CS21004 Formal Languages and Automata Theory, Spring 2019–2020

Rice's Theorems

---

1. Prove/Disprove: No non-trivial property of r.e. languages is semidecidable.

Solution *False.* Non-emptiness is a non-trivial property that is semidecidable (although not decidable). If a TM M accepts any strings at all, this can be detected by simulating M on all inputs on a time-sharing basis.

2. Use Rice's theorems to prove that neither the following languages nor their complements are r.e.

(a) $\mathrm{FIN} = \{M \mid \mathscr{L}(M) \text{ is finite}\}$.

Solution Finiteness is a non-monotone property. So by Rice's theorem, Part 2, FIN is not r.e. Infiniteness is however a monotone property, so Rice's theorem, Part 2, is not applicable to $\overline{\mathrm{FIN}}$. Infiniteness being a non-trivial property, Rice's theorem, Part 1, implies that $\overline{\mathrm{FIN}}$ is not recursive. It seems that nothing more about $\overline{\mathrm{FIN}}$ follows from Rice's theorems. We need a separate proof for the non-r.e.-ness of $\overline{\mathrm{FIN}}$ (already covered).

(b) $\mathrm{REG} = \{M \mid \mathscr{L}(M) \text{ is regular}\}$.

Solution Neither regularity nor non-regularity is a monotone property. For example, consider

$$\emptyset \subseteq \{0^p \mid p \text{ is a prime}\} \subseteq \Sigma^*.$$

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.

Solution A property of a pair of r.e. languages is a map $P : \mathrm{RE}^2 \to \{T, F\}$.

(b) How do you specify a property of a pair of r.e. languages?

Solution Such a property is specified by a pair of Turing machines M_1, M_2. We look at whether $P(\mathscr{L}(M_1), \mathscr{L}(M_2))$ is true (T) or false (F). A property must be independent of the representative TMs.

(c) Which properties of pairs of r.e. languages should be called non-trivial?

Solution The constant maps $P_T : \mathrm{RE}^2 \to \{T, F\}$ taking every (L_1, L_2) to T, and $P_F : \mathrm{RE}^2 \to \{T, F\}$ taking every (L_1, L_2) to F are trivial. Any other property is non-trivial.

(d) Prove that every non-trivial property of pairs of r.e. languages is undecidable.

Solution Let P be a non-trivial property of RE^2. We need to show that the language

$$\Pi = \{M_1 \# M_2 \mid P(\mathscr{L}(M_1), \mathscr{L}(M_2)) = T\}$$

is not recursive. Assume that $P(\emptyset, \emptyset) = F$. Since P is non-trivial, $P(L_1, L_2) = T$ for some $L_1, L_2 \in \mathrm{RE}$. Take two TMs K_1, K_2 with $\mathscr{L}(K_1) = L_1$ and $\mathscr{L}(K_2) = L_2$.

Now, use a reduction $\mathrm{HP} \leqslant_m \Pi$. The input to the reduction algorithm is $M \# w$, and the output is a pair of Turing machines M_1 and M_2 such that $P(\mathscr{L}(M_1), \mathscr{L}(M_2)) = T$ if and only if M halts on w. The reduction algorithm can embed the information of M, w, K_1, K_2 in the finite controls of M_1 and M_2.

Behavior of M_1 on input v_1:

1. Store v_1 on a second tape.
2. Copy w to the first track, and simulate M on w.
3. If the simulation halts, simulate K_1 on v_1 on the second tape, and accept v_1 if and only if K_1 accepts v_1.

Behavior of M_2 on input v_2:

1. Store v_2 on a second tape.
2. Copy w to the first track, and simulate M on w.
3. If the simulation halts, simulate K_2 on v_2 on the second tape, and accept v_2 if and only if K_2 accepts v_2.

If M halts on w, both M_1 and M_2 get a chance to simulate K_1 and K_2, respectively. In this case, $\mathscr{L}(M_1) = \mathscr{L}(K_1) = L_1$ and $\mathscr{L}(M_2) = \mathscr{L}(K_2) = L_2$. We have $P(L_1, L_2) = T$.

If M does not halt on w, neither M_1 nor M_2 gets a chance to run their respective simulations of K_1 and K_2. In this case, $\mathscr{L}(M_1) = \mathscr{L}(M_2) = \emptyset$. We have $P(\emptyset, \emptyset) = F$.

Remark: If $P(\emptyset, \emptyset) = T$, take $L_1, L_2 \in$ RE with $P(L_1, L_2) = F$. Use a reduction $\overline{\mathrm{HP}} \leqslant_m \Pi$.

4. Use the previous exercise to prove that the following languages are not recursive.

(a) $\{M \# N \mid \mathscr{L}(M) = \mathscr{L}(N)\}$.

Solution We have to show that the given set is neither empty nor equal to the entire set of pairs of TMs. Take M with $\mathscr{L}(M) = \emptyset$, N_1 with $\mathscr{L}(N_1) = \emptyset$, and N_2 with $\mathscr{L}(N_2) = \Sigma^*$. But then, $M \# N_1$ is in the given set, whereas $M \# N_2$ is not in the given set.

(b) $\{M \# N \mid \mathscr{L}(M) \cap \mathscr{L}(N)$ is recursive$\}$.

Solution Let U be the universal Turing machine. Its language $\mathrm{MP} = \mathscr{L}(U)$ is not recursive. Take $M = N_2 = U$, and N_1 with $\mathscr{L}(N_1) = \emptyset$. We see that $\mathscr{L}(M) \cap \mathscr{L}(N_1) = \emptyset$ is recursive, whereas $\mathscr{L}(M) \cap \mathscr{L}(N_2) = \mathrm{MP}$ is not recursive.

(c) $\{M \# N \mid \mathscr{L}(M) \cup \mathscr{L}(N)$ is recursive$\}$.

Solution Take $M = N_2 = U$, and N_1 with $\mathscr{L}(N_1) = \Sigma^*$. Proceed as in Part (b).