BU CS 332 – Theory of Computation

Lecture 14:

- Countability
- Undecidable Languages

Reading:

Sipser Ch 4.2

Mark Bun

March 18, 2020

Last Time

A_{DFA} decidable	A_{CFG} decidable
E_{DFA} decidable	E_{CFG} decidable
EQ_{DFA} decidable	

One more decidable language (Sipser 4.10)

Theorem: $INFINITE_{\mathrm{DFA}} = \{\langle D \rangle \mid D \text{ is a DFA that accepts infinitely many strings}\}$ is decidable

Lemma: Let D be a DFA with n states. Then D accepts infinitely many strings if and only if D accepts some string of length $\geq n$.

One more decidable language (Sipser 4.10)

Theorem: $INFINITE_{\text{DFA}} =$
$\{\langle D \rangle \mid D \text{ is a DFA that accepts infinitely many strings}\}$
is decidable

Proof: The following TM decides $INFINITE_{\text{DFA}}$:

On input $\langle D \rangle$, where D is a DFA with n states:

1. Let A be a DFA recognizing all strings of length $\geq n$
2. Let B be a DFA recognizing $L(A) \cap L(D)$
3. Run the decider for E_{DFA} on input $\langle B \rangle$. Accept if it *rejects*, and reject if it *accepts*.

Problems in Language Theory

A_{DFA} decidable	A_{CFG} decidable	A_{TM} ?
E_{DFA} decidable	E_{CFG} decidable	E_{TM} ?
EQ_{DFA} decidable	EQ_{CFG} ?	EQ_{TM} ?

Undecidability

These natural computational questions about computational models are **undecidable**

I.e., computers cannot solve these problems no matter how much time they are given

Countability and Diagonalizaiton

Set Theory Review

A function $f: A \to B$ is

- 1-to-1 (injective) if $f(a) \neq f(a')$ for all $a \neq a'$
- onto (surjective) if for all $b \in B$, there exists $a \in A$ such that $f(a) = b$
- a correspondence (bijective) if it is 1-to-1 and onto, i.e., every $b \in B$ has a unique $a \in A$ with $f(a) = b$

How can we compare sizes of infinite sets?

Definition: Two sets have the same size if there is a bijection between them

A set is countable if

- it is a finite set, or
- it has the same size as $\mathbb{N}$, the set of natural numbers

Examples of countable sets

- $\emptyset$
- $\{0,1\}$
- $\{0, 1, 2, \dots 8675309\}$

- $E \;=\; \{2, 4, 6, 8, \dots\}$
- $SQUARES = \{1, 4, 9, 16, 25, \dots\}$
- $POW2 = \{1, 2, 4, 8, 16, 32, \dots\}$

$$|E| = |SQUARES| = |POW2| = |\mathbb{N}|$$

How to show that $\mathbb{N} \times \mathbb{N}$ is countable?

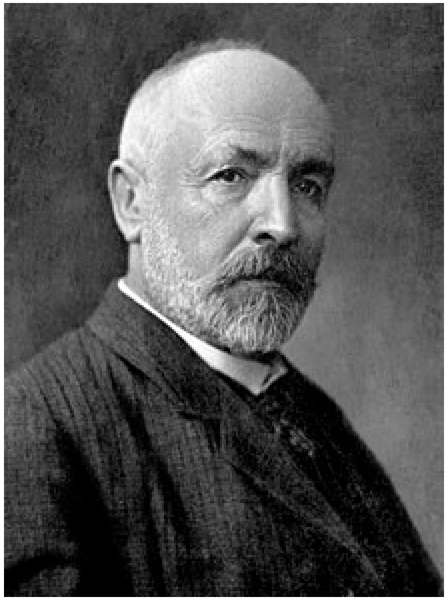
$$
\begin{array}{cccccc}
(1,1) & (2,1) & (3,1) & (4,1) & (5,1) & \dots \\
(1,2) & (2,2) & (3,2) & (4,2) & (5,2) & \dots \\
(1,3) & (2,3) & (3,3) & (4,3) & (5,3) & \dots \\
(1,4) & (2,4) & (3,4) & (4,4) & (5,4) & \dots \\
(1,5) & (2,5) & (3,5) & (4,5) & (5,5) & \\
 & & & & & \ddots
\end{array}
$$

More examples of countable sets

- $\{0,1\}^*$
- $\{\langle M \rangle \mid M \text{ is a Turing machine}\}$
- $\mathbb{Q} = \{\text{rational numbers}\}$

So what *isn't* countable?

