Lecture 7:
Other undecidable languages, Rice’s theorem, and Reductions
Valentine Kabanets
October 4, 2016

1 Semi-decidable vs. decidable

We know that a language may be semi-decidable but not decidable. However, if both $L$ and its complement $\overline{L}$ are semi-decidable then $L$ must in fact be decidable.

Claim 1. If a language $L$ and its complement $\overline{L}$ are both semi-decidable, then $L$ is decidable.

Proof. Let $M_L$ be a TM accepting $L$, and let $M_{\overline{L}}$ be a TM accepting $\overline{L}$. On input $x$, run both TMs “in parallel”, until one of them accepts. (At some finite point in time, one of the machines must accept as every input $x$ is either in $L$ or in $\overline{L}$.) If $M_L$ accepted, then halt and accept. If $M_{\overline{L}}$ accepted, then halt and reject. □

As a corollary, we get that the complement of $A_{TM}$ is not semi-decidable! Do you see why?

We also have the following.

Theorem 1. The class of decidable languages is closed under complementation. On the other hand, the class of semi-decidable languages is not closed under complementation.

Proof. Given a DTM $M$ deciding a language $L = L(M)$, construct a new DTM $M'$ by taking $M$ and swapping $q_{\text{accept}}$ and $q_{\text{reject}}$ states. It’s easy to see that the new DTM $M'$ accepts exactly those strings that are rejected by $M$, and rejects exactly those strings that are accepted by $M$. So, we have $L(M') = \overline{L}$, as required.

On the other hand, $A_{TM}$ is semi-decidable, but, as observed above, its complement is not semi-decidable. □

2 Examples of undecidable languages

Theorem 2. The language

$$E_{TM} = \{ \langle M \rangle \mid L(M) \text{ is empty} \}$$

is undecidable.

Proof. Proof by reduction from $A_{TM}$. Given input $\langle M, w \rangle$, design a TM $M'$ as follows:

$M'$: “On input $x$, simulate $M$ on input $w$. If $M$ accepts, then Accept.”

Observe that
1. if $M$ accepts $w$, then $L(M') = \Sigma^*$ (i.e., $M'$ accepts every input $x$).
2. if $M$ does not accept $w$, then $L(M') = \emptyset$.

Now, if we have a decider TM $R$ for the language $E_{TM}$, we can decide $A_{TM}$ as follows:

“On input $\langle M, w \rangle$,
1. Construct the TM $M'$ for this pair $\langle M, w \rangle$, as explained above.
2. Run $R$ on input $\langle M' \rangle$.
3. If $R$ accepts $\langle M' \rangle$, then Reject. If $R$ rejects $\langle M' \rangle$, then Accept.”

Theorem 3. The language

$$ALL_{TM} = \{\langle M \rangle \mid L(M) = \Sigma^*\}$$

is undecidable.

Proof. Suppose that $ALL_{TM}$ is decidable by $R$. Show how to decide $A_{TM}$.

On input $\langle M, w \rangle$, construct TM $M'$ as follows:

$M'$: “On input $x$, simulate $M$ on $w$, accepting if $M$ accepts $w$”.

Now, if $M$ accepts $w$, then $L(M') = \Sigma^*$; and if $M$ does not accept $w$, then $L(M') = \emptyset$.

So to decide $A_{TM}$, do the following:

“On input $\langle M, w \rangle$, construct TM $M'$ defined above. Run $R$ on input $\langle M' \rangle$. If $R$ accepts $\langle M' \rangle$, then Accept; otherwise, Reject.”

Since $A_{TM}$ is undecidable, we conclude that $R$ cannot exist.

3 Another example of undecidability

Theorem 4. The language

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

is undecidable.

Proof. Suppose it is decidable by some decider $R$. We reduce $E_{TM}$ to $EQ_{TM}$.

Given $\langle M \rangle$, construct $M_1 = M$ and $M_2 = M_0$, where $M_0$ is some fixed TM such that $L(M_0) = \emptyset$.

Clearly, we have $L(M) = \emptyset$ iff $L(M_1) = L(M_2)$.

So, to decide $E_{TM}$, we do the following:

“On input $\langle M \rangle$, construct $M_1$ and $M_2$, as described above. Run $R$ on $\langle M_1, M_2 \rangle$. If $R$ accepts, then Accept; otherwise, Reject.”

