BU CS 332 – Theory of Computation

Lecture 17:

- Mapping Reductions

Reading:

Sipser Ch 5.3

Mark Bun

March 21, 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 $\Rightarrow$ 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

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

What's wrong with the following "proof"?

Bogus "Theorem": A_{TM} is not Turing-recognizable

(false statement)

Bogus "Proof": Suppose for contradiction that there exists a recognizer R for A_{TM}. We construct a recognizer for $\overline{A_{\mathrm{TM}}}$:

not recognizable

TM S

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

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

Problem: S has wrong behavior when M loops on w

If M loops on w, S loops on $\langle M,w\rangle \Rightarrow$ $\langle M,w\rangle \notin L(S)$

But $\langle M,w\rangle \notin A_{\mathrm{TM}}$ $\Leftrightarrow \langle M,w\rangle \in \overline{A_{\mathrm{TM}}}$

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

$L(S) \neq \overline{A_{\mathrm{TM}}}$

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

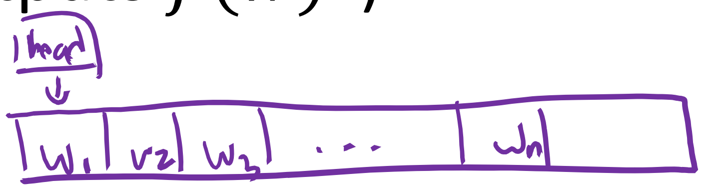
So far: Comput. devices solve decision (yes/no) problems

Definition:

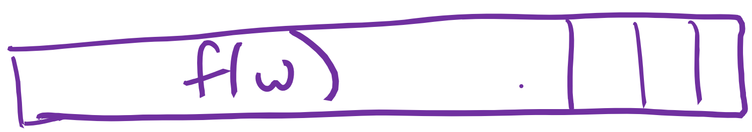
Now: TMs can compute functions

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 $f(w)$")

Start:



End:



Computable Functions

Definition:

Ex: There are uncomputable functions

$f_{A_{TM}}(\langle M,w\rangle) = \begin{cases} 1 & \text{if } M \text{ accepts } w \\ 0 & \text{o.w.} \end{cases}$

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 $f(w)$")

$\langle x, y\rangle = x\#y$

Start: $| x_1 | \dots | x_n | \# | y_1 | \dots | y_m |$

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

End: $| z_1 | z_2 | \dots | z_k |$ where $z = x + y$

$\Sigma = \{0, 1, \#\}$

$f(x\#y) = x+y$

$f(w) = \#$ if w not of form $x\#y$

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

M' = "On input x:
Ignore x, run M on w, output the result"

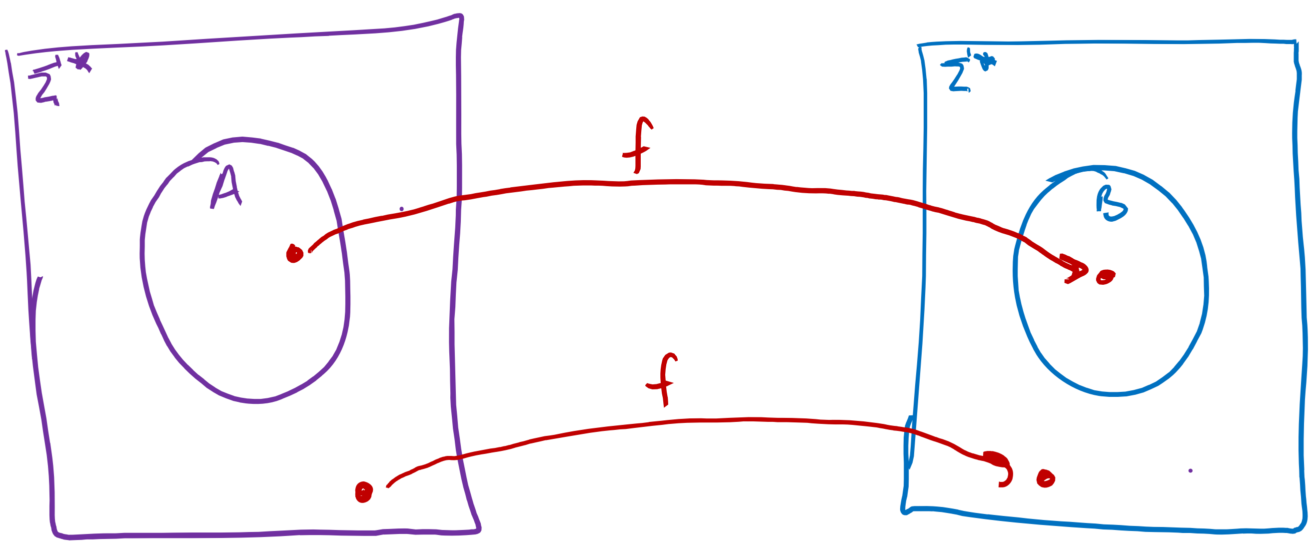
Mapping Reductions

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 $w \in A \Leftrightarrow f(w) \in B$



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 $w \in A \Leftrightarrow f(w) \in B$

Theorem: $A \leq_m B \Rightarrow \bar{A} \leq_m \bar{B}$.

