Class 24: Halting Problems

Schedule

Problem Set Omega is due on **Sunday, 4 December** or **Tuesday, 6 December** (see problem set for details). It is not like the others, and counts as a "bonus" optional assignment.

The **final exam** is scheduled by the registrar for **Saturday, 10 December, 9am-noon** in our normal classroom. See the Final Exam Preparation handout for more information on the final and some **practice problems**.

Turing Machine Definitions

$$TM = (S, T \subseteq S \times \Gamma \rightarrow S \times \Gamma \times \textit{dir}, q_0 \in S, q_{Accept} \subseteq S)$$

S is a finite set (the "in-the-head" processing states)
Γ is a finite set (symbols that can be written on the tape)
$\textit{dir} = \{\textbf{Left}, \textbf{Right}, \textbf{Halt}\}$ is the direction to move on the tape.

An *execution* of a Turing Machine, $TM = (S, T \subseteq S \times \Gamma \rightarrow S \times \Gamma \times \textit{dir}, q_0 \in S, q_{Accept} \subseteq S)$, is a (possibly infinite) sequence of **configurations**, $(x_0, x_1, \ldots)$ where $x_i \in \textit{Tsil} \times S \times \textit{List}$ (elements of the lists are in the finite set of symbols, Γ), such that:

– $x_0 = (\textbf{null}, q_0, \textbf{input})$
– and all transitions follow the rules (need to be specified in detail).

Recognizing Languages

A Turing Machine, $M = (S, T \subseteq S \times \Gamma \rightarrow S \times \Gamma \times \textit{dir}, q_0 \in S, q_{Accept} \subseteq S)$, **accepts** a string *x*, if there is an execution of M that starts in configuration $(\textbf{null}, q_0, x)$, and terminates in a configuration, (l, q_f, r), where $q_f \in q_{Accept}$.

A Turing Machine, $M = (S, T \subseteq S \times \Gamma \rightarrow S \times \Gamma \times \textit{dir}, q_0 \in S, q_{Accept} \subseteq S)$, **recognizes** a language $\mathcal{L}$, if for all strings $s \in \mathcal{L}$, M accepts s, and there is no string $t \notin L$ such that M accepts t.

A Turing Machine, $M = (S, T \subseteq S \times \Gamma \rightarrow S \times \Gamma \times \textit{dir}, q_0 \in S, q_{Accept} \subseteq S)$, **decides** a language $\mathcal{L}$, if for all strings $s \in \mathcal{L}$, M accepts s, and for all strings $t \notin L$, M *terminates* in a non-accepting state.

A language $\mathcal{L}$ is **Turing-recognizable** if there is some Turing Machine that recognizes it. A language $\mathcal{L}$ is **Turing-decidable** if there is some Turing Machine that decides it.

Are all Turing-decidable languages Turing-recognizable?

Are all Turing-recognizable languages Turing-decidable?

Undecidable Languages

$$\text{SelfRejecting} := \{w \in \Sigma^* \mid w \notin \mathcal{L}(M(w))\}$$

where $M(w)$ is the Turing Machine described by string w if w describes a valid Turing Machine, otherwise, a $M(w)$ is a machine that rejects all inputs.

Is there a $M_{SR} = M(w_{SR})$ that recognizes the language SelfRejecting?

$$A_{TM} = \{(w, x) \mid M(w) \text{ accepts on input } x\}$$

Is the language A_{TM} Turing-recognizable?

Is the language A_{TM} Turing-decidable?

$$Halts_{TM} = \{(w, x) \mid M(w) \text{ terminates on input } x\}$$

```
def paradox():
   if halts('paradox()'):
       while True:
          pass
```