Since $E_{TM}$ is undecidable (as shown above), we conclude that $R$ cannot exist.
4 Rice’s Theorem

Generalizing the arguments above, we will prove that essentially every nontrivial property of TM languages is undecidable. More precisely,

**Theorem 5 (Rice’s theorem).** Any nontrivial property $P$ of TMs is undecidable.

Here a *property* is a collection of TM descriptions $⟨M⟩$ such that, for any two $M_1$ and $M_2$, if $L(M_1) = L(M_2)$ then either $⟨M_1⟩, ⟨M_2⟩ ∈ P$, or $⟨M_1⟩, ⟨M_2⟩ ∉ P$.

Nontrivial means that it is neither empty nor everything: some TM $M_1$ exists such that $⟨M_1⟩ ∈ P$, and some TM $M_2$ exists such that $⟨M_2⟩ ∉ P$.

Before we do the proof, consider the language $E_{TM} = \{⟨M⟩ \mid L(M) = ∅\}$. Verify that $E_{TM}$ satisfies the definition of a nontrivial property. Thus, Rice’s theorem implies that $E_{TM}$ is undecidable!

For an example of a non-property, consider the set of TM descriptions $⟨M⟩$ such that the length of the description $|⟨M⟩| > 100$. This set is not a property in the above sense because we can have two TMs $M_1$ and $M_2$ with $L(M_1) = L(M_2) = ∅$, but $|⟨M_1⟩| < 100$ while $|⟨M_2⟩| > 100$. Can you see how to design such $M_1$ and $M_2$?

**Proof of Rice’s Theorem.** Towards a contradiction, suppose some nontrivial property $P$ is decidable by $R$. We’ll show how to decide $A_{TM}$.

Assume, without loss of generality, that $⟨M_1⟩ ∈ P$ for a TM $M_1$ such that $L(M_1) = ∅$. Let $M_2$ be any TM such that $⟨M_2⟩ ∉ P$. (Such $M_2$ exists since $P$ is nontrivial.)

Given an instance $⟨M, w⟩$ of $A_{TM}$, do the following:

1. Construct TM $A$:
   
   $A$: “On input $x$, run $M$ on $w$. If $M$ accepts $w$, then simulate $M_2$ on $x$, accepting if $M_2$ accepts.”

2. Run $R$ on input $⟨A⟩$.

3. If $R$ accepts $⟨A⟩$, then Reject; If $R$ rejects $⟨A⟩$, then Accept.

   Note on the construction of TM $A$:

   • If $M$ accepts $w$, then $L(A) = L(M_2)$, and so $⟨A⟩ ∉ P$ (since $⟨M_2⟩ ∉ P$).

   • If $M$ does not accept $w$, then $L(A) = L(M_1) = ∅$, and so $⟨A⟩ ∈ P$ (since $⟨M_1⟩ ∈ P$).

   Thus, by being able to decide whether $⟨A⟩$ is in $P$ or is not in $P$, we can decide whether $M$ accepts $w$, or not. In other words, we can decide $A_{TM}$. A contradiction.

   **Justification** of “without loss of generality, can assume $⟨M_1⟩ ∈ P$, with $L(M_1) = ∅$”:

   Take some fixed $M_1$ such that $L(M_1) = ∅$. Either $⟨M_1⟩ ∈ P$, or $⟨M_1⟩ ∉ P$. If it is in $P$, then we’re done. If $⟨M_1⟩ ∉ P$, then $⟨M_1⟩ ∈ \overline{P}$, where $\overline{P}$ is the complement of $P$. It’s easy to see that $\overline{P}$ is also a property if $P$ is a property, and that $\overline{P}$ is nontrivial if $P$ is nontrivial. Then we argue about the property $\overline{P}$ as before, reaching the conclusion that $\overline{P}$ is undecidable. Since $\overline{P}$ is decidable iff its complement is decidable, we get that $P$ is undecidable as well.

---

1One can define an *equivalence relation* $≡$ on TM descriptions: $⟨M_1⟩ \equiv ⟨M_2⟩$ iff $L(M_1) = L(M_2)$. Then a property $P$ can be thought of as a collection of *equivalence classes* under the equivalence relation $≡$. 
5 Reductions

We will consider a special kind of reductions: mapping reductions.

