BU CS 332 – Theory of Computation

Lecture 14:

- Decidable problems re DFAs, CFGs
- Unrecognizability
- Undecidability

Reading:

Sipser Ch. 3.2, 4

Ran Canetti

October 22, 2020

Church-Turing *Thesis v. I*: The TM model captures our intuitive notion of a computational algorithm.

Church-Turing *Thesis v. I*: The TM model captures our intuitive notion of a computational algorithm.

Church-Turing *Thesis v. II*: Any physical computation process can be simulated on a TM.

Church-Turing *Thesis v. I*: The TM model captures our intuitive notion of a computational algorithm.

Church-Turing *Thesis v. II*: Any physical computation process can be simulated on a TM.

The Church-Turing Thesis is not a mathematical statement!

Universal computation

A universal algorithm for a computational model is an algorithm U that takes a description $\langle A, x \rangle$ of an algorithm A and an input x in that model, and outputs $A(x)$, i.e. the result of running A on x.

Universal computation

A universal algorithm for a computational model is an algorithm U that takes a description $\langle A, x \rangle$ of an algorithm A and an input x in that model, and outputs $A(x)$, i.e. the result of running A on x.

We saw:

- A universal DFA: U_{DFA} : $U(\langle DFA, x \rangle) = DFA(x)$

Universal computation

A universal algorithm for a computational model is an algorithm U that takes a description $\langle A, x\rangle$ of an algorithm A and an input x in that model, and outputs $A(x)$, i.e. the result of running A on x.

We saw:

- A universal DFA: $U_{\mathrm{DFA}}: \; U(\langle DFA, x\rangle) = DFA(x)$
- A universal CFG: $U_{\mathrm{DFA}}: \; U(\langle CFG, Der\rangle) = CFG(Der)$ w

(Der = Derivation tree)

Universal computation

A universal algorithm for a computational model is an algorithm U that takes a description $\langle A, x\rangle$ of an algorithm A and an input x in that model, and outputs $A(x)$, i.e. the result of running A on x.

We saw:

- A universal DFA: $U_{\text{DFA}}: \quad U(\langle DFA, x\rangle) = DFA(x)$
- A universal CFG: $U_{\text{DFA}}: \quad U(\langle CFG, Der\rangle) = CFG(Der)$

  (Der = Derivation tree)
- A universal TM: $U_{\text{TM}}: \quad U(\langle M, x\rangle) = M(x)$

Representation independence

- Two representations of a computational task are equivalent if there is an algorithmic way to translate each representation to the other:

$\Rightarrow \exists T$

  - $x \in L \leftrightarrow T(x) \in L'$
  - Both T and T^{-1} are computable.

Decidable languages

- $A_{\text{DFA}} = \{\langle D, w\rangle \mid \text{DFA } D \text{ accepts } w\}$ is decidable

- $A_{\text{NFA}} = \{\langle N, w\rangle \mid \text{NFA } N \text{ accepts } w\}$ is decidable

  DFAs

- $A_{\text{CFG}} = \{\langle G, w\rangle \mid \text{CFG } G \text{ generates } w\}$ is decidable

How about other questions?

$$E_{\mathrm{DFA}} = \{\langle D\rangle \mid D \text{ is a DFA},\ L(D) = \emptyset\}$$

How about other questions?

$E_{\mathrm{DFA}} = \{\langle D\rangle \mid D \text{ is a DFA},\ L(D) = \emptyset\}$ Decidable

$E_{\mathrm{NFA}} = \{\langle D\rangle \mid D \text{ is an NFA},\ L(D) = \emptyset\}$ Decidable

How about other questions?

$$E_{\text{DFA}} = \{\langle D\rangle \mid D \text{ is a DFA},\ L(D) = \emptyset\}$$

$$E_{\text{NFA}} = \{\langle D\rangle \mid D \text{ is an NFA},\ L(D) = \emptyset\}$$

$$EQ_{\text{DFA}} = \{\langle D_1, D_2\rangle \mid D_1, D_2 \text{ DFAs},\ L(D_1) = L(D_2)\}$$

How about other questions?

$$E_{\mathrm{DFA}} = \{\langle D \rangle \mid D \text{ } is \text{ } a \text{ } DFA, \ L(D) = \emptyset\}$$

$$E_{\mathrm{NFA}} = \{\langle D \rangle \mid D \text{ } is \text{ } an \text{ } NFA, \ L(D) = \emptyset\}$$

$$EQ_{\mathrm{DFA}} = \{\langle D_1, D_2 \rangle \mid D_1, D_2 \ \ DFAs, \ L(D_1) = L(D_2)\}$$

$$E_{\mathrm{CFG}} = \{\langle G \rangle \mid \mathrm{G} \text{ } is \text{ } a \text{ } CFG, \ L(G) = \emptyset\}$$

How about other questions?

$$E_{\text{DFA}} = \{\langle D\rangle \,|D \text{ } is \text{ } a \text{ } DFA,\ L(D) = \emptyset\}$$

$$E_{\text{NFA}} = \{\langle D\rangle \,|D \text{ } is \text{ } an \text{ } NFA,\ L(D) = \emptyset\}$$

$$EQ_{\text{DFA}} = \{\langle D_1, D_2\rangle \,|D_1, D_2 \ \ DFAs,\ L(D_1) = L(D_2)\}$$

$$E_{\text{CFG}} = \{\langle G\rangle \,|\text{G } is \text{ } a \text{ } CFG,\ L(G) = \emptyset\}$$ Decidable

L(G): ∃ p ∀ x ∈ L be partitioned to x = uvwyz s.t. uwz ∈ L

$$NEQ_{\text{CFG}} = \{\langle G_1, G_2\rangle \,|G_1, G_2 \ \ CFGs,\ L(G_1) \neq L(G_2)\}$$ Recognizable

G_2

decidable?

