BU CS 332 – Theory of Computation

https://forms.gle/dNF9ECAsFxeJ48dp8

Lecture 17:

- Mapping Reductions

Reading:

Sipser Ch 5.3

I'll be around to take HW questions after class today

Mark Bun

April 3, 2024

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

False because Universal TM recognizes A_{TM}

Bogus "Proof": Let R be an alleged recognizer for A_{TM}. We construct a recognizer S for unrecognizable language A_{TM}:

S does not actually recognize $\overline{A_{TM}}$ even if R recognizes A_{TM}

TM S

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

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

Issue: Suppose $\langle M, w \rangle \in A_{TM}$ because M loops when run on input w

R run on input $\langle M, w \rangle$ might loop $\Rightarrow$ S run on input $\langle M, w \rangle$ might loop

But S was supposed to accept $\langle M, w \rangle$!

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

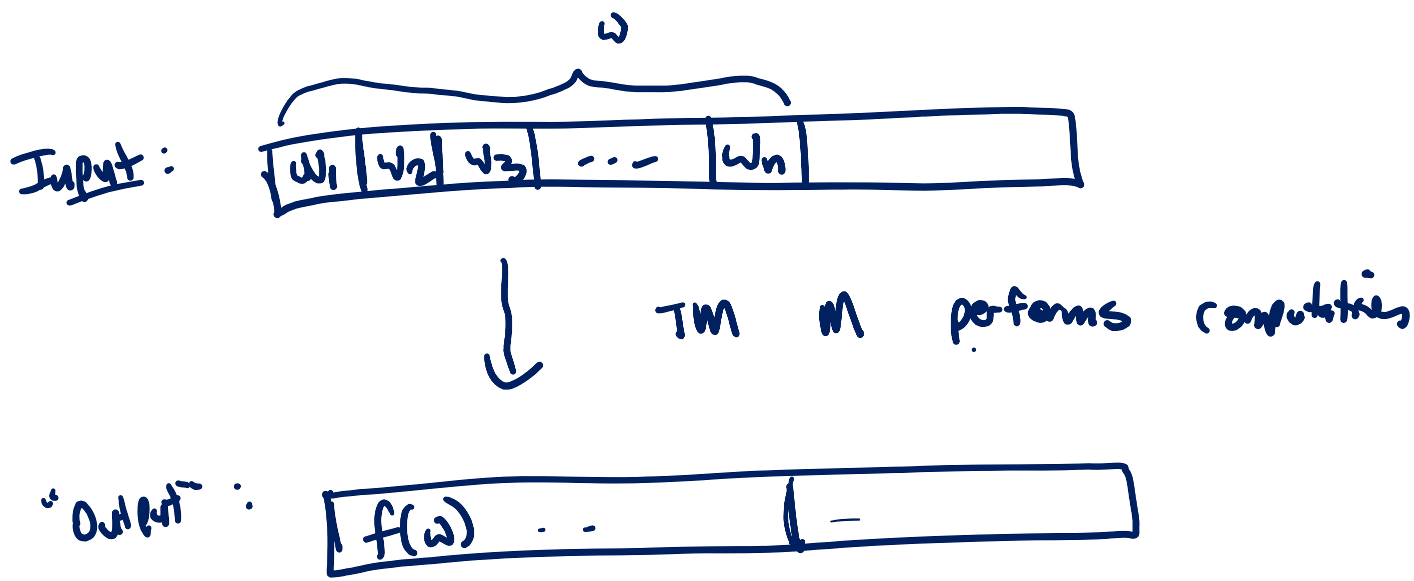
Mapping Reductions: Motivation

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

<u>Definition:</u>

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 $f(w)$")



Computable Functions

Why not computable?
If some TM N computes f, then the following TM decides A_{TM}:
1. Run N.
2. If 1, accept. If 0, reject.

The following function is not computable:

$$f(\langle M, w\rangle) = \begin{cases} 1 & \text{if TM } M \text{ accepts } w \\ 0 & \text{if TM } M \text{ does not accept } w \end{cases}$$

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 $f(w)$")

0	1	0	0	1	1	

$\Downarrow$

0	0	0	1	1	1	

Example 1: $f(w) = \text{sort}(w)$

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

Input

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

Output

z_1	z_2	...	z_s	

where $z = x + y$

Language $L \subseteq \Sigma^*$

$\Updownarrow$

Comp. problem: Given x, is $x \in L$?

$\Updownarrow$

Given x, evaluate $f(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{if } x \notin L \end{cases}$

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:
   1. Run M on w. If accepts, accept. If rejects, reject"
2. Output $\langle M' \rangle$.

