Computer Science 351

Proofs of Undecidability — Examples II

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #17

Goal for Today

- Another proof that a language is undecidable, using a many-one reduction, will be presented.
- ***Note:*** This example is more complicated than anything that you will be asked to supply on an assignment or test in this course.

Decidable Languages

Recall that the following languages have been proved to be *decidable*:

- TM $\subseteq \Sigma^{\star}_{\text{TM}}$: Valid encodings of Turing machines
- TM+I $\subseteq \Sigma^{\star}_{\text{TM}}$: Valid encodings of Turing machines *M* and strings of symbols over the input alphabet for *M*.

Undecidable Languages

The following languages are *undecidable:*

- $A_{TM} \subseteq \text{TM+I} \subseteq \Sigma_{TM}^{*}$: Encodings of Turing machines M and strings ω of symbols over the input alphabet for M such that M accepts ω (see Lecture #13).
- $\text{HALT}_{TM} \subseteq \text{TM+I} \subseteq \Sigma_{TM}^{\star}$: Encodings of Turing machines M and strings ω of symbols over the input alphabet for M such that M halts when executed on input ω (see Lecture #15).
- $\text{All}_{TM} \subseteq \text{TM} \subseteq \Sigma_{TM}^{\star}$: Encodings of Turing machines that accept all possible input strings — that is, Turing machines M with an input alphabet Σ such that $L(M) = \Sigma^{\star}$ (see Lecture #16).

The Language $\text{Regular}_{\text{TM}}$

Let

$$\text{Regular}_{\text{TM}} \subseteq \text{TM} \subseteq \Sigma_{\text{TM}}^{\star}$$

be the set of encodings of Turing machines M such that $L(M)$ is a regular language.

- We will prove that $\text{Regular}_{\text{TM}}$ is ***undecidable*** by showing that $A_{\text{TM}} \preceq_{\text{M}} \text{Regular}_{\text{TM}}$.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

What Do We Need to Do?

We must describe a total function $f : \Sigma_{\text{TM}}^{\star} \to \Sigma_{\text{TM}}^{\star}$ which satisfies the following properties:

- For every string $\mu \in \Sigma_{\text{TM}}^{\star}$,

  $$\mu \in \text{A}_{\text{TM}} \textit{ if and only if } f(\mu) \in \text{Regular}_{\text{TM}}.$$

- The function f is computable.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

Handling a Pesky Case

- Not all strings in $\Sigma_{\text{TM}}^{\star}$ encode Turing machines and input strings for them — only strings in the ***decidable*** language TM+I do.
- If $\mu \in \Sigma_{\text{TM}}^{\star}$ and $\mu \notin$ TM+I then $\mu \notin \text{A}_{\text{TM}}$, since $\text{A}_{\text{TM}} \subseteq$ TM+I. We want to define $f(\mu)$ so that $f(\mu) \notin \text{Regular}_{\text{TM}}$ in this case.
- Recall that $\text{Regular}_{\text{TM}} \subseteq$ TM, where TM is the language of encodings of Turing machines. If x_{No} is any string in $\Sigma_{\text{TM}}^{\star}$ such that $x_{\text{No}} \notin$ TM then $x_{\text{No}} \notin \text{Regular}_{\text{TM}}$ — so that setting $f(\mu)$ to be x_{No} ensures that $f(\mu) \notin \text{Regular}_{\text{TM}}$, as is needed here.
- Since $\lambda \notin$ TM we can choose x_{No} to be λ for this problem.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

We are left with the problem of defining $f(\mu)$ when $\mu \in$ TM+I.

- In this case μ is the encoding of some Turing machine

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

  and some input string $\omega \in \Sigma^\star$ for the encoded Turing machine M.