Cantor's Diagonalization Method



Georg Cantor 1845-1918

- Invented set theory
- Defined countability, uncountability, cardinal and ordinal numbers, …

Some praise for his work:

"Scientific charlatan…renegade…corruptor of youth"
–L. Kronecker

"Set theory is wrong…utter nonsense…laughable"
–L. Wittgenstein

Sylvester Medal, Royal Society, 1904

Uncountability of the reals

Theorem: The real interval $(0, 1)$ is uncountable.

Proof: Assume for the sake of contradiction it were countable, and let $f: \mathbb{N} \to (0,1)$ be a correspondence

n	$f(n)$
1	$0 . d_1^1\ d_2^1\ d_3^1\ d_4^1\ d_5^1 \dots$
2	$0 . d_1^2\ d_2^2\ d_3^2\ d_4^2\ d_5^2 \dots$
3	$0 . d_1^3\ d_2^3\ d_3^3\ d_4^3\ d_5^3 \dots$
4	$0 . d_1^4\ d_2^4\ d_3^4\ d_4^4\ d_5^4 \dots$
5	$0 . d_1^5\ d_2^5\ d_3^5\ d_4^5\ d_5^5 \dots$

Construct $b \in (0,1)$ which does not appear in this table

– contradiction!

$b = 0. d_1 d_2 d_3 \dots$ where $d_i \neq$ digit i of $f(i)$

Uncountability of the reals

A concrete example:

n	$f(n)$
1	$0.8675309\ldots$
2	$0.1415926\ldots$
3	$0.7182818\ldots$
4	$0.4444444\ldots$
5	$0.1337133\ldots$

Construct $b \in (0,1)$ which does not appear in this table – contradiction!

$b = 0.d_1 d_2 d_3 \ldots$ where $d_i \neq$ digit i of $f(i)$

Diagonalization

This process of constructing a counterexample by "contradicting the diagonal" is called diagonalization

What if we try to do this with the rationals?

What happens if we try to use this argument to show that $\mathbb{Q} \cap (0,1)$ [rational numbers in $(0,1)$] is uncountable?

Let $f: \mathbb{N} \to \mathbb{Q} \cap (0,1)$ be a correspondence

n	$f(n)$
1	$0.8678678\ldots$
2	$0.1414141\ldots$
3	$0.7182718\ldots$
4	$0.4444444\ldots$
5	$0.1337133\ldots$

Construct $b \in (0,1)$ which does not appear in this table

$b = 0.d_1 d_2 d_3\ldots$ where $d_i \neq$ digit i of $f(i)$

A general theorem about set sizes

Theorem: Let X be a set. Then the power set $P(X)$ does **not** have the same size as X.

Proof: Assume for the sake of contradiction that there is a correspondence $f: X \rightarrow P(X)$

Goal: Construct a set $S \in P(X)$ that cannot be the output $f(x)$ for any $x \in X$

Diagonalization argument

Assume a correspondence $f: X \to P(X)$

x					
x_1					
x_2					
x_3					
x_4					
$\vdots$					

Diagonalization argument

Assume a correspondence $f: X \to P(X)$

x	$x_1 \in f(x)$?	$x_2 \in f(x)$?	$x_3 \in f(x)$?	$x_4 \in f(x)$?	…
x_1	Y	N	Y	Y	
x_2	N	N	Y	Y	
x_3	Y	Y	Y	N	
x_4	N	N	Y	N	
$\vdots$					$\ddots$

Define S by flipping the diagonal:

Put $x_i \in S \iff x_i \notin f(x_i)$

Example

Let $X = \{1, 2, 3\}$, $P(X) = \{\emptyset, \{1\}, \{2\}, \{1,2\}, \{2,3\}, \{1,2,3\}\}$

Ex. $f(1) = \{1, 2\}$, $f(2) = \emptyset$, $f(3) = \{2\}$

x	$1 \in f(x)?$	$2 \in f(x)?$	$3 \in f(x)?$	…
1				
2				
3				
$\vdots$				$\ddots$

Construct $S =$

A general theorem about set sizes

Theorem: Let X be a set. Then the power set $P(X)$ does **not** have the same size as X.

Proof: Assume for the sake of contradiction that there is a correspondence $f: X \to P(X)$

Construct a set $S \in P(X)$ that cannot be the output $f(x)$ for any $x \in X$:

$$S = \{x \in X \mid x \notin f(x)\}$$

If $S = f(y)$ for some $y \in X$,

then $y \in S$ if and only if $y \notin S$

Undecidable Languages

An Existential Proof

Theorem: There exists an undecidable language over $\{0,1\}$

Proof:

A simplifying assumption: Every string in $\{0,1\}^*$ is the encoding $\langle M \rangle$ of some Turing machine M

Set of all Turing machines: $X = \{0,1\}^*$

Set of all languages over $\{0,1\}$ = all subsets of $\{0,1\}^*$

= $P(X)$

There are more languages than there are TM deciders!

