Lecture 17: Mapping Reductions

Reading: Sipser Ch 5.3

Mark Bun
November 9, 2021
Reductions

A reduction from problem $A$ to problem $B$ is an algorithm for problem $A$ which uses an algorithm for problem $B$ as a subroutine.

If such a reduction exists, we say “$A$ reduces to $B$”

Positive uses: If $A$ reduces to $B$ and $B$ is decidable, then $A$ is also decidable.

Ex. $E_{DFA}$ is decidable $\Rightarrow E_{Q_{DFA}}$ is decidable

Negative uses: If $A$ reduces to $B$ and $A$ is undecidable, then $B$ is also undecidable.

Ex. $E_{TM}$ is undecidable $\Rightarrow E_{Q_{TM}}$ is undecidable
What’s wrong with the following “proof”?

Bogus “Theorem”: $A_{TM}$ is not Turing-recognizable

Bogus “Proof”: Let $R$ be an alleged recognizer for $A_{TM}$. We construct a recognizer $S$ for unrecognizable language $\overline{A_{TM}}$:

On input $\langle M, w \rangle$:
1. Run $R$ on input $\langle M, w \rangle$
2. If $R$ accepts, reject. Otherwise, accept.

This sure looks like a reduction from $\overline{A_{TM}}$ to $A_{TM}$
Mapping Reductions: Motivation

1. How do we formalize the notion of a reduction?
2. How do we use reductions to show that languages are unrecognizable?
3. How do we protect ourselves from accidentally “proving” bogus statements about recognizability?
Computable Functions

Definition:

A function $f: \Sigma^* \to \Sigma^*$ is **computable** if there is a TM $M$ which, given as input any $w \in \Sigma^*$, halts with only $f(w)$ on its tape. (“Outputs $f(w)$”)
Computable Functions

Definition:
A function $f : \Sigma^* \rightarrow \Sigma^*$ is **computable** if there is a TM $M$ which, given as input any $w \in \Sigma^*$, halts with only $f(w)$ on its tape. ("Outputs $f(w)$")

Example 1: $f(\langle x, y \rangle) = x + y$

Example 2: $f(\langle M, w \rangle) = \langle M' \rangle$ where $M$ is a TM, $w$ is a string, and $M'$ is a TM that ignores its input and simulates running $M$ on $w$
Mapping Reductions

Definition:

Language $A$ is mapping reducible to language $B$, written $A \leq_m B$, if there is a computable function $f: \Sigma^* \rightarrow \Sigma^*$ such that for all strings $w \in \Sigma^*$, we have $w \in A \iff f(w) \in B$. 
Mapping Reductions

Definition:
Language $A$ is mapping reducible to language $B$, written $A \leq_m B$ if there is a computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that for all strings $w \in \Sigma^*$, we have $w \in A \iff f(w) \in B$.

If $A \leq_m B$, which of the following is true?

a) $\overline{A} \leq_m B$

b) $A \leq_m \overline{B}$

c) $\overline{A} \leq_m \overline{B}$

d) $\overline{B} \leq_m \overline{A}$
Decidability

**Theorem:** If $A \leq_m B$ and $B$ is decidable, then $A$ is also decidable

**Proof:** Let $M$ be a decider for $B$ and let $f : \Sigma^* \to \Sigma^*$ be a mapping reduction from $A$ to $B$. Construct a decider for $A$ as follows:

On input $w$:

1. Compute $f(w)$
2. Run $M$ on input $f(w)$
3. If $M$ accepts, accept. If it rejects, reject.
Undecidability

**Theorem:** If $A \leq_m B$ and $B$ is decidable, then $A$ is also decidable

**Corollary:** If $A \leq_m B$ and $A$ is undecidable, then $B$ is also undecidable
Old Proof: Equality Testing for TMs

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

**Theorem:** \( EQ_{TM} \) is undecidable

**Proof:** Suppose for contradiction that there exists a decider \( R \) for \( EQ_{TM} \). We construct a decider for \( E_{TM} \) as follows:

On input \( \langle M \rangle \):

1. Construct TMs \( M_1, M_2 \) as follows:
   
   \( M_1 = M \)
   
   \( M_2 = \text{"On input } x, \text{ 1. Ignore } x \text{ and reject"} \)

2. Run \( R \) on input \( \langle M_1, M_2 \rangle \)

3. If \( R \) accepts, accept. Otherwise, reject.

This is a reduction from \( E_{TM} \) to \( EQ_{TM} \)
New Proof: Equality Testing for TMs

