Many-One Reductions
Supplement for Preparatory Viewing

First Example of a Many-One Reduction

Let $L, \widehat{L} \subseteq \Sigma^\star$ for an alphabet Σ, such that $L \neq \Sigma^\star$, and suppose that $\widehat{L}$ is ***decidable***. Since $L \neq \Sigma^\star$ there exists a string $\mu_{\text{No}} \in \Sigma^\star$ such that $\mu_{\text{No}} \notin L$.

Let $f : \Sigma^\star \to \Sigma^\star$ such that, for all $\omega \in \Sigma^\star$,

$$f(\omega) = \begin{cases} \omega & \text{if } \omega \in \widehat{L}, \\ \mu_{\text{No}} & \text{if } \omega \notin \widehat{L}. \end{cases}$$

During the lecture it is proved that f is a ***many-one reduction*** from $L \cap \widehat{L}$ to L.

Second Example of a Many-One Reduction

Consider the following ***decision problems***:

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?

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 executed on input ω?

In order to prove that the Halting Problem is reducible to the Acceptance Problem, a mapping φ, from instances of the Halting Problem to instances of the Acceptance Problem was introduced. Let

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

be a Turing machine and let $\omega \in \Sigma^\star$ — so that ω can be thought of as an "input string" for M, and the pair (M, ω) is an instance of the "Halting Problem". The mapping, φ, maps this instance to the instance $(\widehat{M}, \widehat{\omega})$, where

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

— where, for every state $q \in Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}$ and for every symbol $\sigma \in \Gamma$,

$$\widehat{\delta}(q, \sigma) = \begin{cases} \delta(q, \sigma) & \text{if } \delta(q, \sigma) = (r, \tau, m) \text{ where } r \neq q_{\text{reject}}, \\ (q_{\text{accept}}, \tau, m) & \text{if } \delta(q, \sigma) = (q_{\text{reject}}, \tau, m) \end{cases}$$

where $r \in Q$, $\tau \in \Gamma$, and $m \in \{\mathtt{L}, \mathtt{R}\}$ — and where $\widehat{\omega} = \omega$.

During the lecture presentation, a sketch of a proof is given that for every Turing machine M and input string ω for M, if $(\widehat{M}, \widehat{\omega}) = \varphi((M, \omega))$ (as described above), then

$$M \text{ halts, when executed on } \omega \iff \widehat{M} \text{ accepts } \widehat{\omega}.$$

The alphabet Σ_{TM} and ***encoding scheme*** from instances of the Acceptance Problem to strings in $\Sigma_{\text{TM}}^\star$, from the lecture material for "Universal Turing Machines and First Hard Problems", can now be applied. Three languages can now be defined from the Acceptance Problem:

- The ***language of instances*** of the Acceptance Problem is the language, $L_{\text{TM+I}} \subseteq \Sigma_{\text{TM}}^\star$, that was proved to ***decidable*** in in the lecture material for "Universal Turing Machines and First Hard Problems".
- The ***language of Yes-instances*** of the Acceptance Problem is the language, $A_{\text{TM}} \subseteq L_{\text{TM+I}}$, that was proved to be ***recognizable***, but also ***undecidable***, in the lecture material for "Universal Turing Machines and First Hard Problems".

- The ***language of No-instances*** of the Acceptance Problem is the language, $NA_{\mathsf{TM}} \subseteq L_{\mathsf{TM+I}}$, that was proved to be ***undecidable*** in the lecture material for "Oracle Reductions".

The same encoding scheme can be applied to instances of the Halting Problem.

- The ***language of instances*** of the Halting Problem is the language $L_{\mathsf{TM+I}} \subseteq \Sigma_{\mathsf{TM}}^{\star}$ — the same of the language of instances of the Acceptance Problem. As noted above, this was shown to be ***decidable*** in the lecture material for "Universal Turing Machines and First Hard Problems".
- The ***language of Yes-instances*** of the Halting problem is the language $\textit{Halt}_{\mathsf{TM}} \subseteq L_{\mathsf{TM+I}}$ of encodings of Turing machines M, and input strings ω for M, such that M halts when executed on input ω.
- The ***language of No-instances*** of the Halting problem is the language $\textit{Loop}_{\mathsf{TM}} \subseteq L_{\mathsf{TM+I}}$ of encodings of Turing machines M, and input strings ω for M, such that M *does not* halt when executed on input ω, that is, such that M loops on ω.

As noted in the lecture presentation, when we say that we wish to show that "the Halting Problem is reducible to the Acceptance Problem", we mean that we wish to show that

$$\textit{Halt}_{\mathsf{TM}} \preceq_{\mathsf{M}} A_{\mathsf{TM}}.$$

When proving this, used the fact that the language, $L_{\mathsf{TM+I}}$, of instances of the Halting Problem, is decidable. We also used a string $\mu_{\mathsf{Junk}} \in \Sigma_{\mathsf{TM}}^{\star}$ that *does not* encode an instance of the Acceptance Problem, that is, such that $\mu_{\mathsf{Junk}} \notin L_{\mathsf{TM+I}}$. *For this example*, the string μ_{Junk} was chosen to be the empty string, λ.