Lecture #8: Proving Undecidability — Part One
Key Concepts

Universal Turing Machines

Following the introduction of a minor additional condition — we will no longer consider Turing machines

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}}) \tag{1}$$

such that either $q_0 = q_{\text{accept}}$ or $q_0 = q_{\text{reject}}$ — an ***encoding*** of a (standard) Turing machine M as a string of symbols over an alphabet

$$\Sigma_{\text{TM}} = \{\texttt{s}, \texttt{q}, \texttt{0}, \texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}, \texttt{7}, \texttt{8}, \texttt{9}, \texttt{L}, \texttt{R}, \texttt{Y}, \texttt{N}, \texttt{(}, \texttt{)}, \texttt{,}, \texttt{;}\}$$

was introduced. Encodings of Turing machines M (with components as at line (1), above) and input strings $\omega \in \Sigma^\star$ were introduced, as well.

Three languages were introduced:

- $L_{\text{TM}} \subseteq \Sigma_{\text{TM}}^\star$ is the language of encodings of Turing machines.
- $L_{\text{TM+I}} \subseteq \Sigma_{\text{TM}}^\star$ is the language of encodings of Turing machines M (with some input alphabet Σ) and input strings $\omega \in \Sigma^\star$.
- $A_{\text{TM}} \subseteq L_{\text{TM+I}}$ is the language of encodings of Turing machines M (with some input alphabet Σ) and input strings $\omega \in \Sigma^\star$ such that M ***accepts*** ω.

The languages L_{TM} and $L_{\text{TM+I}}$ are both ***decidable***.

A ***universal Turing machine*** is a (single tape, or multi-tape) Turing machine M_{UTM}, with input alphabet Σ_{TM}, which satisfies the following properties:

- If $\mu \in \Sigma_{\text{TM}}^\star$ and $\omega \notin L_{\text{TM+I}}$ — so that ω is *not* the encoding of any Turing machine M and input string ω for it — then M_{UTM} ***rejects*** μ.

Suppose, instead, that $\mu \in L_{\mathsf{TM+I}}$ — so that μ is the encoding of some Turing machine M (with an input alphabet Σ) and input string ω for M.

- If M ***accepts*** μ then M_{UTM} ***accepts*** the string, μ, that encodes M and ω.
- If M ***rejects*** μ then M_{UTM} ***rejects*** the string, μ. that encodes M and ω.
- If M ***loops*** on ω then M_{UTM} ***loops*** on the string, μ, that encodes M and ω.

Universal Turings exist — and one such (multi-tape) Turing machine was described in the preparatory material for this lecture.

First Hard Languages

Since A_{TM} is the language of any universal Turing machine, it follows (by the existence of these Turing machines) that the language, A_{TM}, is ***recognizable***.

However, it is possible to prove (by contradiction) that the language A_{TM} is ***undecidable***.

Furthermore, another proof by contradiction can be given to prove that the language

$$A^{C}_{\mathsf{TM}} = \{\mu \in \Sigma^{\star}_{\mathsf{TM}} \mid \mu \notin A_{\mathsf{TM}}\}$$

is ***unrecognizable***.

Reducibilities

Definition 1. A ***reducibility*** is any binary relation $\preceq_{\mathsf{Q}}$ between languages (possibly over different alphabets) such that the following properties are satisfied.

(a) $L \preceq_{\mathsf{Q}} L$ for every language $L \subseteq \Sigma^{\star}$ (and for every alphabet Σ).

(b) For all languages $L_1 \subseteq \Sigma_1^{\star}$, $L_2 \subseteq \Sigma_2^{\star}$ and $L_3 \subseteq \Sigma_3^{\star}$ (and alphabets Σ_1, Σ_2 and Σ_3) if $L_1 \preceq_{\mathsf{Q}} L_2$ and $L_2 \preceq_{\mathsf{Q}} L_3$ then $L_1 \preceq_{\mathsf{Q}} L_3$.

Oracle Reductions

Definitions

Definition 2. An ***oracle for a language*** $\boldsymbol{L \subseteq \Sigma_L^{\star}}$ is a device that is capable of reporting whether any string $\omega \in \Sigma_L^{\star}$ is a member of L.

Definition 3. Let $L_1 \subseteq \Sigma_1^\star$ and $L_2 \subseteq \Sigma_2^\star$. Then L_1 is ***oracle reducible*** to L_2,

$$L_1 \preceq_{\mathsf{O}} L_2,$$

if there exists an algorithm that decides membership in L_1 that uses an oracle (that is, a subroutine) deciding membership in L_2.

Describing an Oracle Reduction Between Languages

To describe an oracle reduction from a language $L_1 \subseteq \Sigma_1^\star$ to a language $L_2 \subseteq \Sigma_2^\star$:

1. Give pseudocode for an algorithm that decides membership in L_1 using a separate ***Boolean method*** that decides membership in L_1 *You do not need to write a method deciding L_2 — just assume that one exists.*
2. If this is not obvious, *prove the correctness* of the algorithm that decides membership in L_1, assuming the correctness of an (unknown) algorithm that decides membership in L_2 and that is used whenever it is needed.
3. Adding more detail (only) as needed, show that *if there exists* a multi-tape Turing machine M_2 that decided membership in L_2 then this can be used as a component in a multi-tape Turing machine M_1 that would decide membership in L_1.

Useful Properties

The set of oracle reductions forms a ***reducibility*** — as this is defined above.

Let $L_1 \subseteq \Sigma_1^\star$ and let $L_2 \subseteq \Sigma_2^\star$, for alphabets Σ_1 and Σ_2. If

- $L_1 \preceq_{\mathsf{O}} L_2$, and
- L_2 is decidable,

then L_1 is decidable as well. That is, the set of decidable languages is ***closed*** under oracle reductions.

Note: It follows that if

- $L_1 \preceq_{\mathsf{O}} L_2$, and
- L_1 is ***undecidable***,

then L_2 is ***undecidable***.

However, the set of recognizable language is ***not*** closed under oracle languages: There exist languages $L_1 \subseteq \Sigma_1^\star$ and $L_2 \subseteq \Sigma_2^\star$ such that

- $L_1 \preceq_{\mathsf{O}} L_2$,

- L_2 is recognizable, but
- L_1 is unrecognizable.

Thus oracle reductions ***cannot*** be used to prove that languages are unrecognizable.

Another Undecidable Language

Let $NA_{\mathsf{TM}} \subseteq \Sigma^{\star}_{\mathsf{TM}}$ be the set of encodings of Turing machines M, and input strings ω for M, such that M ***does not*** accept μ. It was proved (in the preparatory material for this lecture), using an oracle reduction, that the language NA_{TM} is undecidable.