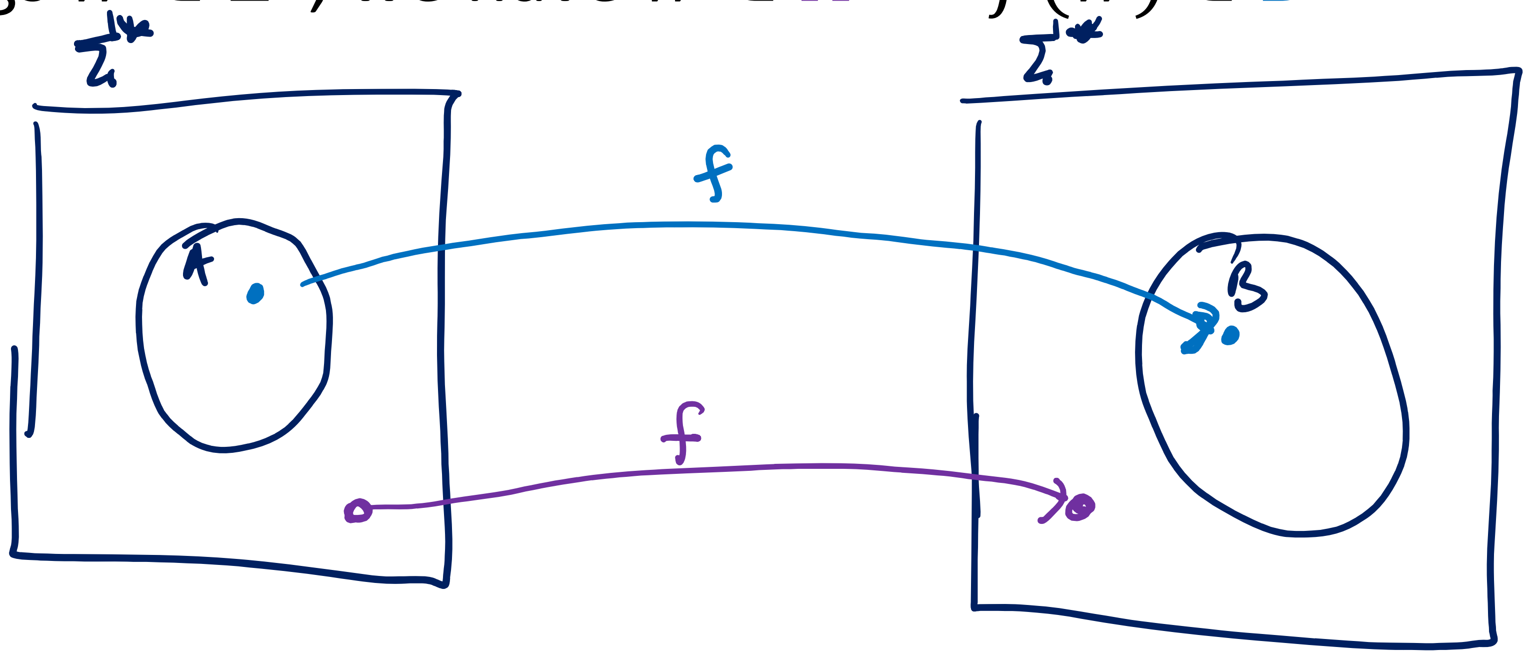
Mapping Reductions

Definition:

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

$$A \leq_m B$$

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

If f is a mapping reduction from A to B, it is also a mapping reduction from $\bar{A}$ to $\bar{B}$.

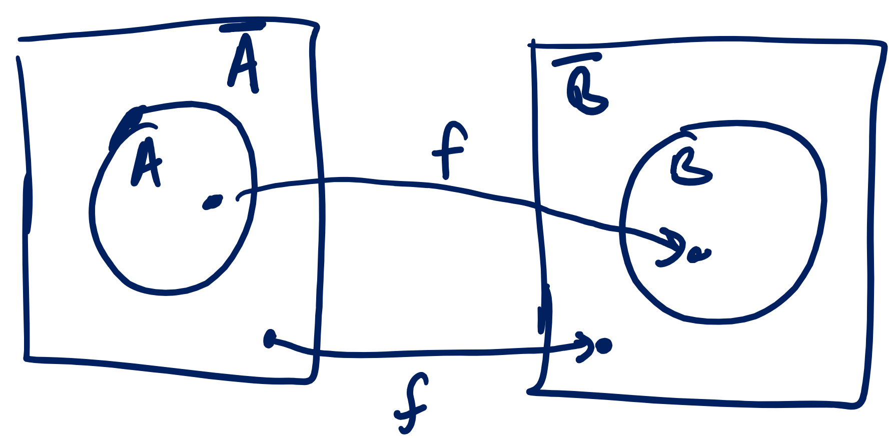
If $A \leq_{\mathrm{m}} B$, which of the following is true?

