Reminder

We continue to study the hardness of various computational problems. We currently have the following hierarchy of classes of languages:

\[
\emptyset \subseteq \text{REG} \subseteq \text{CFL} \subseteq \text{DEC} \subseteq \text{RECOG} \subseteq \mathcal{L}
\]

where \(\text{REG}\) denotes the set of regular languages, \(\text{CFL}\) the set of context-free languages, \(\text{DEC}\) the set of decidable languages, \(\text{RECOG}\) the set of \(T\)-recognizable languages and \(\mathcal{L}\) is the set of all possible languages. Let \(\text{UNDEC}\) denote the set of undecidable languages and let \(\text{UNRECOG}\) denote the set of \(T\)-unrecognizable languages. Also, let \(\text{CO-RECOG}\) the set of all languages whose complement is \(T\)-recognizable. As a reminder, here are some examples in the boundaries of the hierarchy:

- \(a^*b^* \in \text{REG}\)
- \(\{a^k b^k | k \geq 1\} \in \text{CFL} \setminus \text{REG}\)
- \(\{a^k b^k c^k | k \geq 1\} \in \text{DEC} \setminus \text{CFL}\)
- \(A_{TM} \in \text{RECOG} \setminus \text{DEC}\)
- \(\overline{A_{TM}} \in \mathcal{L} \setminus \text{RECOG}\)

Note that \(A_{TM} \in \text{UNDEC}\) and \(\overline{A_{TM}} \in \text{UNRECOG}\). The former was derived by the the diagonalization method while the latter follows directly from these results:

**Lemma 1.** For any language \(L\), \(\overline{L} \in \text{DEC} \iff L, \overline{L} \in \text{RECOG}\).

**Proof sketch:** The forward direction is trivial while the backward direction requires more work. To prove it, we need to build a decider \(D\) that recognizes \(L\). This can be accomplished by having \(D\) running in parallel the two Turing machines \(M_L\) and \(M_{\overline{L}}\) that recognize \(L\) and \(\overline{L}\) respectively. Since every input string is accepted by either \(M_L\) and \(M_{\overline{L}}\), \(D\) will accept if \(M_L\) accepts or will reject if \(M_{\overline{L}}\) accepts. With a very similar procedure we can also build a decider for \(\overline{L}\), which proves that both \(L\) and \(\overline{L}\) are decidable.

A direct application of the above lemma gives us the following:

**Lemma 2.** For any language \(L\) if \(L \in \text{RECOG} \setminus \text{DEC}\), then \(\overline{L} \in \text{UNRECOG}\).

From which follows directly that \(\overline{A_{TM}}\) is \(\text{UNRECOG}\). These two lemmas also gives us an interesting graphical interpretation about decidable, \(T\)-recognizable and \(T\)-unrecognizable languages. The set \(\text{DEC}\) is just the intersection of the sets \(\text{RECOG}\) and \(\text{CO-RECOG}\). Also, it holds that if \(L \in \text{RECOG} \setminus \text{DEC}\), then \(\overline{L} \in \text{CO-RECOG} \setminus \text{DEC}\) and vice-versa:
In addition, there are languages such as $EQ_{TM}$ that are even higher in the hierarchy because neither this language nor its complement are T-recognizable.

**Mapping reducibility**

Since we are currently focusing on the Turing Machine as our model of computation, we need a tool that can help us studying its power and limits. The first interesting examples outside of the capabilities of a TM were that $A_{TM}$ was undecidable and $\overline{A_{TM}}$ T-unrecognizable. However, there is a non-trivial amount of work needed to prove these results. That is where mapping reducibility comes into play. It is a tool to compare the hardness of different problems, allowing us to prove certain languages are undecidable or even T-unrecognizable, just like the pumping lemma was a tool for proving that languages were not regular or CFLs.

Recall that a language $A$ is **mapping reducible** to a language $B$, denoted by $A \leq_m B$, if and only if there exists a computable function or reduction $f : \Sigma^* \rightarrow \Sigma^*$ such that $a \in A \iff f(a) \in B$.

Let’s unpack this definition. A computable function means that there exists a Turing machine that if it starts with $a$ on its input tape, it will run and halt with just $f(a)$ on its tape. Intuitively, this means that there is an algorithm that computes $f(a)$ given any string $a$. The property that $a \in A \iff f(a) \in B$ means that $f$ maps every string inside of $A$ to a string inside of $B$ and every string outside of $A$ to a string outside of $B$:
Note that this definition does not say anything about $f$ being one-to-one or onto. In particular, a simple computable reduction $f$ that just assigns everything in $A$ to single string in $B$ and everything outside of $A$ to a single string outside of $B$ is still valid.

