BU CS 332 – Theory of Computation

https://forms.gle/qHbdz9X5A4MyzTUs5

Lecture 18:

- Mapping Reductions

Reading:

Sipser Ch 5.3

HW 8 due Tuesday 11/22

Mark Bun

November 15, 2022

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 $\Rightarrow$ EQ_{DFA} is decidable

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

Ex. E_{TM} is undecidable $\Rightarrow$ EQ_{TM} is undecidable

Warning

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

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 A_{TM}:

Problem: Even if R recognizes A_{TM}, S doesn't necessarily recognize $\overline{A_{TM}}$.

TM S:

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

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

If $\langle M, w\rangle \in \overline{A_{TM}}$ where M loops on input w

$\Rightarrow$ R could loop on $\langle M, w\rangle$ $\Rightarrow$ S could loop on w $\Rightarrow$ $\langle M, w\rangle \notin L(S)$

This sure looks like a reduction from $\overline{A_{\mathrm{TM}}}$ 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?

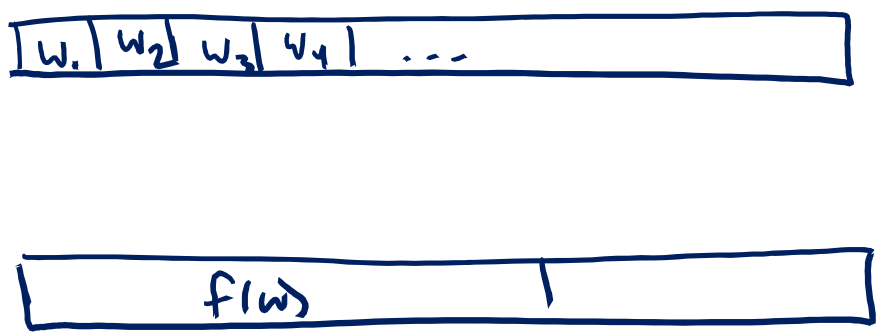
Computable Functions

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 $f(w)$”)

Input:

“Output”:



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 $f(w)$")

Example 1: $f(w) = \underline{\text{sort}}(w)$ HW5 Problem 3

is a computable function

Example 2: $f(\langle x, y \rangle) = x + y$

Input:

x_1	x_2	...	x_n	#	y_1	y_2	...	y_m	

Output:

z_1	z_2	z_3	...	z_k	

where $z = x + y$

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 $f(w)$")

Example 3: $f(\langle M, w \rangle) = \langle M' \rangle$ where M is a TM, w is a string, and M' is a TM that ignores its input and simulates running M on w

TM computing f:

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

1. Construct TM M':
   "On input x: Ignore x. Run M on w. If it accepts, accept, if rejects, reject"
2. Output $\langle M' \rangle$

