Recitation 7 - $\mathcal{R}$ vs. $\mathcal{RE}$

Orit Moskovich
Gal Rotem

Tel Aviv University

December 2, 2015
Overview

1. $R$, $RE$ and $co-RE$

2. Closure Properties

3. Diagonalization
Accept vs. Decide

Definition 1
A TM $M$ accepts a language $L$ if:
1. $w \in L$, then $M$ enters $q_{accept}$
2. $w \notin L$, then $M$ can either enter $q_{reject}$ or loop

Definition 2
A TM $M$ decides a language $L$ if:
1. $w \in L$, then $M$ enters $q_{accept}$
2. $w \notin L$, then $M$ enters $q_{reject}$
Definition 3

$\mathcal{RE}$ - the class of enumerable languages, i.e.

\[ \forall L \in \mathcal{RE}. \exists M. M \text{ accepts } L \]

Definition 4

$\mathcal{R}$ - the class of decidable languages, i.e.

\[ \forall L \in \mathcal{R}. \exists M. M \text{ decides } L \]

Definition 5

$\text{co} - \mathcal{RE}$ - the class of of languages whose complement is enumerable, i.e.

\[ \forall L \in \text{co} - \mathcal{RE}. \exists M. M \text{ accepts } \overline{L} \]
The Acceptance Problem

**Theorem 6**

A language is decidable $\iff$ it is Turing-recognizable and co-Turing-recognizable

i.e.,

$$\mathcal{R} = \mathcal{RE} \cap \text{co-RE}$$

**Theorem 7**

$$A_{TM} = \{\langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w \} \notin \mathcal{R}$$

**Corollary 8**

$$\overline{A_{TM}} = \{\langle M, w \rangle \mid M \text{ is a TM and } M \text{ doesn’t accept } w \} \notin \mathcal{RE}$$
Warm Up

Theorem 9

*Every CFL is decidable*

Proof.

Let \( L \) be a CFL.

**Goal:** Build a TM that decides \( L \).

First try: Convert a PDA for \( L \) into a TM (how?)

Why is that a bad idea? If the PDA is non-deterministic, then some branches of the PDA’s computation might never halt. Thus, in such case, the simulating TM will also loop.

Second (and better) try: Let \( G \) be a CFG for \( L \). Design a TM \( M_G \) that decides \( L \) as follows:
Proof.

Define $A_{CFG} = \{ \langle G, w \rangle \mid G \text{ is a CFG and } G \text{ generates } w \}$. 

Recall, we already built an algorithm that on input CFG $G$ and input word $w$, outputs accept if $G \Rightarrow^* w$, and reject otherwise.

Thus, $A_{CFG} \in \mathcal{R}$. Denote by $S_1$ the TM deciding $A_{CFG}$.

Pseudocode $M_G$

On input $w$

1. Run $S_1$ on $\langle G, w \rangle$
2. If $S_1$ accepts, accept; If $S_1$ rejects, reject

Thus, $M_G$ is a deciding TM for $L$. 

Orit Moskovich and Gal Rotem (TAU)
Example 10

Show that $\mathcal{R}$ is closed under $\oplus$ operation.
i.e., if $L_1, L_2 \in \mathcal{R}$, then

$$L_1 \oplus L_2 = \{ w \mid w \in L_1 \setminus L_2 \text{ or } w \in L_2 \setminus L_1 \} \in \mathcal{R}$$

Proof #1 - Using closure properties.

Let $L_1, L_2 \in \mathcal{R}$.
Then, $L_1 \oplus L_2 = (L_1 \setminus L_2) \cup (L_2 \setminus L_1) = (L_1 \cap \overline{L_2}) \cup (L_2 \cap \overline{L_1})$.

Since $\mathcal{R}$ is closed under complement,
$$\overline{L_1}, \overline{L_2} \in \mathcal{R}$$

Since $\mathcal{R}$ is closed under intersection and union,
$$(L_1 \cap \overline{L_2}) \cup (L_2 \cap \overline{L_1}) \in \mathcal{R}$$
Example - Closure Under $\oplus$ Operation

Proof #2 - Direct construction.

Let $L_1, L_2 \in \mathcal{R}$. Therefore, there exist TMs $M_1, M_2$ that decide $L_1, L_2$, respectively.

