RICE'S

THEOREMS

Abhijit Das

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

March 14, 2021

Properties of RE Languages

- Class of all r.e. languages: $\text{RE} = \left\{ \mathscr{L}(M) \;\middle|\; M \text{ is a Turing machine} \right\}$.
- Each member of RE is specified by a Turing machine.
- Unrestricted grammars can also be used to specify r.e. languages.

- A property of r.e. sets is a map $P : \text{RE} \to \{T, F\}$.
- Example: Emptiness is a property defined as $P_{EMP}(L) = \begin{cases} T & \text{if } L = \emptyset, \\ F & \text{if } L \neq \emptyset. \end{cases}$
- Properties too are specified by Turing machines.
- Example: The emptiness property is specified by *any member* of

$$P_{EMP} = \left\{ M \;\middle|\; \mathscr{L}(M) = \emptyset \right\}.$$

Examples of Properties

- Finiteness property: Any member of $\left\{M \mid \mathscr{L}(M) \text{ is finite}\right\}$.
- Regularity property: Any member of $\left\{M \mid \mathscr{L}(M) \text{ is regular}\right\}$.
- Context-free property: Any member of $\left\{M \mid \mathscr{L}(M) \text{ is context free}\right\}$.
- Acceptance of a string: Any member of $\left\{M \mid 01011000 \in \mathscr{L}(M)\right\}$.
- Full-ness property: Any member of $\left\{M \mid \mathscr{L}(M) = \Sigma^*\right\}$.

- We specify a property by a *single Turing machine*, the language of which has that property.
- Properties are properties of *r.e. sets*, *not* of Turing machines.
- A property must be independent of the representative machine.

Non-Examples

- Any member of $\left\{M \mid M \text{ has at least 2021 states}\right\}$.
    - We can design two TMs M_1 and M_2 both accepting $\emptyset$.
    - M_1 has less than 2021 states.
    - M_2 has 2021 or more states.
    - If $\emptyset$ is represented by M_1, the property is false for $\emptyset$.
    - If $\emptyset$ is represented by M_2, the property is true for $\emptyset$.
- Any member of $\left\{M \mid M \text{ is a total TM}\right\}$.
- Any member of $\left\{M \mid M \text{ rejects } 01011000 \text{ and halts}\right\}$.
- Any member of $\left\{M \mid M \text{ ever goes to the right of the input}\right\}$.
- Any member of

  $$\left\{M \mid M \text{ has the least number of states among all machines accepting } \mathscr{L}(M)\right\}.$$

Types of Properties

- Trivial properties
  - The constant map RE $\to \{T, F\}$ taking all $L \in$ RE to T.
  - The constant map RE $\to \{T, F\}$ taking all $L \in$ RE to F.
- Any other property is called non-trivial.
- Example of trivial property: $\mathscr{L}(M)$ is recursively enumerable.
- Example of non-trivial property: $\mathscr{L}(M)$ is recursive.

- Monotone properties
  - Assume $F \leqslant T$.
  - Whenever $A \subseteq B$, we have $P(A) \leqslant P(B)$.
  - Examples of monotone properties: $\mathscr{L}(M)$ is infinite, $\mathscr{L}(M) = \Sigma^*$.
  - Examples of non-monotone properties: $\mathscr{L}(M)$ is finite, $\mathscr{L}(M) = \emptyset$.

Rice's Theorem (Part 1)

Theorem

Any non-trivial property P of r.e. languages is undecidable. In other words, the set

$$\Pi = \left\{ N \mid P(\mathscr{L}(N)) = T \right\}$$

is not recursive.

Proof

- Let P be a non-trivial property of r.e. languages.
- Suppose $P(\emptyset) = F$.
- Since P is non-trivial, there exist $L \in \mathrm{RE}$, $L \neq \emptyset$, such that $P(L) = T$.
- Let K be a Turing machine with $\mathscr{L}(K) = L$.
- We make a reduction from HP to Π.
- If $P(\emptyset) = T$, we take K with $\mathscr{L}(K) = L \neq \emptyset$ and $P(L) = F$. This establishes $\overline{\mathrm{HP}} \leqslant_m \Pi$.

Rice's Theorem: The Reduction HP $\leqslant_m \Pi$

- **Input:** $M \# w$ (an instance of HP)
- **Output:** A Turing machine N such that $P(\mathscr{L}(N)) = T$ if and only if M halts on w.
- Behavior of N on input v:
  - Copy v to a separate tape.
  - Write w to the first tape, and simulate M on w.
  - If the simulation halts:
    - Simulate K on v.
    - Accept if and only if K accepts v.

