Lecture 6

Gidon Rosalki

2025-05-04

1 Reminder of languages

We have RE, all the languages recognisable by Turing machines, R, the set of decidable languages by TMs, and REG, all the regular languages (recognisable by a DFA). These hold that $REG \subset R \subset RE$. Note that $HALT \in RE$. Let us define

$$\begin{aligned} CO - RE &= \left\{L : \overline{L} \in RE\right\} \\ &\quad \left(\overline{L} = \Sigma^* \setminus L \leftarrow L \subseteq \Sigma^*\right) \\ &= \{L : \text{ There exists a TM M that rejects the input } x \Leftrightarrow x \notin L\} \end{aligned}$$

Now, is it true that $R \subseteq CO-RE$? IT is true that $R \subset CO-RE$, but is there anything else? Anything that is within RE, but not R, or anything that is in CO-RE, but not in RE?

Theorem 1.

$$RE \cap CO - RE = R$$

Proof. It is trivial that

$$R \subseteq RE \cap CO - RE$$

from the previous lecture, so now we simply need to show the other direction:

$$RE \cap CO - RE \subseteq R$$

Since if $L \in RE \cap CO - RE$, then there is a TM M_L and $M_{\overline{L}}$ the recognise L and $\overline{L}$ accordingly. We may thus build the TM M_D (D for decide), that decides L by given an input x, M_D will run M_L and $M_{\overline{L}}$ on x a total of i steps for every $i \in \mathbb{N}$. If M_L will accept or reject, then M_D will accordingly accept / reject, and if $M_{\overline{L}}$ will accept or reject, then M_D will reject / accept accordingly. It is certain that at elast one of these will occur, and thus M_D will give the correct response. □

Is there a language $L \in CO - RE \setminus RE$? Yes: $\overline{HALT}$. $\overline{HALT} \in CO - RE \setminus RE$, since otherwise it would be in R, and thus both it and $HALT$ would be decidable, which is a contradiction to the fact that they are undecidable.

2 Reductions

Reductions are translating problems into other problems, and we may use them to say that a given problem is no more difficult than a different problem, given a certain function that translates between them. Consider $L \subseteq \Sigma^*, L' \subseteq \Gamma^*$, languages, and the TM that can recognise L, M. Thus, with the reduction Red, we can give Red a word $w \in \Sigma^*$, and thus recognise whether or not it is in fact in L' through using M upon it. Formally:

Definition 2.1 (Reduction)**.** $L \subseteq \Sigma^*, L' \subseteq \Gamma^*$. *A reduction machine from L' to L is the TM Red that stops on every input $x \in \Gamma^*$ and returns a word in Σ^* such that*

$$Red\,(x) \in L \Leftrightarrow x \in L'$$

If there exists such a TM, then we will say that there exists a reduction from L' to L, and we will write

$$L' \leq_m L$$

and the function $x \to Red\,(x)$ will be called the reduction function from L' to L.

Theorem 2 (Reduction theorem)**.** *If $L' \leq_m L$, then all the following are true:*

1. $L \in R \implies L' \in R$

2. $L \in RE \implies L' \in RE$

3. $L \in CO - RE \implies L' \in CO - RE$

Proof . Homework (the concept is that if M "solves" L, then $M \circ Red$ solves L') □

Theorem 3 ($HALT \leq A_{TM}$)**.** *Let*

$$A_{TM} = \{\langle M, w \rangle : M \ accepts \ w\}$$

Proof . The reduction will return, given an input $\langle M, w \rangle$ a coding $\langle M', w \rangle$, where M' is a TM that runs like M, apart from the fact that M' will also accept, even when M rejects (i.e., M' will accept whenever M stops, thus describing $HALT$).
Correctness: The described reduction may be described by a TM that always stops. $\langle M, w \rangle \in HALT$ **if and only if** M stops on w **if and only if** M' accepts w **if and only if** $\langle M', w \rangle \in A_{TM}$ □

Theorem 4.

$$A_{TM} \leq_m HALT$$

Proof . Given $\langle M, w \rangle$, the reduction will return $\langle M', w \rangle$, where M' is a TM that runs like M, however when M rejects, then M' enters an infinite loop.
Correctness: Computation - trivial, $\langle M, w \rangle \in A_{TM}$ **if and only if** M accepts w **if and only if** M' stops (and accepts, not that important) on w **if and only if** $\langle M', w \rangle \in HALT$ □

Theorem 5. *For every language $L \in RE$, it holds that $L \leq_m A_{TM}$*

Proof . Continued below □

Definition 2.2. *If $\mathcal{C}$ is a set of languages, and L is a language such that $L' \leq_m L$, for every $L' \in \mathcal{C}$, then we will say that L is $\mathcal{C}$-hard. If, additionally $L \in \mathcal{C}$, we will say that L is C-complete.*