We build a TM $M_\oplus$ that decides $L_1 \oplus L_2$ as follows:

Pseudocode $M_\oplus$

On input $w$

1. Run $M_1$ on $w$
2. Run $M_2$ on $w$
3. If $M_1$ accepts and $M_2$ rejects, accept;
   If $M_1$ rejects and $M_2$ accepts, accept;
   Otherwise, reject

Note: $M_\oplus$ actually returns $M_1(w) \oplus M_2(w)$. 
Example

Example 11

Let \( \textit{EVEN}_{DFA} = \{ \langle D \rangle \mid D \text{ is a DFA and } L(D) \text{ contains a word of even length} \} \)

Prove that \( \textit{EVEN}_{DFA} \) is decidable

Reminder

Define \( \textit{E}_{DFA} = \{ \langle D \rangle \mid D \text{ is a DFA and } L(D) = \emptyset \} \).

Then, \( \textit{E}_{DFA} \in \mathcal{R} \).
Proof.
Denote by $S_2$ the TM deciding $E_{DFA}$.
We build a TM $M_{even}$ that decides $EVEN_{DFA}$ as follows:

**Pseudocode $M_{even}$ deciding $EVEN_{DFA}$**

On input $\langle D \rangle$
1. Construct a DFA $D_{even}$ for $(00 \cup 01 \cup 10 \cup 11)^*$ (how?)
2. Construct a DFA $D'$ of $D \cap D_{even}$ (how?)
3. Run $S_2$ on $\langle D' \rangle$
4. If $S_2$ accepts, reject; If $S_2$ rejects, accept

Correctness
$L(D)$ contains a word of even length $\iff L(D \cap D_{even}) \neq \emptyset$
Example

Example 12

Let $\text{INF\_EVEN}_{DFA} = \{ \langle D \rangle \mid D \text{ is a DFA and } L(D) \text{ contains infinite number of even-length strings} \}$

Prove that $\text{INF\_EVEN}_{DFA}$ is decidable

Reminder

Define $\text{INF}_{DFA} = \{ \langle D \rangle \mid D \text{ is a DFA and } L(D) \text{ is infinite} \}$. Then, $\text{INF}_{DFA} \in \mathcal{R}$. 
Example

Proof.

Denote by $S_3$ the TM deciding $\text{INF}_{DFA}$.

We build a TM $M_{\text{inf\_even}}$ that decides $\text{INF\_EVEN}_{DFA}$ as follows:

**Pseudocode $M_{\text{inf\_even}}$ deciding $\text{INF\_EVEN}_{DFA}$**

On input $\langle D \rangle$

1. Construct a DFA $D_{\text{even}}$ for $(00 \cup 01 \cup 10 \cup 11)^*$
2. Construct a DFA $D'$ of $D \cap D_{\text{even}}$
3. Run $S_3$ on $\langle D' \rangle$
4. If $S_3$ accepts, accept; If $S_3$ rejects, reject

**Correctness**

$L(D)$ contains infinite number of even-length strings $\iff L(D \cap D_{\text{even}})$ is infinite
Example 13

Show that for every infinite language $L$, there exists a sub-language $L'$ of $L$ that is not Turing-recognizable (specifically, $L'$ is undecidable)

Proof.

We prove the claim using counting arguments:

- We saw in class that the set of all Turing machines is **countable**
- Let $L'$ be the set of all sub-languages of $L$
- Thus, we aim to show that $L'$ is **uncountable**

$L = \{ \varepsilon, 0, 01, 10, 000, \ldots \}$

$L \supseteq L' = \{ 0, 01, 000, \ldots \}$

$\chi_{L'} = 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ \ldots$

We found a correspondence between $L'$ and $B$ (the set of all infinite binary sequences), which is an uncountable set.
Example

Example 14
Show that there exists an infinite language $L$, such that every infinite sub-language $L'$ of $L$ is undecidable

Proof.

Goal: To construct a language $L$ such that for every infinite decidable language $A$, $A \not\subseteq L$.

We will use diagonalization to prove the claim.
Example

Proof (Cont.)

Let $\mathcal{D}$ be the set of infinite decidable languages. Thus, $\mathcal{D}$ is countable.