How about questions on TMs?

$A_{\mathrm{TM}} = \{\langle M, x\rangle \mid TM\ M\ accepts\ x\}$ recognizable

decidable?

How about questions on TMs?

$A_{\mathrm{TM}} = \{\langle M, x\rangle \mid TM\ M\ accepts\ x\}$ recognize dec?

$E_{\mathrm{TM}} = \{\langle M\rangle \mid \mathrm{M}\ is\ a\ TM,\ L(M) = \emptyset\}$? ?

Summary

A_{DFA} ~~decidable~~	A_{CFG} decidable	A_{TM} Recog ?
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

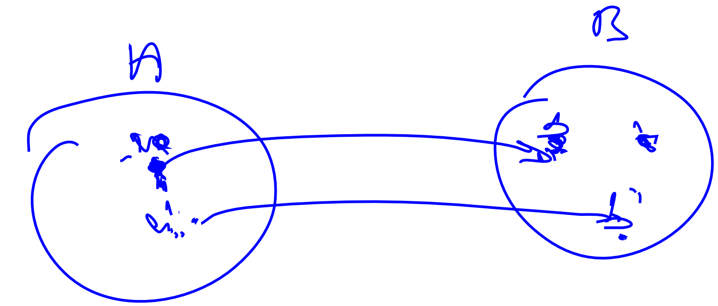
Set Theory Review

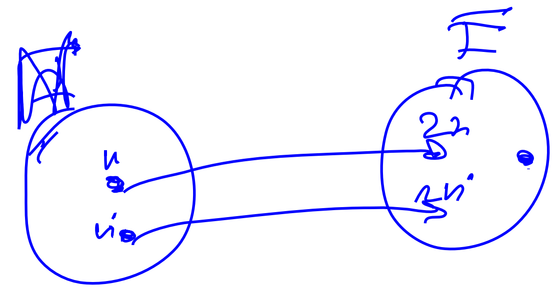
A function $f: A \to B$ is

- 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 $f(a) = b$



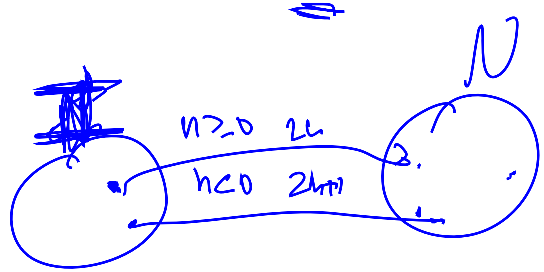


How can we compare sizes of infinite sets?

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

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

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

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

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

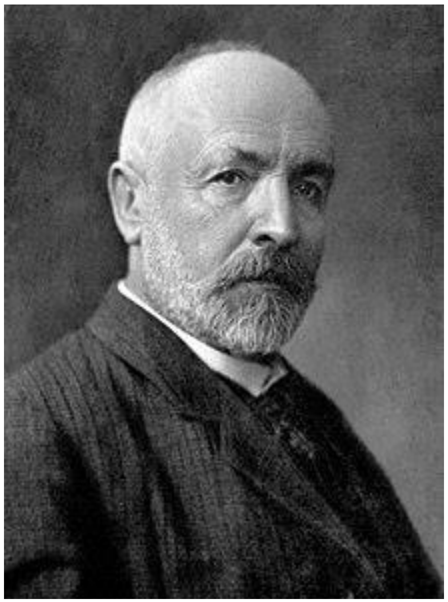
$(1,1)$	$(2,1)$	$(3,1)$	$(4,1)$	$(5,1)$	…
$(1,2)$	$(2,2)$	$(3,2)$	$(4,2)$	$(5,2)$	…
$(1,3)$	$(2,3)$	$(3,3)$	$(4,3)$	$(5,3)$	…
$(1,4)$	$(2,4)$	$(3,4)$	$(4,4)$	$(5,4)$	…
$(1,5)$	$(2,5)$	$(3,5)$	$(4,5)$	$(5,5)$	
					$\ddots$

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	$0 . d_1^1 \, d_2^1 \, d_3^1 \, d_4^1 \, d_5^1 \ldots$
2	$0 . d_1^2 \, d_2^2 \, d_3^2 \, d_4^2 \, d_5^2 \ldots$
3	$0 . d_1^3 \, d_2^3 \, d_3^3 \, d_4^3 \, d_5^3 \ldots$
4	$0 . d_1^4 \, d_2^4 \, d_3^4 \, d_4^4 \, d_5^4 \ldots$
5	$0 . d_1^5 \, d_2^5 \, d_3^5 \, d_4^5 \, d_5^5 \ldots$

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 $f(i)$

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!

$b = 0.d_1d_2d_3\ldots$ 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 . 8 6 7 8 6 7 8 …
2	0 . 1 4 1 4 1 4 1 …
3	0 . 7 1 8 2 7 1 8 …
4	0 . 4 4 4 4 4 4 4 …
5	0 . 1 3 3 7 1 3 3 …

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 $f : X \to P(X)$

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 $f: X \rightarrow P(X)$

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 $\quad x_i \in S \quad \Longleftrightarrow \quad x_i \notin f(x_i)$

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 $\quad S =$

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 $y \notin S$

countable sets

x_0 x_1 x_2 . ...

Undecidable Languages

An Existential Proof

$|\{0,1\}^*| = \aleph_0$

$|L(\{0,1\})| = |P(\{0,1\}^*)| = \aleph_1$

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

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

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

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					
$\vdots$					

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 $\langle M_i \rangle \iff M_i$ 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 $\langle M_i \rangle$

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

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

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