COL352 Lecture 29

Contents

1 Recap

2 Definitions

Definition 1

A language L is said to be co-Turing recognizable if its complement is Turing recognizable.

Definition 2

The membership language of Turing machines is defined as

$$A_{TM} = \{(\langle M \rangle, x) \mid \text{Turing machine } M \text{ accepts string } x\}$$

Definition 3

A DTM recognizing A_{TM} is called a universal TM.

Definition 4

$\mathbf{HALT} = \{(\langle M \rangle, x) \mid M \text{ halts on } x\}$.

Definition 5

Computable function: Let $f : \Sigma_1^* \to \Sigma_2^*$ be a function. We say f is computable if there exists a Turing machine C which does the following:

$\forall x \in \Sigma_1^*$, C on input x

1. halts,
2. leaves $f(x)$ on the tape,
3. and at the end of execution leaves the read/write head on the leftmost cell of the tape.

Definition 6

Let $L_1 \subseteq \Sigma_1^*, L_2 \subseteq \Sigma_2^*$ be two languages. A function $f : \Sigma_1^* \to \Sigma_2^*$ is called a mapping reduction from L_1 to L_2 if:

1. f is computable.
2. $\forall x \in \Sigma_1^*, x \in L_1 \iff f(x) \in L_2$.

We say that L_1 is mapping reducible to L_2, and write $L_1 \leq_m L_2$ if such a mapping reduction exists.

3 Content

Theorem 1

HALT is undecidable.

Proof. Assume **HALT** is decidable. Then construct a decider for A_{TM}. This gives a contradiction. Details follow:

Recall from the last class that $A_{TM} = \{(\langle M \rangle, x) \mid M \text{ accepts } x\}$.

Let H be a decider for **HALT**. Do the following: if H halts on $(\langle M \rangle, x)$, then reject, else run a universal Turing machine on the input.

So, consider the Turing machine D which, on input $(\langle M \rangle, x)$ does the following:

1. Use H to decide whether M halts on x. If H rejects $(\langle M \rangle, x)$, then reject.
2. (Now we know H accepts). Run a universal Turing machine U on $(\langle M \rangle, x)$ and output its answer.

Proof of correctness:

$(\langle M \rangle, x) \in A_{TM} \iff M \text{ accepts } x \iff H, U \text{ accept } (\langle M \rangle, x) \iff D \text{ accepts } (\langle M \rangle, x)$. If D accepts, we are done. Now note that for D to run forever on the input, since H is a decider, only U can run forever on the input. However, for U to run on the input, H has to accept it, so M halts on x, and hence U halts on $(\langle M \rangle, x)$, which is a contradiction. So D always halts, and is hence a decider. □

Now coming to computability, check that basic arithmetic operations are computable. Check that $f(G) =$ breadth first traversal of G is also computable.

Question 1

Show that the class of computable functions from Σ^* to Σ^* is closed under composition.

Consider the following situation: we have a computable function f from Σ_1^* to Σ_2^*, and there are languages L_1 and L_2 such that $f(L_1) \subseteq L_2$ and $f(\overline{L_1}) \subseteq \overline{L_2}$.

Now suppose TM R computes f, and L_2 is decidable. Then $x \in L_1 \iff f(x) \in L_2 \iff$ decider for L_2 accepts $f(x)$.

Example of a mapping reduction: $\overline{\mathbf{DIAG}} \leq_m \mathbf{A_{TM}}$, $f(w) = (w, w)$.

Theorem 2

Suppose $L_1 \leq_m L_2$. Then the following hold.

1. If L_2 is decidable, then L_1 is decidable.
2. If L_2 is recognizable, then L_1 is recognizable.
3. If L_2 is co-recognizable, then L_2 is also co-recognizable.

Proof.

1. $L_1 \leq_m L_2$, so there exists a computable function f such that $\forall x$, $x \in L_1 \equiv f(x) \in L_2$. Suppose TM R computes f. Let M_2 be a decider for L_2. Now consider the Turing machine M_1 which on input x does the following:
   (a) Use R to compute $f(x)$.
   (b) Run M_2 on $f(x)$ and return its answer.

   Since R, M_2 halt on every input, M_1 also halts on every input. Now $x \in L_1 \iff f(x) \in L_2 \iff M_2$ accepts $f(x) \iff M_1$ accepts x.
2. The same construction works (except that M_2 need not halt but R always halts).

□

Theorem 3

Equivalently,

Suppose $L_1 \leq_m L_2$. Then the following hold.

1. If L_1 is undecidable, then L_2 is undecidable.
2. If L_1 is unrecognizable, then L_2 is unrecognizable.
3. If L_1 is unco-recognizable, then L_2 is also unco-recognizable.

Goal: To prove L_2 is undecidable.

Recipe: Take a known undecidable language L_1 (for instance, $L_1 = A_{TM}$), and show that $L_1 \leq_m L_2$.

Example 1

Let $E_{TM} = \{w \mid M_w \text{ recognizes the empty language}\}$.

Claim 3.1

$\mathbf{DIAG} \leq_m E_{TM}$

Proof. Define $f(w)$ as follows. $f(w)$ is the description of a TM that does the following: On input x,

1. Erase x and write w on the tape.
2. Simulate M_w on w and return the answer.

Observe that f is computable. Also, $w \in \mathbf{DIAG} \iff M_w$ doesn't accept $w \iff M_{f(w)}$ doesn't accept any $x \iff f(w) \in E_{TM}$. □