This notion of a reduction enjoys the following nice and intuitive properties (that one should prove):

1. If $A \leq_m B$, then $A$ is no harder to solve than $B$. In fact, if there exists a solver for $B$ that of a certain caliber, then there exists one for $A$ of the very same caliber. This gives us the following:
   - $B \in DEC \implies A \in DEC$, or $A \in UNDEC \implies B \in UNDEC$.
   - $B \in RECOG \implies A \in RECOG$, or $A \in UNRECOG \implies B \in UNRECOG$.
2. It is possible to have $A \leq_m B$ and $B \leq_m A$ which would imply that the two are equally easy/hard.
3. Mapping reducibility is transitive, that is, if $A \leq_m B$ and $B \leq_m C$, then $A \leq_m C$. Intuitively, if $A$ is no harder than $B$ and $B$ is no harder than $C$, then $A$ is no harder than $C$.
4. An unexpected but wonderful property: $A \leq_m B \iff \overline{A} \leq_m \overline{B}$. In fact, if $A \leq_m B$ by virtue of the reduction $f$, the same reduction is a witness to the fact that $\overline{A} \leq_m \overline{B}$.

From these properties we can derive very useful results. For instance, it follows from property 1 that to prove a language $B$ is undecidable or T-unrecognizable, all we need to show is that $A_{TM} \leq_m B$ or $\overline{A_{TM}} \leq_m B$ respectively. This fact gives us a new tool for proving languages are not decidable nor T-recognizable, just like the pumping lemma was useful for proving languages were not regular or not CFLs.

However, it is important to know that the other direction might not be true. Our notion of a problem being harder than another does not mean that there is a reduction between them. $A_{TM}$ and $\overline{A_{TM}}$ are an example of this. Even though it is true that $\overline{A_{TM}}$ is undecidable and even T-unrecognizable, one can prove that there is no mapping reduction from $A_{TM}$ to $\overline{A_{TM}}$ and vice-versa by properties 1 and 4.

Next, we show some examples for proving that languages are undecidable or T-unrecognizable.

**Example 1 — Will a 2-tape TM used its second tape?**

Consider 2-tape TMs as TMs that have a standard input tape and an additional workspace tape. Both tapes are read-write tapes. Let

$$L_{2\text{tape}} = \{ \langle N, w \rangle \mid N \text{ is a 2-tape TM that uses its second tape while running on } w \}$$

Show that $L_{2\text{tape}} \in UNDEC$. 
Solution

As mentioned before, we can show that $L_{2tape} \in \text{UNDEC}$ by showing that $A_{TM} \leq_m L_{2tape}$. In other words, we need to come up with an algorithm that transforms a pair $\langle M, w \rangle$ of a 1-tape TM and a string into a pair $\langle N, x \rangle$ of a 2-tape TM and a string such that $M$ accepts $w$ if and only if $N$ uses its second tape when running on $x$. However, a key fact here is that a 2-tape TM can simulate a normal TM with only one tape! (not surprising but useful). Thus, we can have $N$ to simply simulate $M$ on $w$ and use its second tape only if $M$ accepts $w$.

More formally, we describe here the reduction $f$ from $A_{TM}$ to $L_{2tape}$. Let $M$ be a standard 1-tape TM and let $w$ be a string. We have $f(\langle M, w \rangle) = \langle N_M, w \rangle$ where $N_M$ is a 2-tape TM defined as follows:

- On input $w$, $N_M$ simulates $M$ on $w$ on the input (first) tape of $N_M$.
- If $M$ ever halts and accepts $w$, then $N_M$ writes a symbol on its second tape.

Firstly note that $f$ is computable because we are basing the description of $N_M$ by using the description of $M$. Furthermore, $\langle M, w \rangle \in A_{TM} \iff \langle N_M, w \rangle \in L_{2tape}$. This shows that $A_{TM} \leq_m L_{2tape}$ and hence completing the proof.

Example 2 — Are the languages of two different TMs equal?

Let $EQ_{TM} = \{ \langle M_1, M_2 \rangle | M_1 \text{ and } M_2 \text{ are TMs and } L(M_1) = L(M_2) \}$

Show that both $EQ_{TM}$ and $\overline{EQ_{TM}} \in \text{UNRECOG}$.

Solution

Let’s start by showing that $EQ_{TM} \in \text{UNRECOG}$. As mentioned before, we just need to show that $\overline{A_{TM}} \leq_m EQ_{TM}$. This means that we need to come up with a computable function $f$ that maps $\langle M, w \rangle$ to a pair of TMs $\langle M_1, M_2 \rangle$ with the following property:

- $\langle M, w \rangle \in A_{TM}$, then $L(M_1) = L(M_2)$. In other words, if $M$ rejects $w$ then the language of $M_1$ and $M_2$ are the same.
- $\langle M, w \rangle \notin A_{TM}$, then $L(M_1) \neq L(M_2)$. In other words, if $M$ accepts $w$ then the language of $M_1$ and $M_2$ are different.