Let $L_1, L_2, \ldots$ be a list of all infinite decidable languages.

We define a sequence of integers $n_0, n_1, \ldots$ as follows:

1. $n_0 = 0$
2. Let $w_k$ be the shortest string in $L_k$ such that $n_{k-1} < |w_k|$ (why is that a correct definition?)
3. Define $n_k = |w_k| + 1$ (thus, $n_{k-1} < |w_k| < n_k$)

Now, define $L = \{1^{n_1}, 1^{n_2}, \ldots \}$. 
Proof (Cont.)

Recall, \( L = \{1^{n_0}, 1^{n_1}, 1^{n_2}, \ldots\} \), where \( \forall k. \ n_{k-1} < |w_k| < n_k \). Also, \( L_1, L_2, \ldots \) is the list of all infinite decidable languages.

Claim: Every infinite decidable language \( A, A \not\subseteq L \).

Proof: Suppose for contradiction that there exists an infinite decidable subset of \( L \), i.e., \( \exists i. L_i \subseteq L \).

Examine \( w_i \) defined in previous slide:

- \( w_i \in L_i \)
- \( n_{i-1} < |w_i| < n_i \implies w_i \notin L \)

Contradiction.
Example 15

Prove:

$A$ is Turing recognizable $\iff$ there exists some decidable language $B$ such that $A = \{ x \mid \exists w \in \Sigma^*. \langle x, w \rangle \in B \}$

1. Let $B$ be a decidable language such that $A = \{ x \mid \exists w \in \Sigma^*. \langle x, w \rangle \in B \}$. Denote by $M_B$ the TM deciding $B$.

We build a TM $M$ that accepts $A$:

**Pseudocode $M$ accepting $A$**

On input $x$

1. For every $w \in \Sigma^*$:
2. Run $M_B$ on $\langle x, w \rangle$
3. If $M_B$ accepts, accept
Let $M$ be a TM that accepts $A$.

Define $B = \{ \langle x, 1^t \rangle \mid M \text{ accepts } x \text{ within } t \text{ steps} \}$.

Claim: $B$ is a decidable language such that

$$A = \{ x \mid \exists w \in \Sigma^*. \langle x, w \rangle \in B \}$$

Proof:

- $B$ is clearly decidable (why?)
- $x \in A \iff \exists t \in \mathbb{N}. M \text{ accepts } x \text{ within } t \text{ steps}$. Thus, exists $w$ such that $\langle x, w \rangle \in B$ ($w = 1^t$).

Therefore, $A = \{ x \mid \exists w \in \Sigma^*. \langle x, w \rangle \in B \}$
Example

Example 16
Let $UL_{DFA} = \{ \langle D, q \rangle \mid D \text{ is a DFA and } q \text{ is a useless state} \}$
Where state $q$ in DFA $D$ is useless if it is never entered on any input.
Prove that $UL_{DFA}$ is decidable

Reminder
Define $E_{DFA} = \{ \langle D \rangle \mid D \text{ is a DFA and } L(D) = \emptyset \}$. Then, $E_{DFA} \in \mathcal{R}$. 
Proof.

Denote by $S_2$ the TM deciding $E_{DFA}$.

We build a TM $M_{ul}$ that decides $UL_{DFA}$ as follows:

**Pseudocode $M_{ul}$ deciding $UL_{DFA}$**

On input $\langle D, q \rangle$

1. Construct a DFA $D'$ that is identical to $D$, except that $q$ is the only accept state
2. Run $S_2$ on $\langle D' \rangle$
3. If $S_2$ accepts, accepts; If $S_2$ rejects, rejects
Example

Correctness

- $\langle D, q \rangle \in UL_{DFA}$:
  Then, by definition, for all $w \in \Sigma^*$, $D$ on $w$ doesn’t reach $q$. Therefore, for all $w \in \Sigma^*$, $D'$ on $w$ doesn’t reach $q$, and thus $L(D') = \emptyset$.

- $\langle D, q \rangle \notin UL_{DFA}$ ...

Orit Moskovich and Gal Rotem (TAU)
Don’t forget the quiz... 

Oh, crap! Was that TODAY?

Good Luck!