Automata Theory (CS1.302)

Monsoon 2021, IIIT Hyderabad
22 November, Monday (Lecture 12)

Models of Computation

Turing Machines

Some Undecidable Languages (contd.)

The Halting Problem asks if we can construct a total TM H that accepts the language

$$H_{\text{TM}} = \{\langle M, w\rangle \mid M \text{ halts on input} w\}.$$

Such a TM, in fact, does not exist. We can prove this by first assuming that H exists, and then constructing from it a total TM A for $A_t extTM$ (which we know cannot exist). In other words, we solve the Halting Problem by reduction from the accepting problem.

More concretely, given H, we can construct

$$\begin{aligned}
A(\langle M, w\rangle) = &\text{Run } H(\langle M, w\rangle) \\
&\text{If } H \text{ accepts, run } M(w) \\
&\text{If } H \text{ rejects, reject.}
\end{aligned}$$

Note that $H_t extTM$ is *partially* decidable (like A_{TM}), as it can be recognised by simply simulating M on w.

Reduction

The key idea of the proof of undecidability of H_{TM} is reduction (in this case, from A_{TM}).

When a language X can be solved using a solver for Y, we say that X *reduces to* Y, or $X \preccurlyeq Y$.

Thus, if $A \preccurlyeq B$, then

- B is decidable $\implies$ A is decidable.

- A is undecidable $\implies$ B is undecidable.

Some problems that can be proved undecidable by reduction from H_{TM} are

$$E_{\text{TM}} = \{\langle M \rangle \mid L(M) = \Phi\},$$

$$EQ_{\text{TM}} = \{\langle M_1, M_2 \rangle \mid L(M_1) = L(M_2)\},$$

$$ALL_{\text{TM}}$$

Closure Properties of Decidable Languages

Recursive languages are closed under union; if R_1 and R_2 are decidable, then so is $R_1 \cup R_2$. We can prove this by constructing

$$\begin{aligned} M'(w) = &\text{Run } M_1(w) \text{ and } M_2(w) \\ &\text{Accept if either accepts.} \end{aligned}$$

The proofs for recursive languages being closed under intersection and complementation are analogous.

An important property is that L and $\overline{L}$ are both recursively enumerable iff L is recursive.
To prove one direction, if L is recursive, L is trivially also recursively enumerable. $\overline{L}$ can be decided by checking for L and giving the opposite output.
For the other direction, if L and $\overline{L}$ are recursively enumerable, check for both of them simultaneously. At least one will halt because for any w, either $w \in L$ or $w \in \overline{L}$. Then if L halts first, give the same input; and if $\overline{L}$ halts first, give the opposite output.

The two can be checked simultaneously using a time-sharing (or *dovetailing*) technique, which is to run each of them alternately for a finite number of steps.

Closure Properties of Recognisable Languages

Using dovetailing, it is easy to prove that recursively enumerable languages are also closed under union and intersection. However, they are *not* closed under complementation; the above proved result makes this clear.

The class **coRE** consists of languages whose complements are RE, *i.e.* $L \in \text{coRE} \iff \overline{L} \in \text{RE}$. Also, $R = \text{RE} \cap \text{coRE}$.

In fact, **RE** consists of partially decidable problems, and **coRE** of completely undecidable ones.