1 Rice’s Theorem

1.1 Properties

Checking Properties

Given $M$

\[
\begin{align*}
\text{Does } & L(M) \text{ contain } \langle M \rangle? \\
\text{Is } & L(M) \text{ non-empty?} \\
\text{Is } & L(M) \text{ empty?} \\
\text{Is } & L(M) \text{ infinite?} \\
\text{Is } & L(M) \text{ finite?} \\
\text{Is } & L(M) \text{ co-finite (i.e., is } L(M) \text{ finite?)} \\
\text{Is } & L(M) = \Sigma^*?
\end{align*}
\]

\[
\begin{align*}
\text{Undecidable} \\
\text{Undecidable}
\end{align*}
\]

None of these properties can be decided. This is the content of Rice’s Theorem.

Properties

Definition 1. A property of languages is simply a set of languages. We say $L$ satisfies the property $\mathcal{P}$ if $L \in \mathcal{P}$.

Definition 2. For any property $\mathcal{P}$, define language $L_\mathcal{P}$ to consist of Turing Machines which accept a language in $\mathcal{P}$:

\[ L_\mathcal{P} = \{ \langle M \rangle \mid L(M) \in \mathcal{P} \} \]

Deciding $L_\mathcal{P}$: deciding if a language represented as a TM satisfies the property $\mathcal{P}$.

- Example: \{\langle M \rangle \mid L(M) \text{ is infinite} \}; $E_{TM} = \{ \langle M \rangle \mid L(M) = \emptyset \}$
- Non-example: \{\langle M \rangle \mid M \text{ has 15 states} \} ← This is a property of TMs, and not languages!

Trivial Properties

Definition 3. A property is trivial if either it is not satisfied by any r.e. language, or if it is satisfied by all r.e. languages. Otherwise it is non-trivial.

Example 4. Some trivial properties:

- $\mathcal{P}_{\text{ALL}} = \text{set of all languages}$
- $\mathcal{P}_{\text{r.e.}} = \text{set of all r.e. languages}$
- $\mathcal{P}$ where $\mathcal{P}$ is trivial
- $\mathcal{P} = \{ L \mid L \text{ is recognized by a TM with an even number of states} \} = \mathcal{P}_{\text{r.e.}}$

Observation. For any trivial property $\mathcal{P}$, $L_\mathcal{P}$ is decidable. (Why?) Then $L_\mathcal{P} = \Sigma^*$ or $L_\mathcal{P} = \emptyset$. _
1.2 Main Theorem

Rice’s Theorem

Proposition 5. If $\mathcal{P}$ is a non-trivial property, then $L_\mathcal{P}$ is undecidable.

- Thus $\{ \langle M \rangle \mid L(M) \in \mathcal{P} \}$ is not decidable (unless $\mathcal{P}$ is trivial)

We cannot algorithmically determine any interesting property of languages represented as Turing Machines!

Properties of TMs

Note. Properties of TMs, as opposed to those of languages they accept, may or may not be decidable.

Example 6.

\[
\begin{align*}
\{ \langle M \rangle \mid M \text{ has 193 states} \} & \quad \text{Decidable} \\
\{ \langle M \rangle \mid M \text{ uses at most 32 tape cells on blank input} \} & \quad \text{Decidable} \\
\{ \langle M \rangle \mid M \text{ halts on blank input} \} & \\
\{ \langle M \rangle \mid M \text{ halts on blank input} \} & \quad \text{Undecidable} \\
\{ \langle M \rangle \mid \text{on input 0011 } M \text{ at some point writes the symbol } $\text{ on its tape} \} &
\end{align*}
\]

Proof of Rice’s Theorem

Rice’s Theorem

If $\mathcal{P}$ is a non-trivial property, then $L_\mathcal{P}$ is undecidable.

Proof. Suppose $\mathcal{P}$ non-trivial and $\emptyset \not\in \mathcal{P}$. If $\emptyset \in \mathcal{P}$, then in the following we will be showing $L_\mathcal{P}$ is undecidable. Then $L_\mathcal{P} = \overline{L_\mathcal{P}}$ is also undecidable.

Recall $L_\mathcal{P} = \{ \langle M \rangle \mid L(M) \text{ satisfies } \mathcal{P} \}$. We’ll reduce $A_{TM}$ to $L_\mathcal{P}$. Then, since $A_{TM}$ is undecidable, $L_\mathcal{P}$ is also undecidable. Broadly the idea behind the reduction is as follows. Since $\mathcal{P}$ is non-trivial, at least one r.e. language satisfies $\mathcal{P}$, i.e., $L(M_0) \in \mathcal{P}$ for some TM $M_0$. We will show a reduction $f$ that maps an instance $\langle M, w \rangle$ for $A_{TM}$, to $N$ such that

- If $M$ accepts $w$ then $N$ accepts the same language as $M_0$. Then $L(M) = L(M_0) \in \mathcal{P}$
- If $M$ does not accept $w$ then $N$ accepts $\emptyset$. Then $L(N) = \emptyset \not\in \mathcal{P}$

Thus, $\langle M, w \rangle \in A_{TM}$ iff $N \in L_\mathcal{P}$.

We now describe the reduction precisely. The reduction $f$ maps $\langle M, w \rangle$ to $\langle N \rangle$, where $N$ is a TM that behaves as follows:

On input $x$

- Ignore the input and run $M$ on $w$
- If $M$ does not accept (or doesn’t halt) then do not accept $x$ (or do not halt)
- If $M$ does accept $w$ then run $M_0$ on $x$ and accept $x$ iff $M_0$ does.
Notice that indeed if $M$ accepts $w$ then $L(N) = L(M_0)$. Otherwise $L(N) = \emptyset$.

---

**Rice’s Theorem**

*Recap*

Every non-trivial property of r.e. languages is undecidable

- Rice’s theorem says nothing about properties of Turing machines
- Rice’s theorem says nothing about whether a property of languages is recursively enumerable or not.

---

**Big Picture . . . again**

```
Languages
  Recursively Enumerable
    Decidable
      CFL
      Regular

“almost all” properties!

$\downarrow$

$L_d, A_{TM}, E_{TM}$

$A_{TM}, E_{TM}, HALT$

$L_{anbncn}$

$L_{0n1n}$


```