COMPSCI 501: Formal Language Theory

Lecture 15: Undecidability

Marius Minea
`marius@cs.umass.edu`

University of Massachusetts Amherst

25 February 2019

Review: Decidable problems

For regular and context-free languages:

Acceptance problem

Given a pair of finite automaton / regular expression / grammar, and a string, is the string accepted / generated ?

Emptiness problem

Is a language defined by a finite automaton / regular expression / grammar empty ?

Equivalence Problem

Do two finite automata / regular expressions recognize the same language? **decidable**

Do two CFGs generate the same language ? **undecidable**

Acceptance Problem for Turing Machines

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

A_{TM} is **Turing-recognizable**: simulate M on input w, accept and reject accordingly.

Universal Turing machine: can simulate any Turing machine
think: general purpose processor

Countable sets

We say two sets A and B are the same size if there exists a **bijective** function $f : A \to B$ (a **correspondence**) between them
injective (one-to-one): $x \neq y \to f(x) \neq f(y)$
surjective (onto) $\forall y \in B \exists x \in A : f(x) = y$

Can now compare sizes also for **infinite** sets.

A set is **countable** if it is finite or has the same size as $\mathbb{N}$.
equivalently: the same size as some subset of $\mathbb{N}$
or: the is a one-to-one function from the set to $\mathbb{N}$

Diagonalization: Rationals are Countable

1/1	1/2	1/3	1/4	...
2/1	2/2	2/3	2/4	...
3/1	3/2	3/3	3/4	...
...	...	...	... ...	

Count elements by diagonal (m/n with $m + n$ constant):
$1/1, \quad 1/2, 2/1, \quad 1/3, 2/2, 3/3, \quad 1/4, 2/3, 3/2, 4/1, \ldots$
ignore repetitions, they don't matter

Closure of Countable Sets

The union of two countable sets is countable

A countable union of countable sets is countable
C countable, A_i countable for $i \in C \Rightarrow \bigcup_{i\in C} A_i$ is countable

The cartesian product $A \times B$ of two countable sets is countable
same as for rationals

If A is countable, $A^* = \bigcup_{k\geq 0} A^k$ is countable

Diagonalization: Reals are Uncountable

Assume reals were countable and enumerate all reals in interval [0, 1) (in decimal notation)

$$\begin{array}{lllll} r_1 = 0. & d_{11} & d_{12} & d_{13} & \ldots & 0.12749\ldots \\ r_2 = 0. & d_{21} & d_{22} & d_{23} & \ldots & 0.2340567\ldots \\ r_3 = 0. & d_{31} & d_{32} & d_{33} & \ldots & 0.35790856\ldots \\ \ldots & \ldots & \ldots & & & \end{array}$$

Construct another real number $\mathbf{x} = 0.d_1d_2d_3\ldots$ digit by digit:
let d_i be **different** from d_{ii} (i^{th} digit of i^{th} number.

For instance, take $d_i = d_{ii} - 1$ if $d_{ii} > 0$ and $d_i = 1$ if $d_{ii} = 0$
(avoid duplicate represention with 9: 0.1999... = 0.2)

We get a number $\mathbf{x}$ which is not in the sequence
$\Rightarrow$ reals are uncountable

Cantor's Theorem: $|X| < |\mathcal{P}(X)|$

There is no bijection between any set and its powerset.

Easy to see for finite sets, $n < 2^n$. In general:

Assume a bijection $f : X \to \mathcal{P}(X)$ exists.
Construct the set: $Y = \{x \in X \mid x \notin f(x)\}$

Since $Y \in \mathcal{P}(X)$, and f is bijective, there is $y \in X$ with $f(y) = Y$.
We examine whether $y \in Y$, or equivalently, $y \in f(y)$.

If $y \in Y$, since $Y = f(y)$, we get $y \in f(y)$, so y does not observe the required condition for Y, contradiction.

If $y \notin Y$, then $y \notin f(y)$ so y observes the condition for Y, and we should have $y \in Y$, contradiction.

Thus a bijection cannot exist.

The set of all languages is uncountable

From Cantor's theorem, Σ^* countable, $\mathcal{P}(\Sigma^*)$ is not.

Or: The set $\mathcal{B}$ of all **infinite** binary sequences is uncountable
$\{0,1\}^\omega$: corresponds to set of all real numbers in [0, 1]

Let $\mathcal{L} = \mathcal{P}(\Sigma^*)$ be the set of all languages over Σ.
Construct a bijection from $\mathcal{L}$ to $\mathcal{B}$.

Let $\Sigma^* = \{s_1, s_2, s_3, \ldots\}$.
Construct **characteristic sequence** of language $A \in \Sigma^*$:
bit $b_i = 1$ if $s_i \in A$, and $b_i = 0$ if $s_i \notin A$.

Clearly this is a bijection. So $\mathcal{L}$ is uncountable.

There are more languages than Turing machines
$\Rightarrow$ some languages are not **recognizable** (thus also undecidable).

A_{TM} is Undecidable

$A_{\mathsf{TM}} = \{\langle B, w\rangle \mid B \text{ is a TM that accepts string } w\}$

Assume there were a Turing machine H that decides A_{TM}:

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

M can be encoded, so can run $H(\langle M, \langle M\rangle\rangle)$
Construct TM D that calls H and then does the oopposite:
D: 1. run H on input $\langle M, \langle M\rangle\rangle$
2. if H accepts, reject; if H rejects, accept

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

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

D does the opposite of itself, contradiction! $\Rightarrow$ H cannot exist.

Connection to 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*		*accept*	
M_2	*accept*	*accept*	*accept*	*accept*	$\cdots$	*accept*	$\cdots$
M_3	*reject*	*reject*	*reject*	*reject*		*reject*	
M_4	*accept*	*accept*	*reject*	*reject*		*accept*	
$\vdots$			$\vdots$		$\ddots$		
D	*reject*	*reject*	*accept*	*accept*		?	
$\vdots$			$\vdots$				$\ddots$

FIGURE 4.21
If D is in the figure, a contradiction occurs at "?"

A Turing-Unrecognizable Language

Recognizing = *accept* for sure; may or may not *reject*

Reverse roles: 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.

"$\Rightarrow$": A decidable $\Rightarrow$ $\overline{A}$ decidable (reverse accept/reject)
$\Rightarrow$ both are recognizable

"$\Leftarrow$": Let M_1 be recognizer for A and M_2 recognizer for $\overline{A}$.
run M_1 and M_2 in parallel
if M_1 accepts, accept; if M_2 accepts, reject (one will happen)

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

If it were, since A_{TM} is recognizable, it would be decidable!