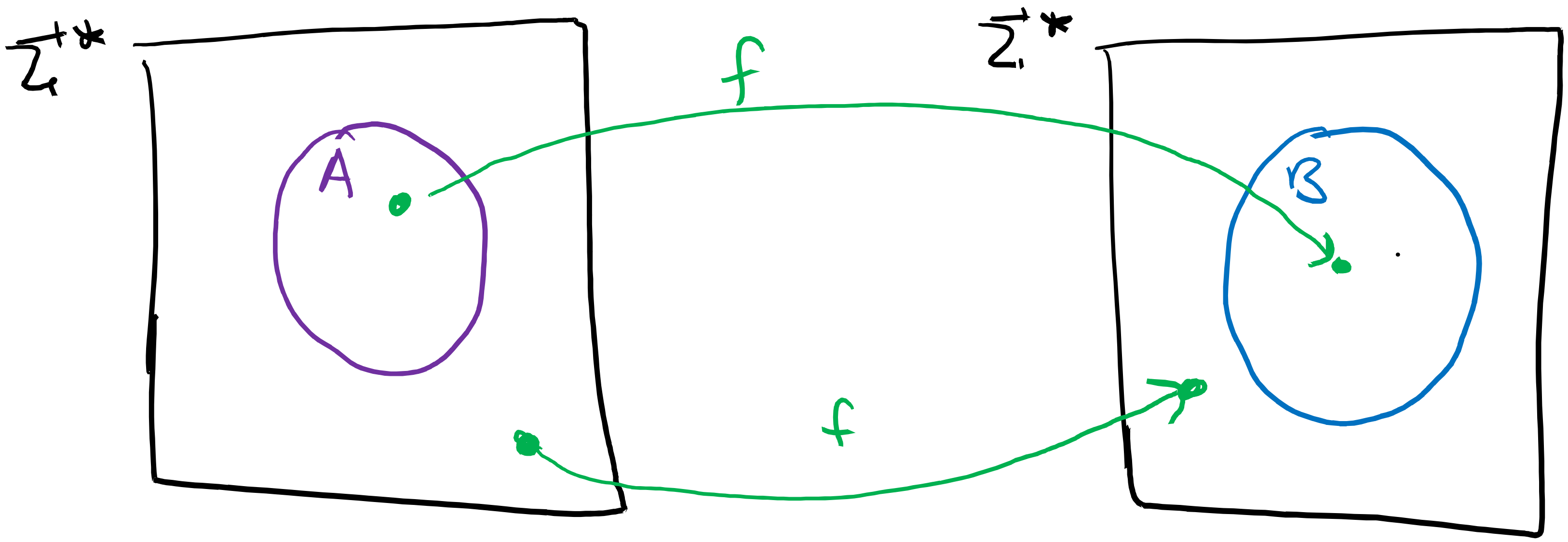
Mapping Reductions

Definition:

Let $A, B \subseteq \Sigma^*$ be languages. We say A is mapping reducible to B, written

$$A \leq_{\mathrm{m}} B$$

if there is a computable function $f\colon \Sigma^* \to \Sigma^*$ such that for all strings $w \in \Sigma^*$, we have $w \in A \Leftrightarrow f(w) \in B$



Mapping Reductions

Definition:

Language A is mapping reducible to language B, written

$$A \leq_{\mathrm{m}} B$$

if there is a computable function $f: \Sigma^* \rightarrow \Sigma^*$ such that for all strings $w \in \Sigma^*$, we have $w \in A \Leftrightarrow f(w) \in B$

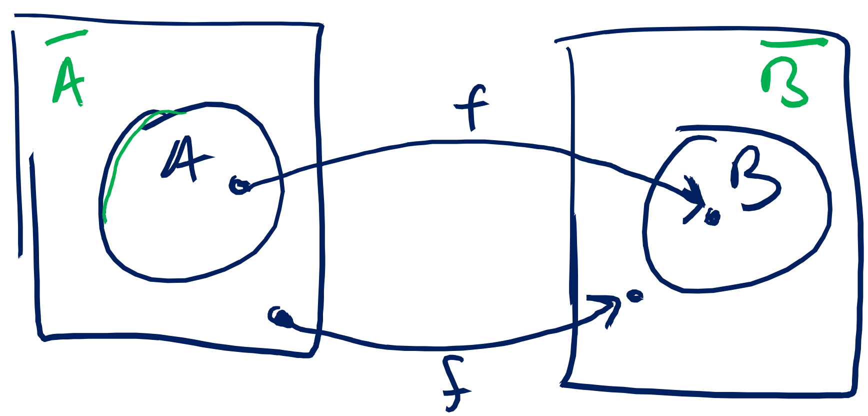
If $A \leq_{\mathrm{m}} B$, which of the following is true?

