Computer Science 351

First Undecidable and Unrecognizable Languages

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #13

Goal for Today

- Identification of a language that is ***undecidable***, as well as a language that is ***unrecognizable***

Two Decidable Languages

Once again, let

$$\Sigma_{\mathsf{TM}} = \{\mathtt{(}, \mathtt{)}, \mathtt{,}, \mathtt{q}, \mathtt{s}, \mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}, \mathtt{4}, \mathtt{5}, \mathtt{6}, \mathtt{7}, \mathtt{8}, \mathtt{9}, \mathtt{Y}, \mathtt{N}, \mathtt{L}, \mathtt{R}, \mathtt{\#}\}.$$

- This is the input alphabet for the ***universal Turing machine*** that was described in Lecture #12.

Two Decidable Languages

- Let TM $\subseteq \Sigma_{TM}^{\star}$ be the language of encodings of Turing machines

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

as described in Lecture #12.

- Let TM+I $\subseteq \Sigma_{TM}^{\star}$ be the language of encodings of Turing machines M, as above, and of input strings $\omega \in \Sigma^{\star}$ for M.
- Lecture #12 includes information that can be used to describe algorithms to decide membership of strings $\mu \in \Sigma_{TM}^{\star}$ in each of TM and TM+I — so that both of these languages are ***decidable***.[1]

[1] The alphabet Σ_{TM}, and the encoding scheme for Turing machines and their input strings were chosen to make it reasonably easy to confirm this.

A_{TM} is Undecidable

Let $A_{TM} \subseteq \Sigma_{TM}^{\star}$ be the language of encodings of Turing machines $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ and input strings $\omega \in \Sigma^{\star}$ such that M ***accepts*** ω.

- This is the language of the universal Turing machine, M_{UTM}, that was described in Lecture #12.
- It follows, from this, that A_{TM} is ***recognizable***.

A_{TM} is Undecidable

Claim #1: A_{TM} is ***undecidable***.

Proof: By contradiction.

- ***Assume*** that A_{TM} is decidable. Then there exists a Turing machine, M_{ATM}, that decides A_{TM}.
- Consider the algorithm on the following side.

A_{TM} is Undecidable

On input $\mu \in \Sigma_{TM}^{\star}$:

1. `if` ($\mu \in$ TM) {
   Let M_μ be the Turing machine encoded by μ.
2. `if` (the input alphabet for M_μ is Σ_{TM}) {
3. `if` (M_μ accepts μ) {
4. `reject` μ
   } `else` {
5. `accept`
   }
6. } `else` { `reject` }
7. } `else` { `reject` }

}

A_{TM} is Undecidable

- The test at line 1 can be carried out because the language TM is decidable.
- If the test at line 1 is passed then μ encodes some Turing machine

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

  The test at line 2 simply asks whether $|\Sigma| = |\Sigma_{\text{TM}}| = 20$ — and this is easily checked using the encoding, μ for M.

A_{TM} is Undecidable

- Suppose that the test at line 2. For a string $\omega \in \Sigma_{TM}^{\star}$, let $e(\omega)$ be the *longer* string in Σ used to "encode"' ω as an input string for the Turing machine, M_{μ}. As described in Lecture #12, this depends on the size of M_{μ}'s tape alphabet, Γ — but $e(\omega)$ can certainly be computed if both ω and the input, μ, of the Turing machine M_{μ} are available.
- The test at line 3 is passed if and only if the string

$$(\mu, e(\mu))$$

belongs to A_{TM}. Since this string can certainly be computed using the input string, μ, it follows by the ***assumption***, that A_{TM} is decidable, that this test can also be carried out.

A_{TM} is Undecidable

- Since the remaining steps simply either accept or reject the input, it follows that there is a Turing machine, M_D, which implements this algorithm, and which ***decides*** a language $L_D \subseteq \Sigma_{TM}^\star$.
- Let $\mu \in \Sigma_{TM}^\star$ be a string that encodes this Turing machine, M_D — so that "M_μ" is the Turing machine M_D.
- Either $\mu \in L_D$, or $\mu \notin L_D$.

A_{TM} is Undecidable

$$
\begin{aligned}
\mu \in L_D &\implies M_D \text{ accepts } \mu && \text{(since } M_D \text{ decides } L_D\text{)} \\
&\implies \text{The step at line 5 is reached when the algorithm implemented by } M_D \text{ is executed on input } \mu \\
&\implies \text{The test at line 3 has failed} \\
&\implies M_\mu \text{ does not accept } \mu \\
&\implies M_D \textit{ does not } \text{accept } \mu && \text{(since } M_\mu = M_D\text{)} \\
&\implies \mu \notin L_D && \text{(since } M_D \text{ decides } L_D\text{).}
\end{aligned}
$$

Since a claim cannot be true if it implies its own negation, it follows that $\mu \notin L_D$.

A_{TM} is Undecidable

On the other hand, since $\mu \in$ TM and μ encodes a Turing machine with input alphabet Σ_{TM}, The tests at lines 1 and 2 are passed when this algorithm is executed on input μ, so that μ can only be rejected by reaching and executing the step at line 4. Thus

$$\begin{aligned}
\mu \notin L_D &\implies M_D \text{ rejects } \mu && \text{(since } M_D \text{ decides } L_D\text{)} \\
&\implies \text{The step at line 4 is reached when the algorithm implemented by } M_D \text{ is executed on input } \mu \\
&\implies \text{The test at line 3 has passed} \\
&\implies M_\mu \text{ accepts } \mu \\
&\implies M_D \text{ accepts } \mu && \text{(since } M_\mu = M_D\text{)} \\
&\implies \mu \in L_D && \text{(since } M_D \text{ decides } L_D\text{).}
\end{aligned}$$

Once again, a claim that implies its own negation cannot be true. It now follows that $\mu \in L_D$.

