Assignment Project Exam Help

Computational Complexity and Computability

Lecture 6 - UTM & Diagonalization

https://powcoder.com

Koushik Pal

University of Toronto

Add WeChat powcoder

January 27, 2021

Turing-recognizable and Turing-decidable

Definition

A language is called Turing-recognizable / semi-decidable / recursively enumerable if some TM **recognizes** it.

A TM is called a decider if it halts on all inputs $x \in \Sigma^*$.

A language is called Turing-decidable / decidable / recursive if some TM **decides** it.

Notation

- $D = \{A \subseteq \Sigma^* \mid A \text{ is decidable}\}$
- $SD = \{A \subseteq \Sigma^* \mid A \text{ is semi-decidable}\}$

Decidability vs NTM

Theorem

A language $\mathcal{L}$ is decidable iff some NTM N decides it.

Proof.

$(\longrightarrow)$

If $\mathcal{L}$ is decidable, it is decided by some TM, and since any TM is by default an NTM, $\mathcal{L}$ is decided by an NTM as well.

Decidability vs NTM

$(\longleftarrow)$
Construct a TM M from N as before, with the additional requirement: *Reject an input w if all branches are exhausted.*

If N accepts w, it is accepted by some branch of N and consequently by M.

If N rejects w, it is rejected by all branches of N since N is a decider. Consequently, each branch has finitely many nodes. Moreover, since each node has finitely many children as well, the whole tree is finite. As a result, M can run an exhaustive search and finally reject w. $\square$

Remark

Similar results hold for other TM variants.

Universal Turing Machine (UTM)

Definition

A UTM is a reprogrammable machine that can simulate any other TM M.

The input to a UTM is a description of transitions of M, states of M and initial tape contents of M. UTM tracks these three things on three tapes:

Tape 1 : Description of M

Tape 2 : States of M

Tape 3 : Tape content of M

To track these things, we need some mechanism for encoding a TM. We use the alphabet $\{0, 1\}$ for encoding a TM.

Encoding of a TM

Alphabet Encoding

Encode $\Gamma = \{a_1, a_2, a_3, \ldots, a_m\}$ as $\{1, 11, 111, \ldots, \underbrace{11\ldots1}_{m}\}$.

State Encoding

Encode $Q = \{q_1, q_2, q_3, \ldots, q_n\}$ as $\{1, 11, 111, \ldots, \underbrace{11\ldots1}_{n}\}$.

Head Move Encoding

Encode $\{L, R\}$ as $\{1, 11\}$.

Transition Encoding

Encode $\delta(q_1, a_1) = (q_2, a_3, R)$ as 1010110111011, with 0 as delimiter.

Machine Encoding

Encode multiple transition rules by separating their individual encodings by the delimiter 00. For example, $\{\delta(q_1, a_1) = (q_2, a_3, R), \delta(q_2, a_1) = (q_1, a_2, L)\}$ is encoded as $1010110111011\,00\,11010101101$.

Encoding of a TM

A TM is, therefore, described as a string over the alphabet $\Sigma = \{0, 1\}$.

Consequently, the set of TMs forms a language $\mathcal{L}_{TM}$ over the alphabet $\{0, 1\}$, each string of which is the binary encoding of some TM.

Countable Sets

Definition

A set S is countable iff there is a surjective function $f : \mathbb{N} \to S$.

Example

1. Any finite set is countable.
2. The set of odd numbers is countable. $(f(n) = 2n + 1)$
3. The set of multiples of 5 is countable. $(f(n) = 5n)$
4. The set of integers is countable. $(f(n) = (-1)^n \lfloor \frac{n+1}{2} \rfloor)$
5. The set of (positive) rational numbers is countable.

$$\begin{array}{cccc} \frac{1}{1} & \frac{2}{1} & \frac{3}{1} & \frac{4}{1} \quad \cdots \\ \frac{1}{2} & \frac{2}{2} & \frac{3}{2} & \cdots \\ \frac{1}{3} & \frac{2}{3} & \cdots & \\ \frac{1}{4} & \cdots & & \end{array}$$

$$f(n) = \frac{\frac{x(1+x)}{2} - n + 1}{\frac{x(1-x)}{2} + n},$$

where $x = \lceil \frac{-1+\sqrt{1+8n}}{2} \rceil$.