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

Theorem: \( E_{TM} \leq_m EQ_{TM} \) hence \( EQ_{TM} \) is undecidable

Proof: The following TM \( N \) computes the reduction \( f \):

On input \( \langle M \rangle \):

1. Construct TMs \( M_1, M_2 \) as follows:
   \[ M_1 = M \]
   \[ M_2 = \text{"On input } x, 1. \text{ Ignore } x \text{ and reject"} \]
2. Output \( \langle M_1, M_2 \rangle \)
Mapping Reductions: Recognizability

Theorem: If $A \leq_m B$ and $B$ is recognizable, then $A$ is also recognizable.

Proof: Let $M$ be a recognizer for $B$ and let $f: \Sigma^* \rightarrow \Sigma^*$ be a mapping reduction from $A$ to $B$. Construct a recognizer for $A$ as follows:

On input $w$:
1. Compute $f(w)$
2. Run $M$ on input $f(w)$
3. If $M$ accepts, accept. Otherwise, reject.
Unrecognizability

**Theorem:** If \( A \leq_m B \) and \( B \) is recognizable, then \( A \) is also recognizable

**Corollary:** If \( A \leq_m B \) and \( A \) is unrecognizable, then \( B \) is also unrecognizable

**Corollary:** If \( \overline{A_{TM}} \leq_m B \), then \( B \) is unrecognizable
Recognizability and $A_{TM}$

Let $L$ be a language. Which of the following is true?

a) If $L \leq_{m} A_{TM}$, then $L$ is recognizable
b) If $A_{TM} \leq_{m} L$, then $L$ is recognizable
c) If $L$ is recognizable, then $L \leq_{m} A_{TM}$
d) If $L$ is recognizable, then $A_{TM} \leq_{m} L$

**Theorem:** $L$ is recognizable if and only if $L \leq_{m} A_{TM}$
Recognizability and $A_{TM}$

**Theorem:** $L$ is recognizable if and only if $L \leq_m A_{TM}$

**Proof:**
Example: Another reduction to $EQ_{TM}$

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

Theorem: $A_{TM} \leq_m EQ_{TM}$

Proof: The following TM $N$ computes the reduction $f$:

What should the inputs and outputs to $f$ be?

a) $f$ should take as input a pair $\langle M_1, M_2 \rangle$ and output a pair $\langle M, w \rangle$
b) $f$ should take as input a pair $\langle M, w \rangle$ and output a pair $\langle M_1, M_2 \rangle$
c) $f$ should take as input a pair $\langle M_1, M_2 \rangle$ and either accept or reject
d) $f$ should take as input a pair $\langle M, w \rangle$ and either accept or reject
Example: Another reduction to $EQ_{TM}$

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

Theorem: $A_{TM} \leq_m EQ_{TM}$

Proof: The following TM computes the reduction:

On input $\langle M, w \rangle$:

1. Construct TMs $M_1, M_2$ as follows:
   $$M_1 = \text{“On input } x, \text{ } M_2 = \text{“On input } x,$$

2. Output $\langle M_1, M_2 \rangle$
Consequences of $A_{TM} \leq_m EQ_{TM}$

1. Since $A_{TM}$ is undecidable, $EQ_{TM}$ is also undecidable

2. $A_{TM} \leq_m EQ_{TM}$ implies $\overline{A_{TM}} \leq_m \overline{EQ_{TM}}$

Since $\overline{A_{TM}}$ is unrecognizable, $\overline{EQ_{TM}}$ is unrecognizable
\( \text{EQ}_{\text{TM}} \) itself is also unrecognizable

\[ \text{EQ}_{\text{TM}} = \{ \langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are TMs and } L(M_1) = L(M_2) \} \]

**Theorem:** \( \overline{A}_{\text{TM}} \leq_m \text{EQ}_{\text{TM}} \) hence \( \text{EQ}_{\text{TM}} \) is unrecognizable

**Proof:** The following TM computes the reduction:

On input \( \langle M, w \rangle \):

1. Construct TMs \( M_1, M_2 \) as follows:

   \( M_1 = \) “On input \( x \),
   1. Ignore \( x \)
   2. Run \( M \) on input \( w \)
   3. If \( M \) accepts, accept.
   Otherwise, reject.”

   \( M_2 = \) “On input \( x \),
   1. Ignore \( x \) and reject”

2. Output \( \langle M_1, M_2 \rangle \)