Computer Science 351

Proofs of Undecidability — Examples I

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #18

Goal for Today

- Another proof that a decision problem — and associated language of Yes-instances — is undecidable, using a many-one reduction, will be presented.

Decidable Languages

- TM $\subseteq \Sigma_{\text{TM}}^{\star}$: Valid encodings of Turing machines (whose start state is not a halting state)
- TM+I $\subseteq \Sigma_{\text{TM}}^{\star}$: Valid encodings of Turing machines M and strings of symbols over the input alphabet for M.

These are the ***languages of instances*** for several decision problems being considered.

Undecidable Problems and Languages

Let us say that "a decision problem is undecidable" if its language of Yes-instances is undecidable. It follows from the results included in the lecture material about *universal Turing machines* that the following decision problem is undecidable.

Acceptance Problem

Instance: A Turing machine

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

and an input string $\omega \in \Sigma^\star$

Question: Does M accept ω?

The language of Yes-instances of this problem — an undecidable language — is the language

$$\mathrm{A}_{\mathrm{TM}} \subseteq \mathrm{TM{+}I} \subseteq \Sigma_{\mathrm{TM}}^\star.$$

Undecidable Problems and Languages

It follows from a many-one reduction, described in the lecture presentation about *many-one reductions*, that the following decision problem is also undecidable.

Halting Problem

Instance: A Turing machine

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

and an input string $\omega \in \Sigma^\star$

Question: Does *M* halt when it is executed on input ω?

The language of Yes-instances of this problem — another undecidable language — is the language

$$\text{HALT}_{\text{TM}} \subseteq \text{TM+I} \subseteq \Sigma_{\text{TM}}^\star.$$

“Accepting All” Problem

Consider the following decision problem.

The “Accepting All” Problem

Instance: A Turing machine

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

Question: Does M accept all input strings? That is, is $L(M) = \Sigma^\star$?

Let $\text{All}_{\text{TM}} \subseteq \text{TM}$ be the language of Yes-instances of this problem (whose language of instances is the language $\text{TM} \subseteq \Sigma_{\text{TM}}^\star$).

Reduction To Be Used

- In order to prove that the "Accepting All" problem is undecidable, let us give a many-one reduction from the "Acceptance" problem to the "Accepting All" problem.
- In other words, let us show that $\mathrm{A}_{\mathrm{TM}} \preceq_{\mathrm{M}} \mathrm{All}_{\mathrm{TM}}$.

A Useful Mapping

- To begin, we must describe a mapping φ from Turing machines

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

and input strings $\omega \in \Sigma^\star$ — instances of the "Acceptance" problem — to Turing machines

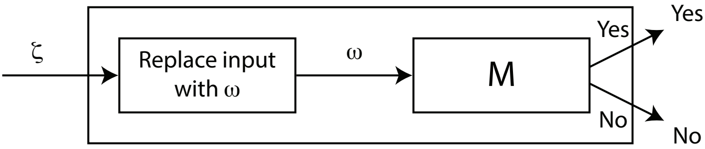
$$\widehat{M} = (\widehat{Q}, \widehat{\Sigma}, \widehat{\Gamma}, \widehat{q}_0, \widehat{q}_{\text{accept}}, \widehat{q}_{\text{reject}})$$

— instances of the "Accepting All" problem — such that, for every Turing machine M and input string ω for M, if $\widehat{M} = \varphi((M, \omega))$ (and $\widehat{\Sigma}$ is the input alphabet for $\widehat{M}$), then

$$M \text{ accepts } \omega \iff L(\widehat{M}) = \widehat{\Sigma}^\star.$$

A Useful Mapping

For a Turing machine M with input alphabet Σ, and a string $\omega \in \Sigma^\star$, let $\varphi((M, \omega)) = \mathcal{M}_{\langle M,\omega \rangle}$, where $\mathcal{M}_{\langle M,\omega \rangle}$ is as follows.



$\mathcal{M}_{\langle M,\, \omega \rangle}$

A Useful Mapping

$\mathcal{M}_{\langle M, \omega \rangle}$ implements the following algorithm:

On input $\zeta \in \Sigma^{\star}$ {

1. Replace ζ with ω on the tape, and enter *M*'s start state (so that *M* is in its initial configuration for input ω).
2. Run *M* (now, with input ω) — *accepting* if *M* eventually accepts ω, *rejecting* if *M* eventually rejects ω, and *looping* otherwise.

}

A Useful Mapping

- $\mathcal{M}_{\langle M, \omega \rangle}$ has the same input alphabet, Σ, as M (so $\widehat{\Sigma} = \Sigma$). Additional details about $\mathcal{M}_{\langle M, \omega \rangle}$ — including its set of states, tape alphabet, and transition function — are given in a supplemental document.
- Assuming $\mathcal{M}_{\langle M, \omega \rangle}$ exists, and implements the above algorithm, the following can be proved, for every Turing machine M (with input alphabet Σ) and string $\omega \in \Sigma^\star$

$$M \text{ accepts } \omega \iff L(\mathcal{M}_{\langle M, \omega \rangle}) = \Sigma^\star.$$

  Furthermore, if M *does not* accept ω then $L(\mathcal{M}_{\langle M, \omega \rangle}) = \emptyset$.
- Thus (since $\varphi((M, \omega)) = \mathcal{M}_{\langle M, \omega \rangle}$) the mapping φ is as required.

Introducing Encodings

- Turing machines, and pairs of Turing machines and input strings, are encoded as described in the lecture material for *universal Turing machines*.
- Thus the language of instances of the “Acceptance” problem is the language TM+I, and the language of instances of the “Accepting All” problem is the language TM.
- The “usual situation”, described in the first set of lecture material for *many-one reductions*, holds: The language, TM+I, of instances of the first problem is decidable, and there exists a string $\mu_{\text{Junk}} \in \Sigma_{\text{TM}}^{\star}$ such that μ_{Junk} does not belong to the language, TM, of instances of the second problem.
- *For this problem,* it is possible to choose μ_{Junk} to be the empty string, λ.

Completion of the Function

Now consider a total function $f : \Sigma_{\text{TM}}^{\star} \to \Sigma_{\text{TM}}^{\star}$ such that the following properties are satisfied, for every string $\mu \in \Sigma_{\text{TM}}^{\star}$:

- If $\mu \in$ TM+I — so that μ is the encoding of a Turing machine M (with some input alphabet Σ) and a string $\omega \in \Sigma^{\star}$ — then $f(\mu)$ is the encoding of the corresponding Turing machine, $\mathcal{M}_{\langle M, \omega \rangle}$, that is described above.
- If $\mu \notin$ TM+I then $f(\mu) = \mu_{\text{Junk}}$ (that is, $f(\mu) = \lambda$).

Completion of the Proof

- It follows from the above that, for every string $\mu \in \Sigma_{\mathsf{TM}}^{\star}$,

$$\mu \in \mathsf{A}_{\mathsf{TM}} \iff f(\mu) \in \mathsf{All}_{\mathsf{TM}}.$$

- As shown in a supplemental document, the function f is computable.
- Thus f is a ***many-one reduction*** from A_{TM} to $\mathsf{All}_{\mathsf{TM}}$ — and from the “Acceptance” problem to the “Accepting All” problem.
- Since A_{TM} is undecidable, this implies that $\mathsf{All}_{\mathsf{TM}}$ is undecidable too. Thus the “Accepting All” problem is undecidable.