**Definition 1.** Language $A$ is $m$-reducible to $B$ (denoted $A < B$) if there is a computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that, for every $x \in \Sigma^*$,

$$x \in A \iff f(x) \in B.$$ 

**Theorem 6.** If $A < B$, and $B$ is decidable, then so is $A$. If $A < B$, and $B$ is semi-decidable, then so is $A$.

The contrapositive: If $A < B$, and $A$ is not (semi-) decidable, then neither is $B$.

**Remark 1.** You can interpret “$<$” as saying “less hard than”.

5.1 Examples

$E_{TM} < EQ_{TM}$ via the reduction $f$ such that $f(\langle M \rangle) = \langle M, M_\emptyset \rangle$, where $M_\emptyset$ is some fixed TM such that $L(M_\emptyset) = \emptyset$. (Check that this is indeed a reduction!)

**Theorem 7.** $EQ_{TM}$ is not semi-decidable.

*Proof.* We reduce the complement of $A_{TM}$ to $EQ_{TM}$. Given $\langle M, w \rangle$, define $f(\langle M, w \rangle) = \langle M_1, M_2 \rangle$ where

- $M_1$: “On input $x$, simulate TM $M$ on input $w$, accepting if $M$ accepts”.
- $M_2 = M_\emptyset$ (where $M_\emptyset$ accepts the empty language).

Note that $L(M_1) = \Sigma^*$, if $M$ accepts $w$; and $L(M_1) = \emptyset$, if $M$ does not accept $w$. So this is indeed a reduction. Since we know that the complement of $A_{TM}$ is not semi-decidable, we conclude that $EQ_{TM}$ is not semi-decidable as well. \qed

6 Hardness of INF

Consider the language $INF = \{ \langle M \rangle \mid L(M) \text{ is infinite} \}$.

We will prove the following:

1. $INF$ is undecidable.
2. $INF$ is not semi-decidable.
3. The complement of $INF$ is not semi-decidable.

To prove $INF$ is undecidable we can either refer to Rice’s theorem (arguing that $INF$ is a non-trivial property), or give a direct reduction, e.g., $A_{TM} < INF$ as follows.

**Theorem 8.** $A_{TM} < INF$.

*Proof.* Given $\langle M, w \rangle$, construct $M'$: “On input $x$, simulate $M$ on $w$. If $M$ accepts $w$, then Accept.”

Clearly, $M$ accepts $w$ iff $L(M') = \Sigma^*$ is infinite. (If $M$ does not accept $w$, then $L(M') = \emptyset$.) \qed
To prove $\text{INF}$ is not semi-decidable, we reduce from $\overline{A_{TM}}$ which is known to be non-semi-decidable.

**Theorem 9.** $\overline{A_{TM}} < \text{INF}$.

**Proof.** Given $\langle M, w \rangle$, construct

$M'$: “On input $x$, simulate $M$ on $w$ for $|x|$ steps. If $M$ accepts $w$ within $|x|$ steps, then
Reject $x$. If $M$ does not accept $w$ within $|x|$ steps, then Accept $x$.”

If $M$ does not accept $w$, then $M'$ will accept every $x$, and so $L(M') = \Sigma^*$ is infinite.

Suppose $M$ accept $w$. Then $M$ accepts $w$ within some $t$ number of steps, where $t$ is a constant dependent on $M$ and $w$. We get that for every input $x$ of length $|x| < t$, our simulation of $M$ on $w$ for $|x|$ steps will not accept, and so $M'$ accepts $x$. On the other hand, for every $x$ of length $|x| \geq t$, our simulation of $M$ on $w$ for $|x|$ steps will complete with success, and so $M'$ will reject $x$. Note that the number of $x$’s that $M'$ accepts is finite (all strings of length less than $t$, which is $2^t - 1$, a constant dependent on $M$ and $w$). So, $L(M')$ is finite in this case.

Thus we get that $M$ accepts $w$ iff $L(M')$ is finite. \hfill $\Box$

Finally, to prove that the complement of $\text{INF}$ is not semi-decidable, we need to give a reduction from $\overline{A_{TM}}$ to $\text{INF}$. By the following easy result (Theorem 10 below), this is equivalent to giving a reduction from $A_{TM}$ to $\text{INF}$. We have given such a reduction earlier (see Theorem 8)! So we’re done.

**Theorem 10.** Let $A$ and $B$ be any two languages. We have $A < B$ iff $\overline{A} < \overline{B}$.

The proof is a simple exercise!