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. We can construct a decider N for A as follows:

TM N

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 $w \in A \Rightarrow f(w) \in B$ [defn. of mapping reduction]

   $\Rightarrow M$ accepts $f(w)$ [correctness of M]

   $\Rightarrow N$ accepts.

2) If $w \notin A \Rightarrow f(w) \notin B$ [defn. of mapping reduction]

   $\Rightarrow M$ rejects $f(w)$ [correctness of M]

   $\Rightarrow N$ rejects

$\Rightarrow N$ decides A

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

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

Theorem: EQ_{TM} is undecidable

$= \{ \langle M \rangle \mid M$ is a TM where $L(M) = \emptyset \}$

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:

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. Output $\langle M_1, M_2\rangle$

Correctness:

1) Let $\langle M\rangle \in E_{TM}$. Then $L(M) = \phi$
$\Rightarrow L(M_1) = \phi$, $L(M_2) = \phi$
$\Rightarrow \langle M_1, M_2\rangle \in EQ_{TM}$

2) Let $\langle M\rangle \notin E_{TM}$. Then $L(M) \neq \phi$
$\Rightarrow L(M_1) \neq \phi$, $L(M_2) = \phi$
$\Rightarrow \langle M_1, M_2\rangle \notin EQ_{TM}$.

Mapping Reductions: Recognizability

Theorem: If $A \leq_{\mathrm{m}} B$ 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 N for A as follows:

Correctness:

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

On input w:

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

   If it rejects, reject.

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

$\Rightarrow N$ recognizes.

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

Because $\overline{A_{TM}}$ is unrecognizable

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

Because $A \leq_m B \iff \overline{A} \leq_m \overline{B}$

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 $L \leq_{\mathrm{m}} A_{\mathrm{TM}}$ Also true
d) If L is recognizable, then $A_{\mathrm{TM}} \leq_{\mathrm{m}} L$

- $A \leq_m B$ and A unrecognizable $\Rightarrow$ B unrecognizable
- $A \leq_m B$ and B recognizable $\Rightarrow$ A recognizable
- A_{TM} is recognizable

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

Recognizability and A_{TM}

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

Proof: $\Leftarrow$] If $L \leq_m A_{TM}$, then since A_{TM} is recognizable, L is recognizable.

$\Rightarrow$] Let L be recognizable. Let $\underline{M}$ be a TM that recognizes L. Construct a mapping reduction f from $\underline{L}$ to $\underline{A_{TM}}$ computed by the following TM:

$w \in L \iff f(w) \in A_{TM}$

<u>TM N computing f:</u>

On input w:

Output $\langle M, w \rangle$

<u>Correctness:</u>

- If $w \in L \Rightarrow M$ accepts w $\Rightarrow \langle M, w \rangle \in A_{TM}$ ✓
- If $w \notin L \Rightarrow M$ does not accept w $\Rightarrow \langle M, w \rangle \notin A_{TM}$ ✓

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

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: $A_{\mathrm{TM}} \leq_{\mathrm{m}} EQ_{\mathrm{TM}}$

Proof: The following TM N computes the reduction f:

What should the inputs and outputs to f be?

instance of EQ_{TM}

instance of A_{TM}

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: $A_{\text{TM}} \leq_{\text{m}} EQ_{\text{TM}}$ $\Rightarrow$ $\overline{A_{TM}} \leq_m \overline{EQ_{TM}}$ $\Rightarrow$ $\overline{EQ_{TM}}$ is unrecognizable

Proof: The following TM computes the reduction f:

M accepts w $\iff$ $L(M_1) = L(M_2)$

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

1. Construct TMs M_1, M_2 as follows:

M_1 = "On input x,
Accept "

$L(M_1) = \Sigma^*$

M_2 = "On input x, (ignore x)
1. Run M on w.
2. If accepts, accept.
If rejects, reject.

$$L(M_2) = \begin{cases} \Sigma^* & \text{if } M \text{ accepts } w \\ \emptyset & \text{if } M \text{ does not accept } w \end{cases}$$

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 $\langle M_1, M_2\rangle$