EECS 2001N: Introduction to the Theory of Computation

Suprakash Datta
Office: LAS 3043

Course page: `http://www.eecs.yorku.ca/course/2001N`
Also on Moodle

Other Questions on Infinite Sets

- The set $\mathbb{N}$ is countable by definition. So a proof showing it is uncountable (using diagonalization) must fail. But where does it fail?
- We showed that $\mathcal{P}(\mathbb{N})$ (and $\mathbb{R}$) are uncountable. What about $\mathcal{P}(\mathbb{R})$?
- What about $\mathcal{P}(\mathcal{P}(\mathbb{R}))$?
- Can we build bigger and bigger infinities this way?
  Cantor's Continuum hypothesis – YES!

Back to TM's and Languages

- We showed that the set of languages is not countable
- We showed that the set of TM's is countable
- So there are many languages that are not Turing recognizable
- Are there interesting languages for which we can prove that there is no Turing machine that recognizes it?

Our First Undecidable Language

The acceptance problem for Turing Machines:
$A_{TM} = \{\langle M, w\rangle | M \text{ is a TM that accepts } w\}$
Theorem: A_{TM} is undecidable

- Proof by contradiction: Assume that TM G decides A_{TM}
- So G is as follows

$$\begin{aligned} G(\langle M, w\rangle) &= \text{“accept” if } M \text{ accepts } w \\ &= \text{“reject” if } M \text{ does not accept } w \end{aligned}$$

- From G we construct a new TM D that will get us into trouble...

Our First Undecidable Language - 2

Design a new TM D that takes as input a TM M as follows

- D runs TM G on input $\langle M, \langle M \rangle \rangle$
- Disagree on the answer of G
- Note that D always terminates because G always terminates
- So in short,

$$\begin{aligned} D(\langle M \rangle) &= \text{“accept” if } G \text{ rejects } \langle M, \langle M \rangle \rangle \\ &= \text{“reject” if if } G \text{ accepts } \langle M, \langle M \rangle \rangle \end{aligned}$$

- So,

$$\begin{aligned} D(\langle M \rangle) &= \text{“accept” if } M \text{ rejects } \langle M, \rangle \\ &= \text{“reject” if if } M \text{ accepts } \langle M \rangle \end{aligned}$$

Our First Undecidable Language - 3

- Recall,

$$\begin{aligned} D(\langle M \rangle) &= \text{“accept” if } M \text{ rejects } \langle M \rangle \\ &= \text{“reject” if if } M \text{ accepts } \langle M \rangle \end{aligned}$$

- Now run D on itself (i.e., $\langle D \rangle$)
- Result:,

$$\begin{aligned} D(\langle D \rangle) &= \text{“accept” if } D \text{ rejects } \langle D \rangle \\ &= \text{“reject” if if } D \text{ accepts } \langle D \rangle \end{aligned}$$

- This makes no sense: D only accepts if it rejects, and vice versa
- This is a contradiction, therefore A_{TM} is undecidable

Viewing the Last Proof as Diagonalization

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$	$\langle D \rangle$	$\cdots$
M_1	accept	reject	accept	reject			
M_2	accept	accept	accept	accept			
M_3	reject	reject	reject	reject	$\ldots$		
M_4	accept	accept	reject	reject			
$\vdots$			$\vdots$		$\ddots$		
D	reject	reject	accept	accept	$\cdots$	?	
$\vdots$							

- This is an instance of self-referencing by a program
- This is sometimes natural - a character counting program can run on itself

Self-referencing Problems

- Some such problems are decidable
  - How big is $\langle M \rangle$?
  - Is $\langle M \rangle$ a proper TM?
- Others are not
  - Does $\langle M \rangle$ halt or not?
  - Is there a smaller program M' that is equivalent?

Turing Unrecognizability

- A_{TM} is not TM-decidable, but it is TM-recognizable. Wy?
- Is there a language that is not TM-recognizable?
- A useful result:
  Theorem: If a language A is TM-recognizable and its complement $\overline{A}$ is recognizable, then A is TM-decidable.
- Proof: Run the recognizing TMs for A and in parallel on input x. Wait for one of the TMs to accept. If the TM for A accepted: "accept x"; if the TM for $\overline{A}$ accepted: "reject x"

$\overline{A}_{TM}$ is not TM-recognizable

- By the previous theorem it follows that $\overline{A}_{TM}$ cannot be TM-recognizable, because this would imply that A_{TM} is TM decidable

- We call languages like $\overline{A}_{TM}$ co-TM recognizable

Other Languages that are not TM-recognizable

- $E_{TM} = \{\langle G \rangle | G \text{ is a TM with } L(G) = \emptyset\}$
  - This is co-TM recognizable
    Obvious strategy: if the language is non-empty, we can find the first string that is accepted ...
  - Is it TM-recognizable (and thus decidable)?
    Answer turns out to be NO
- $EQ_{TM} = \{\langle G, H \rangle | G, H \text{ are TM's with } L(G) = L(H)\}$
  - Is this co-TM recognizable?
  - Is it TM-recognizable?
  - Turns out both answers are NO

We need more tools to reason about these languages