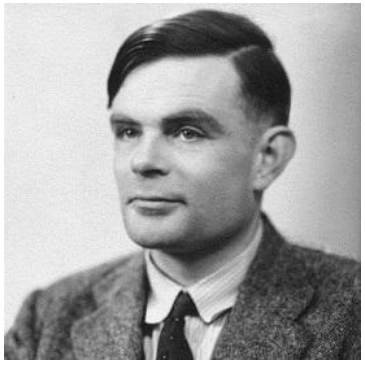
- In 1900: Hilbert posed 23 “challenge problems” in Mathematics

The 10^{th} problem:

> *Devise a process according to which it can be decided by a finite number of operations if a given polynomial has an integral root.*

It became necessary to have a formal definition of “algorithms” to define their expressivity.

Church-Turing Thesis



- ▷ In 1936 Church and Turing independently defined “algorithm”:
  - ▶ λ-calculus
  - ▶ Turing machines
- ▷ Intuitive notion of algorithms = Turing machine algorithms
- ▷ “Any process which could be naturally called *an effective procedure* can be realized by a Turing machine”
- ▷ We now know: Hilbert’s 10^{th} problem is **undecidable**!

Algorithm as Turing Machine

Definition (Algorithm)

An *algorithm* is a decider TM in the standard representation.

- ▷ The input to a TM is always a string.
- ▷ If we want an object other than a string as input, we must first represent that object as a string.
- ▷ Strings can easily represent polynomials, graphs, grammars, automata, and any combination of these objects.

Encoding and Decoding objects

- ▷ Our notation for encoding an object O into its string representation is $\langle O \rangle$
- ▷ If we have several objects $O_1, \ldots, O_n$ we denote their encoding into a string by $\langle O_1, \ldots, O_n \rangle$
- ▷ Encoding itself can be done in many ways. It doesn't matter which encoding we pick because a TM can always translate one encoding into another.
- ▷ A (part of) a TM may be programmed to decode the input representation so that it can be interpreted in the intended way.

Example TM

Let A represent the language consisting of all strings representing connected undirected graphs, i.e. $A = \{\langle G \rangle \mid G \text{ is a connected undirected graph}\}$

- A graph is connected if every node can be reached from every other node.

A TM M deciding A is such that $M =$"On input $\langle G \rangle$, the encoding of G,

1. Select the first node of G and mark it
2. Repeat until no new nodes are marked
    3. For each node in G, mark it if it is attached by an edge to a node that is already marked
4. Scan all the nodes of G and determine whether they are all marked. If they are, *accept*, otherwise *reject*."

Graph encoding

- The encoding $\langle G \rangle$ of a graph as a string is a list of nodes followed by a list of edges.

- Each node is a decimal number, and each edge is a pair of decimal numbers that represent the nodes that edge connects

- Example: what is the graphical representation of the graph encoded as

$$\langle G \rangle = (1,2,3,4)((1,2),(2,3),(3,1),(1,4))$$

Checking the encoding

When M receives the input $\langle G \rangle$ it first checks to determine that the input is a proper encoding of some graph:

1. Scan the tape to be sure that there are two lists and that they are in proper form
2. The first list should be a list of distinct decimal numbers; the second list should be a list of pairs of decimal numbers
3. The list of decimal numbers should contain no repetitions
4. Every node on the second list should appear in the first list too.

Note

Element distinctness problem can be used to format the lists and to implement the checks above.