a) $\bar{A} \leq_{\mathrm{m}} B$

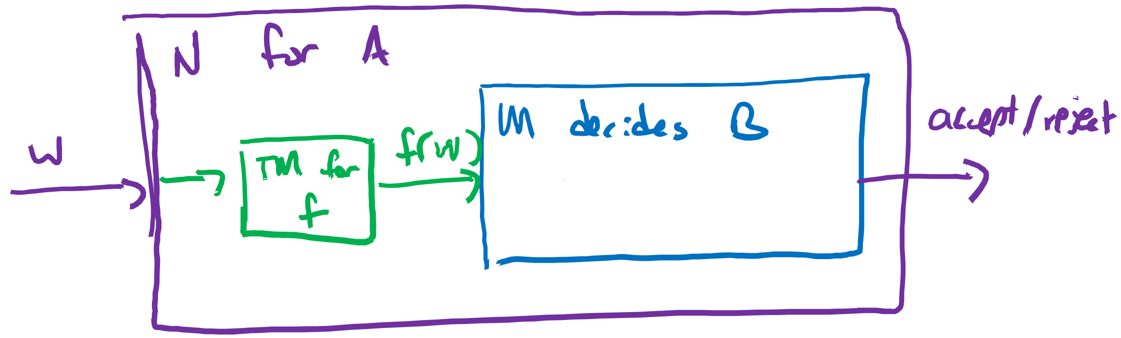
b) $A \leq_{\mathrm{m}} \bar{B}$

c) $\bar{A} \leq_{\mathrm{m}} \bar{B}$

d) $\bar{B} \leq_{\mathrm{m}} \bar{A}$



Decidability



Theorem: If $A \leq_{\mathrm{m}} B$ 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 N 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.

Correctness:

1) If $w \in A \Rightarrow f(w) \in B$ [by defn of mapping reduction]
$\Rightarrow$ M accepts f(w) [M decides B]
$\Rightarrow$ N accepts ✓

2) If $w \notin A \Rightarrow f(w) \notin B$ [by defn of mapping reduction]
$\Rightarrow$ M rejects f(w) [M decides B]
$\Rightarrow$ N rejects ✓

Undecidability

Theorem: If $A \leq_{\mathrm{m}} B$ and B is decidable, then A is also decidable

Corollary: If $A \leq_{\mathrm{m}} B$ and A is undecidable, then B is also undecidable

Contrapositive of Thm

Old Proof: Equality Testing for TMs

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

Theorem: EQ_{TM} is undecidable $E_{TM} = \{\langle M \rangle \mid M \text{ is a TM } L(M) = \emptyset\}$

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:

   $M_1 = M$

   $M_2 =$ "On input x,
   1. Ignore x 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

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

Theorem: $E_{\text{TM}} \leq_{\text{m}} EQ_{\text{TM}}$ hence EQ_{TM} is undecidable

Proof: The following TM N computes the reduction f:

If $\langle M\rangle \in E_{TM} \Rightarrow L(M) = \emptyset \Rightarrow L(M_1) = L(M_2) = \emptyset \Rightarrow \langle M_1, M_2\rangle \in EQ_{TM}$

If $\langle M\rangle \notin E_{TM} \Rightarrow L(M) \neq \emptyset \Rightarrow L(M_1) \neq L(M_2) = \emptyset \Rightarrow \langle M_1, M_2\rangle \notin EQ_{TM}$

On input $\langle M\rangle$:

1. Construct TMs M_1, M_2 as follows: $L(M_2) = \emptyset$

   $M_1 = M$ $L(M_1) = L(M)$

   M_2 = "On input x,
   1. Ignore x and reject"

2. Output $\langle M_1, M_2\rangle$

Claim: f is a mapping reduction from E_{TM} to EQ_{TM}

i.e. 1) $\langle M\rangle \in E_{TM} \Rightarrow f(\langle M\rangle) \in EQ_{TM}$

2) $\langle M\rangle \notin E_{TM} \Rightarrow f(\langle M\rangle) \notin EQ_{TM}$

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 $f: \Sigma^* \to \Sigma^*$ be a mapping reduction from A to B. Construct a recognizer N for A as follows:

Correctness:

1) If $w \in A \Rightarrow f(w) \in B$
$\Rightarrow M$ accepts $\Rightarrow N$ accepts ✓

