CS/ECE 374 A ✧ Fall 2023

“Homework” 12

“Due” Monday, December 4, 2023

This homework is *not* for submission. However, we are planning to ask a few (true/false, multiple-choice, or short-answer) questions about undecidability on the final exam, so we still strongly recommend treating these questions as regular homework. Solutions will be released next Monday.

---

1. Let $\langle M \rangle$ denote the encoding of a Turing machine M (or if you prefer, the Python source code for the executable code M). Recall that w^R denotes the reversal of string w. Prove that the following language is undecidable.

$$\textsc{SelfRevAccept} := \left\{ \langle M \rangle \;\middle|\; M \text{ accepts the string } \langle M \rangle^R \right\}$$

   Note that Rice's theorem does *not* apply to this language.

2. Let M be a Turing machine, let w be a string, and let s be an integer. We say that ***M accepts w in space s*** if, given w as input, M accesses ***at most*** the first s cells on its tape and eventually accepts. (If you prefer to think in terms of programs instead of Turing machines, “space” is how much memory your program needs to run correctly.)

   Prove that the following language is undecidable:

$$\textsc{SomeSquareSpace} = \left\{ \langle M \rangle \;\middle|\; M \text{ accepts at least one string } w \text{ in space } |w|^2 \right\}$$

   Note that Rice's theorem does *not* apply to this language.

   [Hint: The only thing you actually need to know about Turing machines for this problem is that they consume a resource called “space”.]

3. Prove that the following language is undecidable:

$$\textsc{Picky} = \left\{ \langle M \rangle \;\middle|\; \begin{array}{c} M \text{ accepts at least one input string} \\ \text{and } M \text{ rejects at least one input string} \end{array} \right\}$$

   Note that Rice's theorem does *not* apply to this language.

Solved Problem

4. Consider the language $\textsc{SometimesHalt} = \{\langle M \rangle \mid M \text{ halts on at least one input string}\}$. Note that $\langle M \rangle \in \textsc{SometimesHalt}$ does not imply that M *accepts* any strings; it is enough that M *halts* on (and possibly rejects) some string.

   (a) Prove that SometimesHalt is undecidable.

Solution (Rice): Let $\mathcal{L}$ be the family of all non-empty languages. Let N be any Turing machine that never halts, so $\textsc{Halt}(N) = \varnothing \notin \mathcal{L}$. Let Y be any Turing machine that always halts, so $\textsc{Halt}(Y) = \Sigma^* \in \mathcal{L}$. Rice's Halting Theorem immediately implies that $\textsc{SometimesHalt} = \textsc{HaltIn}(\mathcal{L})$ is undecidable. ■

Solution (closure): Let Encodings be the language of all Turing machine encodings (for some fixed universal Turing machine); this language is decidable. We immediately have $\textsc{Encodings} = \textsc{NeverHalt} \cup \textsc{SometimesHalt}$, or equivalently, $\textsc{NeverHalt} = \textsc{Encodings} \setminus \textsc{SometimesHalt}$.

The lectures notes include a proof that NeverHalt is undecidable. On the other hand, the existence of a universal Turing machine implies that Encodings is decidable. So Corollary 3(d) in the undecidability notes implies that SometimesHalt is undecidable. ■

Solution (reduction from Halt): We can reduce the standard halting problem to SometimesHalt as follows:

```
DecideHalt(⟨M, w⟩):
    Encode the following Turing machine M′:
        M′(x):
            (ignore x)
            run M on input w
    return DecideSometimesHalt(⟨M′⟩)
```

We prove this reduction correct as follows:

$\Longrightarrow$ Suppose M halts on input w.
Then M' halts on *every* input string x.
So DecideSometimesHalt must accept the encoding $\langle M' \rangle$.
We conclude that DecideHalt correctly accepts the encoding $\langle M, w \rangle$.

$\Longleftarrow$ Suppose M does not halt on input w.
Then M' diverges on *every* input string x.
So DecideSometimesHalt must reject the encoding $\langle M' \rangle$.
We conclude that DecideHalt correctly rejects the encoding $\langle M, w \rangle$.

■