- Let $m = |\Gamma| - 1$, so that (using the notation from Lecture #12) m is a non-negative integer such that

$$\Gamma = \{\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_m\}.$$

- Let $\widehat{m} = \max(m, 4)$ and let

$$\widehat{\Gamma} = \{\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_{\widehat{m}}\}$$

  — so that $\widehat{\Gamma} = \Gamma$ if $m \geq 4$ and $\Gamma \subset \widehat{\Gamma}$ if $m \leq 3$.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

- Let

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

be the Turing machine with the same set Q of states as M, the same input alphabet Σ as M, tape alphabet Γ as given above, and where $\widehat{\delta} : Q \times \Gamma \to Q \times \Gamma \times \{\mathtt{L}, \mathtt{R}\}$ is a partial function such that, for every state $q \in Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}$ and for every integer i such that $0 \leq i \leq \widehat{m}$,

$$\widehat{\delta}(q, \sigma_i) = \begin{cases} \delta(q, \sigma_i) & \text{if } 0 \leq i \leq m, \\ (q_{\text{reject}}, \sigma_i, \mathtt{R}) & \text{if } m + 1 \leq i \leq \widehat{m}. \end{cases}$$

Then $M' = M$ whenever $m \geq 4$.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

Note that, if $\omega \in \Sigma^\star$, then M' follows the same sequence of configurations when executed on input ω as M does. This can be used to complete the following.

Exercise:

1. Prove that M' accepts ω if and only if M accepts ω, for every string $\omega \in \Sigma^\star$.
2. Describe a process that can be used to compute an encoding of M' from the encoding of M.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

- Now consider the Turing machine $\mathcal{M}_{\langle M',\omega\rangle}$ that is as described in the previous lecture, for M' as above and for a string $\omega \in \Sigma^{\star}$: As discussed in the previous lecture, the language of this Turing machine is

$$L\left(\mathcal{M}_{\langle M',\omega\rangle}\right) = \begin{cases} \Sigma^{\star} & \text{if } M' \text{ accepts } \omega, \\ \emptyset & \text{otherwise.} \end{cases}$$

- In particular, $\lambda \in L\left(\mathcal{M}_{\langle M',\omega\rangle}\right)$ if and only if M' accepts ω — and, as noted above, M' accepts ω if and only if M accepts ω.
- Information included in the notes for the previous lecture can be used to show that an encoding of the Turing machine $\mathcal{M}_{\langle M',\omega\rangle}$ can be computed from the input string μ (which encodes M and ω).

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

Finally, let

$$\Sigma_2 = \{\mathtt{a}, \mathtt{b}\}$$

(so that $\sigma_1 = \mathtt{a}$ and $\sigma_2 = \mathtt{b}$, using the encoding for Turing machines described in Lecture #12) and let

$$M_{\text{Nonregular}} = (\widehat{Q}, \Sigma_2, \widehat{\Gamma}, \widetilde{\delta}, q_0, q_{\text{accept}}, q_{\text{reject}})$$

be a Turing machine that decides the *non-regular* language

$$L_{\text{Nonregular}} = \{\mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{Z} \text{ and } n \geq 0\} \subseteq \Sigma_2^{\star}$$

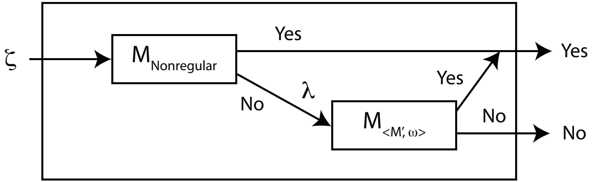
and which satisfies the following additional properties.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

- $|\widehat{Q}| = 10$.
- For every string $\zeta \in \Sigma_2^\star$ such that $\zeta \notin L_{\text{Nonregular}}$, the execution of $M_{\text{Nonregular}}$ on input ζ ends with the tape filled with copies of $\sqcup$, with the tape head resting at the leftmost cell of the tape.
- A string in $\Sigma_{\text{TM}}^\star$, which encodes $M_{\text{Nonregular}}$, can be computed from the unpadded decimal of the integer m that is described above.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

Suppose — finally — that $f(\mu)$ is an encoding of the following Turing machine, $\widetilde{M}_{\langle M', \omega \rangle}$:



This Turing machine has input alphabet Σ_2 and tape alphabet $\widehat{\Gamma}$, and it implements the algorithm on the following slide.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

On input $\zeta \in \Sigma_2^\star$ {

1. `if` ($\zeta \in L_{\text{Nonregular}}$) {
2. `accept` ζ

   } `else` {

3. Execute the Turing machine $\mathcal{M}_{\langle M',\omega\rangle}$ with the empty string, λ, as input. If this execution ends then `accept` if $\mathcal{M}_{\langle M',\omega\rangle}$ accepts λ, and `reject` if $\mathcal{M}_{\langle M',\omega\rangle}$ rejects λ.