The challenge here is to come up with such $M_1$ and $M_2$. Since $M_1$ and $M_2$ have to be somewhat related to $M$ and $w$, it might be a good idea to consider the following simple TM $N_{M,w}$:

- On input $x$, erase $x$ from the tape and simulate $M$ on $w$.
- If $M$ accepts, $N_{M,w}$ also accepts. If $M$ halts and rejects, $N_{M,w}$ also halts and rejects.

Notice that if $M$ accepts $w$, then $L(N_{M,w}) = \Sigma^*$ because every input will be accepted by $N_{M,w}$. Similarly, if $M$ rejects $w$ then all inputs will get rejected by $N_{M,w}$, meaning that $L(N_{M,w}) = \emptyset$. Thus, by replacing $M_1$ with $N_{M,w}$ in the above expressions we obtain that we would need to build a $M_2$ such that:
• $M$ rejects $w$ implies $\emptyset = L(M_2)$.
• $M$ accepts $w$ implies $\Sigma^* \neq L(M_2)$.

This is trivially accomplished if $M_2 = M_{\text{reject}}$, where $M_{\text{reject}}$ is a TM that rejects immediately regardless of the input. Therefore, $f(\langle M, w \rangle) = \langle N_{M,w}, M_{\text{reject}} \rangle$ is a valid reduction and it is also computable since it uses the description of $M$ and the string $w$ to create the description of $N_{M,w}$. This proves that $\overline{A_{TM}} \leq_m E_{TM}$ and that $E_{TM} \in \text{UNRECOG}$, as desired.

Similarly, we can prove that $\overline{E_{TM}} \in \text{UNRECOG}$ with the same proof but with the reduction being $f(\langle M, w \rangle) = \langle N_{M,w}, M_{\text{accept}} \rangle$, where $M_{\text{accept}}$ is the TM that accepts immediately regardless of the input.

**Example 3 — Is the language of a TM regular?**

Let $REG_{TM} = \{\langle N \rangle \mid N \text{ is a TM such that } L(N) \in \text{REG}\}$

Show that $REG_{TM} \in \text{UNDEC}$.

**Solution**

We show that $REG_{TM} \in \text{UNDEC}$ by showing that $A_{TM} \leq_m REG_{TM}$. We describe here the reduction $f$ from $A_{TM}$ to $REG_{TM}$. Let $M$ be a TM and let $w$ be a string. We have $f(\langle M, w \rangle) = \langle N_{M,w} \rangle$ where $N_{M,w}$ is a TM defined as follows:

- On input $w'$, $N_{M,w}$ first checks if $w' \in \{0^k1^k : k \geq 0\}$. If so, it accepts $w'$ and halts.
- If not, $N_{M,w}$ simulates $M$ on $w$. If $M$ ever halts and accepts $w$, then $N_{M,w}$ accepts $w'$ and halts.

We motivate the above construction as follows. The correctness of the reduction stipulates that $L(N_{M,w}) \in \text{REG} \iff \langle M, w \rangle \in A_{TM}$. Let us say we begin designing the reduction with the following goals in mind:

- If $\langle M, w \rangle \in A_{TM}$, then $L(N_{M,w}) = \Sigma^* \in \text{REG}$.
- If $\langle M, w \rangle \notin A_{TM}$, then $L(N_{M,w}) = \{0^k1^k : k \geq 0\} \notin \text{REG}$.

Notice that a reduction that respects the above conditions would be correct. We now argue that our reduction does in fact satisfy the above conditions. Firstly, $\{0^k1^k : k \geq 0\} \subset L(N_{M,w})$ regardless of whether $\langle M, w \rangle \in A_{TM}$ or not. This justifies the first step in the description of $N_{M,w}$ which accepts all $w' \in \{0^k1^k : k \geq 0\}$. Now, by definition, $N_{M,w}$ accepts other $w'$ if and only if $\langle M, w \rangle \in A_{TM}$. This justifies the second step in the description of $N_{M,w}$ which accepts other $w'$ if $M$ accepts $w$. This shows that $\langle M, w \rangle \in A_{TM} \iff \langle N_{M,w} \rangle \in REG_{TM}$, as desired. Finally, $f$ is computable because it uses the description of $M$ and $w$ to build the description of $N_{M,w}$. Thus, $A_{TM} \leq_m REG_{TM}$ and $REG_{TM} \in \text{UNDEC}$.

Note that the above reduction can be trivially modified to show that $CFL_{TM} \in \text{UNDEC}$ where

$$CFL_{TM} = \{\langle N \rangle : N \text{ is a TM such that } L(N) \in CFL\}$$