Computer Science C63 — Winter 2024
Scarborough Campus — University of Toronto

Rice's Theorem

Vassos Hadzilacos

Rice's Theorem is a very general and powerful result. Roughly speaking, it states that it every non-trivial property of recognizable languages is undecidable. As immediate corollaries we have that it is impossible to decide if, given a Turing machine, its language is regular, or is context-free, or is empty, or is finite, or contains the string 0100, or contains only strings of odd length, and so on.

To state the theorem we must define what we mean by "property of recognizable languages". We use a very general and simple definition: A ***property P of recognizable languages*** is any set of recognizable languages. (Intuitively, if we think of the property P as a predicate that a language satisfies or not, the set in question is precisely the set of languages that satisfy the predicate.) Property P is ***non-trivial*** if it is neither empty nor the set of all recognizable languages: In other words, some recognizable languages have property P and some don't. Do not confuse the property $P = \varnothing$ with the property $P' = \{\varnothing\}$. The former is trivial (no language has it) while the second is non-trivial (exactly one language has it, namely the language that contains no strings).

Theorem 5.1 *For any non-trivial property P of recognizable languages, the set*

$$T_P = \{\langle M \rangle : \ \mathcal{L}(M) \in P\},$$

i.e., the set of codes of Turing machines whose languages have property P, is undecidable.

Proof. We consider two cases, depending on whether the empty language has property P.

Case 1. $\varnothing \notin P$. Since P is nontrivial, there is some recognizable language $L \in P$. Let M_L be a Turing machine that recognizes L; so, $\mathcal{L}(M_L) \in P$.

To prove that T_P is undecidable it suffices to show that $U \leq_m T_P$. For this, we show how, given $\langle M, x \rangle$, we can construct a Turing machine M' with the following properties:

(a) If M accepts x, then $\mathcal{L}(M') \in P$.
(b) If M does not accept x, then $\mathcal{L}(M') \notin P$.

Here is how to construct such an M' given $\langle M, x \rangle$:

M' := **on input** y:
1 run M on x
2 **if** M accepts x **then**
3 run M_L on y
4 **if** M_L accepts y **then** accept y
5 **else** reject y
6 **else** reject y

Clearly the above transformation of $\langle M, x \rangle$ to $\langle M' \rangle$ is computable. Furthermore, we have:

- If M accepts x, then line 1 terminates and the conditional on line 2 is true; so M' executes line 3. Thus, in this case, M' accepts y if and only if M_L accepts y. So, $\mathcal{L}(M') = \mathcal{L}(M_L) \in P$, and (a) is satisfied.
- If M does not accept x, then M' does not accept y, for any y: it either loops on y on line 1 or it outright rejects y in line 6. So, in this case $\mathcal{L}(M') = \varnothing$. By the hypothesis of the case, $\varnothing \notin P$, and so (b) is satisfied.

So, in this case we have shown that $U \leq_m T_P$, as wanted.

Case 2. $\varnothing \in P$. Consider the property $\overline{P}$ — i.e., the set of recognizable languages not in P. Since P is non-trivial, so is $\overline{P}$. So, by Case 1, $T_{\overline{P}}$ is undecidable. Thus, since the set of decidable languages is closed under complementation, $\overline{T_{\overline{P}}}$ is undecidable as well. But

$$\overline{T_{\overline{P}}} = \{\langle M\rangle : \ \mathcal{L}(M) \notin \overline{P}\} = \{\langle M\rangle : \ \mathcal{L}(M) \in P\} = T_P.$$

So, in this case too, T_P is undecidable. □

As with any powerful tool, it is important to understand not only how to use Rice's theorem but also the limits to its applicability. Note the following:

(1) Rice's theorem cannot be used directly to prove that a language is ***unrecognizable***; only that it is undecidable. It can be used to prove unrecognizability indirectly, in conjunction with other results. For example, we might be able to prove that L is unrecognizable by (a) using Rice's theorem to argue that $\overline{L}$ is undecidable, and (b) somehow reasoning that $\overline{L}$ is recognizable. It then follows that L is unrecognizable. (Do you see why?) In such cases you should not simply appeal to Rice's theorem; you should also state what other facts you use to obtain your desired result.

(2) Rice's theorem directly applies only to the set of codes of Turing machines whose languages have some property. It does not apply, for example, to the set

$$\big\{\langle M_1, M_2, k\rangle : \ M_1 \text{ and } M_2 \text{ are Turing machines and } k \in \mathbb{N} \text{ such that } |\mathcal{L}(M_1) \cap \mathcal{L}(M_2)| \geq k\big\}$$

(i.e., the set of encodings of a pair of Turing machines and a non-negative integer k such that the two Turing machines accept at least k strings in common.) This is an undecidable (but recognizable) set, but this fact does not follow directly from Rice's theorem.

(3) Rice's theorem is about properties of the ***languages*** recognized by Turing machines. It is ***not*** about

- ***syntactic*** properties of Turing machines (e.g., the set of codes of Turing machines that have exactly 15 states, which is obviously decidable); or
- properties of the ***behaviour*** of Turing machines, some of which are decidable (e.g., the set of codes of Turing machines that never move left on empty tape), while others are undecidable (e.g., the set of codes of Turing machines that halt on empty tape).