   }

}

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

Claim #1: Let $\mu \in \Sigma_{\text{TM}}^{\star}$. If $\mu \in \text{A}_{\text{TM}}$ then $f(\mu) \in \text{Regular}_{\text{TM}}$.

Proof: Let $\mu \in \Sigma_{\text{TM}}^{\star}$ such that $\mu \in \text{A}_{\text{TM}}$. Then μ encodes a Turing machine

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

and an input string $\omega \in \Sigma^{\star}$ such that M accepts ω.

- As noted above, the corresponding Turing machine M' also has input alphabet Σ and accepts the string ω.
- The corresponding Turing machine $\mathcal{M}_{\langle M', \omega \rangle}$ (obtained using the construction given in the previous lecture) is then a Turing machine with input alphabet Σ such that $L\left(\mathcal{M}_{\langle M', \omega \rangle}\right) = \Sigma^{\star}$ — so that, in particular, this Turing machine accepts the empty string λ.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

Now consider an execution of the Turing machine $\widetilde{M}_{\langle M',\omega\rangle}$, given above, on a string $\zeta \in \Sigma_2^\star$. Either $\zeta \in L_{\text{Nonregular}}$ or $\zeta \notin L_{\text{Nonregular}}$.

- If $\zeta \in L_{\text{Nonregular}}$ then $\widetilde{M}_{\langle M',\omega\rangle}$ accepts ζ because $M_{\text{Nonregular}}$ accepts ζ (so that the test at line 1 of the above algorithm would pass) — and then the Turing machine $\widetilde{M}_{\langle M',\omega\rangle}$ would immediately accept ζ as well.
- If $\zeta \notin L_{\text{Nonregular}}$ then $\widetilde{M}_{\langle M',\omega\rangle}$ accepts ζ for a different reason: Now $M_{\text{Nonregular}}$ rejects ζ (and the test at line 1 in the algorithm fails), so that $\mathcal{M}_{\langle M',\omega\rangle}$ is executed on the empty string, λ. As noted above, $\mathcal{M}_{\langle M',\omega\rangle}$ accepts λ — and $\widetilde{M}_{\langle M',\omega\rangle}$ accepts ζ at this point.

Thus the language of $\widetilde{M}_{\langle M',\omega\rangle}$ is $\Sigma_2^\star$ — which is certainly a regular language. Since $f(\mu)$ is the encoding of the Turing machine $\widetilde{M}_{\langle M',\omega\rangle}$ it follows that $f(\mu) \in \text{Regular}_{\text{TM}}$, as claimed. □

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

Claim #2: Let $\mu \in \Sigma^{\star}_{\text{TM}}$. If $\mu \notin \text{A}_{\text{TM}}$ then $f(\mu) \notin \text{Regular}_{\text{TM}}$.

Proof: Let $\mu \in \Sigma^{\star}_{\text{TM}}$ such that $\mu \notin \text{A}_{\text{TM}}$. Then either $\mu \notin$ TM+I, or $\mu \in$ TM+I but $\mu \notin \text{A}_{\text{TM}}$. These cases are considered separately below.

Case: $\mu \notin$ TM+I. In this case $f(\mu) = \lambda$, the empty string. Since $\text{Regular}_{\text{TM}} \subseteq$ TM and $\lambda \notin$ TM, $\lambda \notin \text{Regular}_{\text{TM}}$. That is, $f(\mu) \notin \text{Regular}_{\text{TM}}$ in this case, as claimed.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

Case: $\mu \in$ TM+I but $\mu \notin \text{A}_{\text{TM}}$. In this case μ encodes a Turing machine

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

and an input string $\omega \in \Sigma^\star$ such that M either rejects or loops on ω.