Theorem 6. *A_{TM} is RE-complete*

Proof . $A_{TM} \in RE$. It remains to show if $L \in RE$, then $L \leq_m A_{TM}$. For such an L, there is a TM M_L, that recognises it. Given the input x for L, the reduction will return $\langle M_L, x \rangle$.
Correctness: Trivial, left as an exercise to the reader. □

Theorem 7. *$\overline{HALT}$ is $CO - RE$ complete*

Proof . Let there be $L \in CO - RE$. Therefore $\overline{L} \in RE$. Therefore

$$\overline{L} \leq_m A_{TM} \leq_m HALT \implies \overline{L} \leq_m HALT \implies L \leq_m \overline{HALT}$$

□

What words are in $\overline{HALT}$?

$$\overline{HALT} = \{\langle M, w \rangle : M \text{ does not stop on } w\} \cup E$$

Where E is the set of incorrect inputs (also known as nonsense, or figgldygrak).

Theorem 8. *Non-Halt is complete in $CO - RE$*

Proof . Firstly, is it true that Non-Halt $\in CO - RE$? Well Non-HAlt = Halt $\cup E$, where Halt $\in RE \wedge E \in RE$. So therefore Non-Halt $\in RE$. We will finish with a reduction from HALT to Non-Halt. Let us create a function such that for the words in $\overline{HALT}$, for those accepted are mapped to being accepted by Non-Halt, for those that are figgldygrak, they are also accepted by Non-Halt, and those rejected by $\overline{HALT}$, they too are rejected by Non-Halt. □

Let us define some more languages. Consider

$$Y - HALT = \{\langle M, w \rangle, Y : M \text{ stops on } w\}$$
$$N - HALT = \{\langle M, w \rangle, N : M \text{ does not stop on } w\}$$
$$NY - HALT = Y - HALT \cup N - HALT$$

Theorem 9. *NY-HALT is both RE-hard, and also $CO - RE$-hard.*

Proof RE. We will show a reduction from HALT, since for every other language in RE there is a reduction to HALT, so this is sufficient. Given $\langle M, w \rangle$, the reduction will return $\langle M, w \rangle, Y$. The correctness is trivial, and left as an exercise for the reader. □

Proof CO-RE. We will create a reduction from Non-Halt, as follows:

$$\langle M, w \rangle \rightarrow \langle M, w \rangle, N$$

□

We have thus created a language that is harder than all the languages in RE, and CO-RE. To fully demonstrate this, we must show that $NY - HALT \notin RE$. If it were, then from the reduction theorem $Non - Halt \in RE$, but we have already shown $Non - Halt \in CO - RE$, and thus $Non - Halt \in RE \cap CO - RE = R$, but this cannot be.

3 Things which we have not the time to study further

Consider a shape given by a list of corner points. Is it possible to tile a plane, using copies of this shape? Well, maybe, this problem is not decidable.

3.1 The theory of Turing and Church

Every computation machine (in our universe), can be simulated by a Turing Machine.

4 Mathematical logic

In 250BC Archimedes established that one can prove geometrical problems by assuming mathematical axioms. In 1880 Frege created a system of proofs, where everything is either an axiom, or everything is derived by rules. So, in our language, we may state that for all the proofs (that Frege could conceive) proof $\in R$. In 1900 Hilbert asked if it is possible to compute mathematics (compute as in TM, not 4 + 5), i.e. if every mathematical proof can be proven using Frege's framework. He kept asking, until he asked if you may decide a mathematical theorem by a computation, or in other words, if the language of mathematical theorems is in R? The answer is no (sadly). We may also ask if every mathematical theorem there exists a proof, or disproof. If so, then $CORRECT \in R$, (the collection of correct mathematical theorems), which we just stated was not the case. Then came Godel (with answers we don't like), and showed that given

$$DECIDABLE = \{\text{Theorems that may be proved or disproved}\}$$

the following theorem

Theorem 10.

$$DECIDABLE \in RE$$

Proof . Let us assume that T is a TM that decides $DECIDABLE$. We will show that this shows that $HALT \in R$. We will build a TM S, that works as follows: Given $\langle M, w \rangle$ S will answer the theorem "$\langle M \rangle$ does not stop on $\langle w \rangle$". If T answered "No", then S will answer "No". If T answered "yes", then S will run on the proof.
Since there exists a mathematical proof of the type "$\langle M \rangle$ does not stop on $\langle w \rangle$", which is correct, but has no proof. □