An Existential Proof

Theorem: There exists an unrecognizable language over $\{0, 1\}$

Proof:

A simplifying assumption: Every string in $\{0, 1\}^*$ is the encoding $\langle M \rangle$ of some Turing machine M

Set of all Turing machines: $X = \{0, 1\}^*$

Set of all languages over $\{0, 1\}$ = all subsets of $\{0, 1\}^*$

= $P(X)$

There are more languages than there are TM recognizers!

A Specific Undecidable Language

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

Theorem: A_{TM} is undecidable

But first: A_{TM} *is* Turing-recognizable

The following "universal TM" U recognizes A_{TM}

On input $\langle M, w \rangle$:

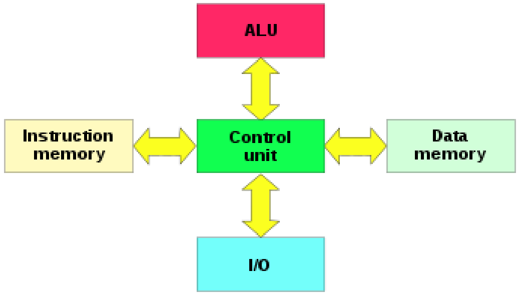
1. Simulate running M on input w
2. If M accepts, accept. If M rejects, reject.

More on the Universal TM

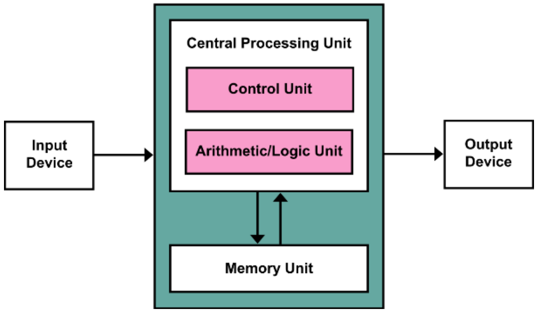
"It is possible to invent a single machine which can be used to compute any computable sequence. If this machine **U** is supplied with a tape on the beginning of which is written the S.D ["standard description"] of some computing machine **M**, then **U** will compute the same sequence as **M**."

- Turing, "On Computable Numbers…" 1936

- Foreshadowed general-purpose programmable computers
- No need for specialized hardware: Virtual machines as software



Harvard architecture:
Separate instruction and data pathways



von Neumann architecture:
Programs can be treated as data

A Specific Undecidable Language

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

Theorem: A_{TM} is undecidable

Proof: Assume for the sake of contradiction that TM H decides A_{TM}:

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

Diagonalization: Use H to check what M when given as input its own description…and do the opposite

A Specific Undecidable Language

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

Suppose H decides A_{TM}

Consider the following TM D.

On input $\langle M\rangle$ where M is a TM:

1. Run H on input $\langle M, \langle M\rangle\rangle$
2. If H accepts, reject. If H rejects, accept.

Question: What does D do on input $\langle D\rangle$?

How is this diagonalization?

TM M					
M_1					
M_2					
M_3					
M_4					
⋮					

How is this diagonalization?

TM M	$M(\langle M_1 \rangle)$?	$M(\langle M_2 \rangle)$?	$M(\langle M_3 \rangle)$?	$M(\langle M_4 \rangle)$?	…
M_1	Y	N	Y	Y	
M_2	N	N	Y	Y	
M_3	Y	Y	Y	N	
M_4	N	N	Y	N	
$\vdots$					$\ddots$

D accepts input $\langle M_i \rangle$ $\Leftrightarrow$ M_i does not accept input $\langle M_i \rangle$

How is this diagonalization?

TM M	$M(\langle M_1 \rangle)$?	$M(\langle M_2 \rangle)$?	$M(\langle M_3 \rangle)$?	$M(\langle M_4 \rangle)$?		$D(\langle D \rangle)$?
M_1	Y	N	Y	Y	…	
M_2	N	N	Y	Y		
M_3	Y	Y	Y	N		
M_4	N	N	Y	N		
$\vdots$					$\ddots$	
D						

D accepts input $\langle M_i \rangle$ $\iff$ M_i does not accept input $\langle M_i \rangle$

$A_{\mathrm{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM that accepts input } w\}$

On input $\langle M, w\rangle$:

1. Simulate running M on input w
2. If M accepts, accept. If M rejects, reject.