- It follows that the corresponding Turing machine M' either rejects or loops on ω as well.
- The corresponding Turing machine $\mathcal{M}_{\langle M', \omega \rangle}$ (obtained using the construction given in the previous lecture) is then a Turing machine with input alphabet Σ such that $L\left(\mathcal{M}_{\langle M', \omega \rangle}\right) = \emptyset$ — so that, in particular, this Turing machine either rejects or loops on the empty string λ.

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

Now consider an execution of the Turing machine $\widetilde{M}_{\langle M',\omega\rangle}$, given above, on a string $\zeta \in \Sigma_2^\star$. Either $\zeta \in L_{\text{Nonregular}}$ or $\zeta \notin L_{\text{Nonregular}}$.

- If $\zeta \in L_{\text{Nonregular}}$ then $\widetilde{M}_{\langle M',\omega\rangle}$ accepts ζ because $M_{\text{Nonregular}}$ accepts ζ (so that the test at line 1 of the above algorithm would pass) — and then the Turing machine $\widetilde{M}_{\langle M',\omega\rangle}$ would immediately accept ζ as well.
- If $\zeta \notin L_{\text{Nonregular}}$ then $M_{\text{Nonregular}}$ rejects ζ (and the test at line 1 in the algorithm fails), so that $\mathcal{M}_{\langle M',\omega\rangle}$ is executed on the empty string, λ. As noted above, $\mathcal{M}_{\langle M',\omega\rangle}$ either rejects or loops on λ — and $\widetilde{M}_{\langle M',\omega\rangle}$ either rejects or loops on ζ.

Thus the language of $\widetilde{M}_{\langle M',\omega\rangle}$ is $L_{\text{Nonregular}}$ — which is *not* a regular language. Since $f(\mu)$ is the encoding of the Turing machine $\widetilde{M}_{\langle M',\omega\rangle}$ it follows that $f(\mu) \notin \text{Regular}_{\text{TM}}$ in this case as well, as is needed to complete the proof of this claim. $\square$

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

Claim #3: The function $f : \Sigma_{\text{TM}}^{\star} \to \Sigma_{\text{TM}}^{\star}$ is a computable function.

The proof of this claim is given in a supplemental document for this lecture. (It is somewhat too long to serve as a good example of this kind of proof.)

A Reduction from A_{TM} to $\text{Regular}_{\text{TM}}$

- Since f is a well-defined total function from $\Sigma_{\text{TM}}^{\star}$ to $\Sigma_{\text{TM}}^{\star}$, Claims #1, #2 and #3 imply that f is a ***many-one reduction*** from A_{TM} to $\text{Regular}_{\text{TM}}$.
- Thus $\text{A}_{\text{TM}} \preceq_{\text{M}} \text{Regular}_{\text{TM}}$.
- Since A_{TM} is undecidable, it now follows that $\text{Regular}_{\text{TM}}$ is undecidable, as well.

A Many-One Reduction

The function *f* has now been shown to have all the properties of a "many-one reduction" from A_{TM} to $\text{Regular}_{\text{TM}}$, so that

$$\text{A}_{\text{TM}} \preceq_{\text{Regular}_{\text{TM}}} .$$

Since A_{TM} is undecidable it now follows that $\text{Regular}_{\text{TM}}$ is undecidable as well.

Rice's Theorem

Rice's Theorem: Suppose P is a property of Turing machines that satisfies the following conditions:

- This property is "nontrivial:" There exists at least one Turing machine M_{Yes} that satisfies this property, and at least one Turing machine "M_{No}" that *does not* satisfy this property.
- This is actually a property of the ***languages*** of these machines: That is, if M_1 and M_2 are Turing machines such that $L(M_1) = L(M_2)$ then M_1 satisfies this property if and only if M_2 does.

Then the language $L_P \subseteq \text{TM}$ including encodings of Turing machines satisfying property P is undecidable.

Rice's Theorem

- A proof of Rice's Theorem will be included in a supplemental document for this lecture.
- Rice's Theorem can be used to identity many more undecidable languages.
- It can be proved using a modification of the argument that was used to show that the language Regular_{TM} is undecidable.
- You will *not* be allowed to use Rice's Theorem to prove that a language is undecidable, on an assignment or test in this course, unless the instructions clearly state that that you can.