A_{TM} is Undecidable

- Since a ***contradiction*** has now been obtained (because it cannot be true both that $\mu \notin L_D$ and that $\mu \in L_D$) the only ***assumption***, that was made, must be incorrect.
- Thus A_{TM} is undecidable, as claimed. □

The Complement of A_{TM} is Unrecognizable

Claim #2: Let $L \subseteq \Sigma^\star$ (for some alphabet Σ). If both L and its complement,

$$L^C = \Sigma^\star \setminus L = \{\omega \in \Sigma^\star \mid \omega \notin L\}$$

are ***recognizable***, then L is ***decidable***.

Proof: Suppose that both L and L^C are recognizable.

- Then there exists a Turing machine M_Y, with input alphabet Σ, whose language is L, as well as a Turing machine M_N, with input alphabet Σ, whose language is L^C.
- We may assume $L \neq \emptyset$ and $L \neq \Sigma^\star$ became these languages are both decidable (and it is sufficient to consider other subsets of $\Sigma^\star$ in order to prove the claim).

The Complement of A_{TM} is Unrecognizable

How Not to Prove This:

- We cannot just run one of these machines on the input string, and then run the other machine after that — because each of these machines might loop on the given input string!

What To Do, Instead:

- We will run both computations by parallel — by interleaving, or ***dovetailing*** them.
- A two-tape Turing machine that uses this approach to ***decide*** the language L will be described.
- It will follow, by a result already established about multi-tape Turing machines, that there is also a standard Turing machine that decides L — that is, L is ***decidable***, as claimed.

The Complement of A_{TM} is Unrecognizable

Starting the Computation:

On input $\omega \in \Sigma^\star$ {

1. Write a copy of ω on the second tape, restoring the copy of ω on the first tape afterwards (so that both store a copy of ω) with both tape heads at the left end of their tapes.
2. Use the finite control to remember that both M_Y and M_N are in their start states.

Now, the *first* tape can be used to simulate the execution of M_Y on input ω while the second tape can be used to simulate the execution of M_N on input ω. The finite control will be used to remember which state each machine would be in, at each point during this simulation.

The Complement of A_{TM} is Unrecognizable

Continuing the Computation:

3. `while` (`true`) {
4.   Use Tape #1 and the finite control to carry out the next step in the execution of M_Y on input ω.
5.   `if` (M_Y accepted, at this point) {
6.   `accept`
7.   } `else if` (M_Y rejected at this point) {
8.   `reject`

    }

The loop, started at line 3, continues on the next slide...

The Complement of A_{TM} is Unrecognizable

Continuing the Computation...

9. Use Tape #2 and the finite control to carry out the next step in the execution of M_N on input ω.
10. `if` (M_N accepted at this point) {
11. `reject`
12. } `else if` (M_N rejected at this point) {
13. `accept`

}

} // *End of Loop*

}

The execution ends if and only if one of the steps at lines 6, 8, 11 or 13 is reached.

The Complement of A_{TM} is Unrecognizable

Consider an execution of a string $\omega \in \Sigma^\star$ such that $\omega \in L$.

- Since $L(M_Y) = L$, M_Y ***accepts*** ω after k steps for positive integer k.
- It is possible that M_N ***rejects*** ω after ℓ steps for some integer ℓ such that $1 \leq \ell \leq k - 1$. In this case the step at line 13 is reached, and ω is ***accepted***, during the ℓ^{th} execution of the body of the loop.
- Otherwise, the step at line 6 is reached and ω is ***accepted*** during the k^{th} execution of the body of the loop.

Thus every string $\omega \in L$ is ***accepted*** by this two-tape Turing machine.

The Complement of A_{TM} is Unrecognizable

Consider an execution of a string $\omega \in \Sigma^\star$ such that $\omega \notin L$.

- Since $L(M_N) = L^C$, M_N ***accepts*** ω after k steps for positive integer k.
- It is possible that M_Y ***rejects*** ω after ℓ steps for some integer ℓ such that $1 \leq \ell \leq k$. In this case the step at line 8 is reached, and ω is ***rejected***, during the ℓ^{th} execution of the body of the loop.
- Otherwise, the step at line 11 is reached and ω is ***rejected*** during the k^{th} execution of the body of the loop.

Thus every string $\omega \in \Sigma^\star$ such that $\omega \notin L$ is ***rejected*** by this two-tape Turing machine.

The Complement of A_{TM} is Unrecognizable

It follows from the above that this two-tape Turing machine ***decides*** the language L.

- As noted above it follows, by a result already established for multi-tape Turing machines, that L is ***decidable***, as claimed. □

The Complement of A_{TM} is Unrecognizable

Corollary #3 The language A_{TM}^C (that is, the complement of the language A_{TM}) is unrecognizable.

Proof: By contradiction.

- ***Assume*** that A_{TM}^C is recognizable.
- As noted above, it was proved in Lecture #12 that A_{TM} is recognizable.
- It now follows by Claim #2 (with $L = A_{TM}$) that A_{TM} is decidable.
- However, this ***contradicts*** Claim #1, which established that A_{TM} is undecidable.
- Our ***assumption*** must, therefore, be false: A_{TM}^C is unrecognizable, as claimed. □

Finishing Up

- Claim #1 was proved using a ***diagonalization*** argument. These have been used in mathematics to prove a variety of significant results — including the fact that the set $\mathbb{R}$ of real numbers is "uncountable".
- The technique used to prove Claim #2 — sometimes called ***dovetailing*** — is also useful for proving at least a few more interesting properties of the set of recognizable languages.
- However, we ***will not*** be using these techniques to prove that other languages are undecidable or unrecognizable. Techniques that *will* be used this will be introduced in Lecture #14.