2) If $w \notin A \Rightarrow f(w) \notin B$
$\Rightarrow M$ does not accept
$\Rightarrow N$ does not accept ✓

On input w:

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

Unrecognizability

Theorem: If $A \leq_{\mathrm{m}} B$ and B is recognizable, then A is also recognizable

Corollary: If $A \leq_{\mathrm{m}} B$ and A is unrecognizable, then B is also unrecognizable

Corollary: If $\overline{A_{\mathrm{TM}}} \leq_{\mathrm{m}} B$, then B is unrecognizable

Corollary: If $A_{TM} \leq_m \overline{B}$, then B is unrecognizable

Why?: $A_{TM} \leq_m \overline{B} \iff \overline{A_{TM}} \leq_m B$

Recognizability and A_{TM}

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

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

Know: $A_{TM} = \{\langle M, w\rangle \mid$ TM M accepts $w\}$ is recognizable

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

Recognizability and A_{TM}

Interpretation: A_{TM} is "the most expensive" recognizable language

A_{TM} is "complete" for the class of recognizable languages

under mapping reductions

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

Proof: $\Leftarrow$ | know A_{TM} is recognizable

By Thm about recognizability under mapping reductions,

$L \leq_m \underbrace{A_{TM}}_{\text{recognizable}} \Rightarrow L$ recognizable

---

$\Rightarrow$ | Let L be recognizable. Goal: Construct mapping reduction from L to A_{TM}.

Let M be a TM recognizing L.

The following TM computes a mapping reduction from L to A_{TM}:

On input w:
1) Output $\langle M, w \rangle$

Correctness:
1) If $w \in L \Rightarrow M$ accepts w
$\Rightarrow \langle M, w \rangle \in A_{TM}$
2) If $w \notin L \Rightarrow M$ does not accept w
$\Rightarrow \langle M, w \rangle \notin A_{TM}$

Example: Another reduction to EQ_{TM}

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

Theorem: $A_{\mathrm{TM}} \leq_{\mathrm{m}} EQ_{\mathrm{TM}}$ $A_{TM} = \{ \langle M, w\rangle \mid \text{TM } M \text{ accepts } w \}$

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}

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

Theorem: $A_{\text{TM}} \leq_{\text{m}} EQ_{\text{TM}}$

Proof: The following TM computes the reduction f:

If $\langle M, w\rangle \in A_{TM} \Rightarrow f(\langle M, w\rangle) = \langle M_1, M_2\rangle \in EQ_{TM}$

If $\langle M, w\rangle \notin A_{TM} \Rightarrow f(\langle M, w\rangle) = \langle M_1, M_2\rangle \notin EQ_{TM}$

$L(M_1) = \Sigma^*$

$L(M_2) = \begin{cases} \Sigma^* & \text{if } M \text{ accepts } w \\ \emptyset & \text{otherwise} \end{cases}$

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

1. Construct TMs M_1, M_2 as follows:

M_1 = "On input x,

Accept"

M_2 = "On input x,

Run M on w. If accepts, accept. If rejects, reject."

2. Output $\langle M_1, M_2\rangle$

Consequences of $A_{\mathrm{TM}} \leq_{\mathrm{m}} EQ_{\mathrm{TM}}$

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

2. $A_{\mathrm{TM}} \leq_{\mathrm{m}} EQ_{\mathrm{TM}}$ implies $\overline{A_{\mathrm{TM}}} \leq_{\mathrm{m}} \overline{EQ_{\mathrm{TM}}}$
   Since $\overline{A_{\mathrm{TM}}}$ is unrecognizable, $\overline{EQ_{\mathrm{TM}}}$ is unrecognizable

EQ_{TM} itself is also unrecognizable

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

Theorem: $\overline{A_{\mathrm{TM}}} \leq_{\mathrm{m}} EQ_{\mathrm{TM}}$ hence EQ_{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 $\langle M_1, M_2\rangle$