SD is countable

Fact

If Σ is a finite set, then Σ^* is countable.

Proof.

Exercise! $\square$

Corollary

The set of all TMs is countable. Consequently, SD is countable as well.

Cantor's Diagonalization

Theorem

$[0,1]$ *is uncountable. Consequently,* $\mathbb{R}$ *is uncountable.*

Proof.

Assume for a contradiction that $[0,1]$ is countable. Consider an enumeration as follows:

$$\begin{array}{llllll}
0. & \color{red}{a_{11}} & a_{12} & a_{13} & a_{14} & a_{15} & \cdots \\
0. & a_{21} & \color{red}{a_{22}} & a_{23} & a_{24} & a_{25} & \cdots \\
0. & a_{31} & a_{32} & \color{red}{a_{33}} & a_{34} & a_{35} & \cdots \\
0. & a_{41} & a_{42} & a_{43} & \color{red}{a_{44}} & a_{45} & \cdots \\
0. & a_{51} & a_{52} & a_{53} & a_{54} & \color{red}{a_{55}} & \cdots \\
\vdots & & & & & &
\end{array}$$

Consider the number $x = 0.\ell_1\ell_2\ell_3\ell_4\ell_5\cdots$ where $\ell_i \neq a_{ii}$ for all i.

Clearly, $x \in [0,1]$, but x is not in the above enumeration. $\lightning$

Cantor's Diagonalization

A very similar proof shows that

Theorem

$\mathcal{P}(S)$ is uncountable for any countably infinite set S.

Corollary

The set of languages over any nonempty finite alphabet Σ (e.g., $\Sigma = \{0, 1\}$) is uncountable.

Proof.

Since Σ is nonempty and finite, Σ^* is countably infinite.

Since a language $\mathcal{L}$ is an arbitrary subset of Σ^*, the set of all languages is $\mathcal{P}(\Sigma^*)$.

By the above theorem, the set of languages over Σ is uncountable. $\square$

Existence of a non-semi-decidable language

Theorem

Given a nonempty finite alphabet Σ, there is a language over Σ that is not semi-decidable

Proof.

We have already shown that the number of TMs, and hence the number of semi-decidable languages over Σ, is countable.

We have also shown that the number of languages over Σ is uncountable.

Hence, there is at least one language over Σ that is not semi-decidable. In fact, there are uncountably many such languages. $\square$

Encoding

Goal : Represent objects as strings to feed to a TM.

Fact

- Strings can easily represent polynomials, graphs, grammars, automata, and any combination of these objects.

- A TM may be programmed to decode the representation so that it can be interpreted in the way we intend.

Notation

We use the notation $\langle O \rangle$ to denote an encoding into a string representation of the object O. For several objects $O_1, \ldots, O_n$, we denote their encoding into a single string as $\langle O_1, \ldots, O_n \rangle$.

Example of a non-semi-decidable language

Definition

$DIAG := \{\langle M \rangle \mid M$ is a TM and $\langle M \rangle \notin \mathcal{L}(M)\}$.

Theorem

$DIAG \notin SD$.

Proof.

Assume for a contradiction that there exists a TM M such that $\mathcal{L}(M) = DIAG$. Then

$$\langle M \rangle \in \mathcal{L}(M) \iff \langle M \rangle \in DIAG \iff \langle M \rangle \notin \mathcal{L}(M). \quad \lightning$$

Example of a semi-decidable but not decidable language

Definition

$A_{TM} := \{\langle M, w\rangle \mid M \text{ accepts } w\}$.

Theorem

$A_{TM} \in SD \setminus D$.

Proof.

$A_{TM} \in SD$ because $A_{TM} = \mathcal{L}(U)$ where U is a UTM that simulates M on w.

$A_{TM} \notin D$ because otherwise $DIAG \in D$ which yields the required contradiction.

$$\langle M\rangle \in DIAG \iff \langle M\rangle \notin \mathcal{L}(M) \iff \langle M, \langle M\rangle\rangle \notin A_{TM}. \quad \square$$

Corollary

$D \subsetneq SD$.