- If M halts on w, $\mathscr{L}(N) = \mathscr{L}(K) = L$.
- If M does not halt on w, $\mathscr{L}(N) = \emptyset$.
- $P(L) = T$ and $P(\emptyset) = F$.

Rice's Theorem: Part 2

Theorem

No non-monotone property P of r.e. languages is semidecidable. In other words, the set

$$\Pi = \left\{ N \mid P(\mathscr{L}(N)) = T \right\}$$

is not recursively enumerable.

Proof

- P is non-monotone. So there exist r.e. languages L_1 and L_2 such that

$$L_1 \subseteq L_2, \qquad P(L_1) = T, \qquad P(L_2) = F.$$

- Take Turing machines K_1, K_2 such that $\mathscr{L}(K_1) = L_1$ and $\mathscr{L}(K_2) = L_2$.
- We supply a reduction from $\overline{\mathrm{HP}}$ to Π.
- The reduction algorithm embeds the information of M, w, K_1, and K_2 in the finite control of N.

Rice's Theorem: Part 2: The Reduction $\overline{\mathbf{HP}} \leqslant_m \Pi$

- **Input:** $M \# w$.
- **Output:** A Turing machine N such that $P(\mathscr{L}(N)) = T$ if and only if M does *not* halt on w.
- Behavior of N on input v:
  - Copy v from the first tape to the second tape, and w from the finite control to the third tape.
  - Run three simulations in parallel (one step of each in round-robin fashion)

    K_1 on v on the first tape,
    K_2 on v on the second tape,
    M on w on the third tape.
  - Accept if and only if one of the following conditions hold:

    (1) K_1 accepts v,
    (2) M halts on w, and K_2 accepts v.

- M does not halt on $w \Rightarrow N$ accepts by (1) $\Rightarrow \mathscr{L}(N) = \mathscr{L}(K_1) = L_1$.
- If M halts on w, N accepts if either K_1 or K_2 accepts v. In this case, $\mathscr{L}(N) = \mathscr{L}(K_1) \cup \mathscr{L}(K_2) = L_1 \cup L_2 = L_2$ (since $L_1 \subseteq L_2$).

Tutorial Exercises

1. Prove/Disprove: No non-trivial property of r.e. languages is semidecidable.
2. Use Rice's theorems to prove that neither the following languages nor their complements are r.e.
   (a) $\mathrm{REG} = \{M \mid \mathscr{L}(M) \text{ is regular}\}$.
   (b) $\mathrm{CFL} = \{M \mid \mathscr{L}(M) \text{ is context-free}\}$.
   (c) $\mathrm{REC} = \{M \mid \mathscr{L}(M) \text{ is recursive}\}$.
3. [*Generalization of Rice's theorem for pairs of r.e. langauges*] Consider the set of pairs of r.e. languages: $\mathrm{RE}^2 = \{(L_1, L_2) \mid L_1, L_2 \in \mathrm{RE}\}$.
   (a) Define a property of pairs of r.e. languages.
   (b) How do you specify a property of a pair of r.e. languages?
   (c) Which properties of pairs of r.e. languages should be called non-trivial?
   (d) Prove that every non-trivial property of pairs of r.e. languages is undecidable.

Tutorial Exercises

4. Use the previous exercise to prove that the following problems about pairs of r.e. languages are undecidable.

   (a) $\mathscr{L}(M) = \mathscr{L}(N)$.
   (b) $\mathscr{L}(M) \subseteq \mathscr{L}(N)$.
   (c) $\mathscr{L}(M) \cap \mathscr{L}(N) = \emptyset$.
   (d) $\mathscr{L}(M) \cap \mathscr{L}(N)$ is finite.
   (e) $\mathscr{L}(M) \cap \mathscr{L}(N)$ is regular.
   (f) $\mathscr{L}(M) \cap \mathscr{L}(N)$ is context-free.
   (g) $\mathscr{L}(M) \cap \mathscr{L}(N)$ is recursive.
   (h) $\mathscr{L}(M) \cup \mathscr{L}(N) = \Sigma^*$.
   (i) $\mathscr{L}(M) \cup \mathscr{L}(N) = \emptyset$.
   (j) $\mathscr{L}(M) \cup \mathscr{L}(N)$ is finite.
   (k) $\mathscr{L}(M) \cup \mathscr{L}(N)$ is recursive.

5. Generalize Rice's theorem, Part 2, for pairs of RE sets.