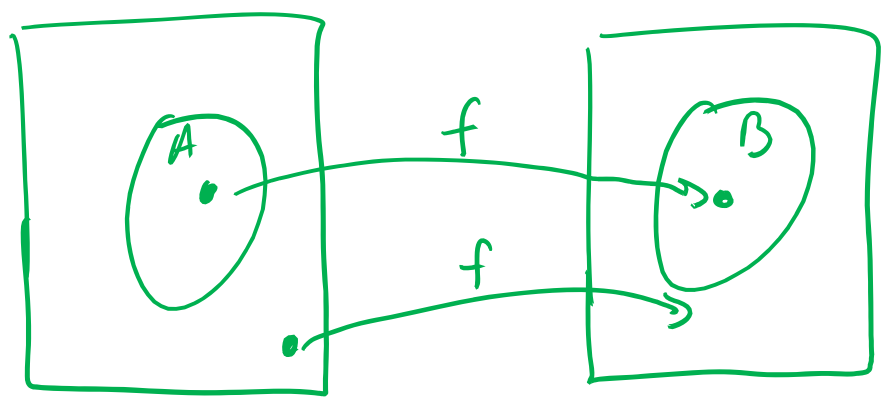
If $A \leq_{\mathrm{m}} B$, which of the following is always true?

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

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

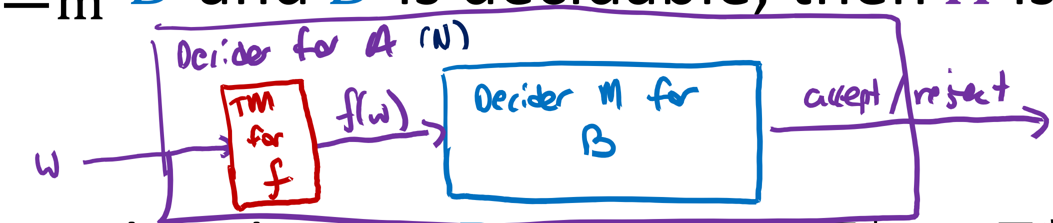
c) $\bar{A} \leq_{\mathrm{m}} \bar{B}$

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 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.

Proof of correctness:

1) If $w \in A$, then $f(w) \in B$ [mapping red.]
$\Rightarrow$ M accepts on input $f(w)$ [M decides B]
$\Rightarrow$ N accepts w

2) If $w \notin A$, then $f(w) \notin B$
$\Rightarrow$ M rejects on input $f(w)$
$\Rightarrow$ N rejects w

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

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

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}

$\langle M_1, M_2\rangle \in EQ_{TM} \iff M \in E_{TM}$

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

Using our prior knowledge that E_{TM} undec. to prove new statement that EQ_{TM} undec.

Proof: The following TM N computes the reduction f:

(Ex: A_{TM} informally reduces to E_{TM}, but $A_{TM} \not\leq_m E_{TM}$)

TM N

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 $\langle M_1, M_2\rangle$

Function f: $f(\langle M\rangle) = \langle M_1, M_2\rangle$

$\langle M\rangle$: instance of E_{TM}; $\langle M_1, M_2\rangle$: instance of EQ_{TM}

$L(M) = \emptyset \iff L(M_1) = L(M_2)$ i.e., $\langle M\rangle \in E_{TM} \iff \langle M_1, M_2\rangle \in EQ_{TM}$

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 for A as follows:

Same proof as for decidability.

TM N

On input w:

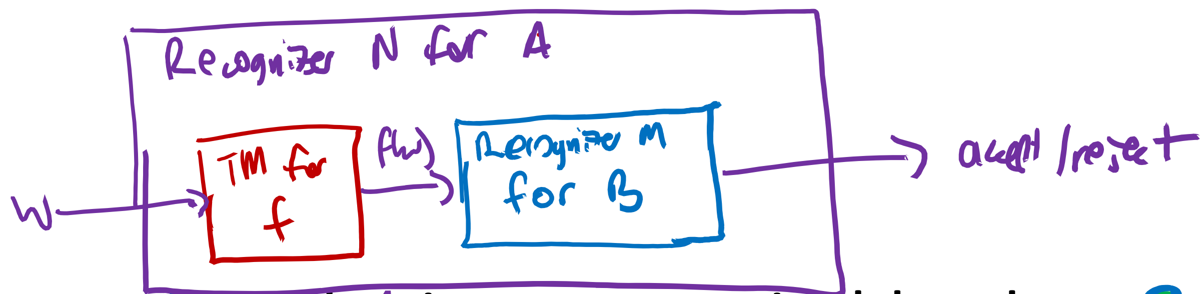
1. Compute $f(w)$
2. Run M on input $f(w)$
3. If M accepts, accept. Otherwise, reject.

1) If $w \in A$, then $f(w) \in B$ [mapping red.]
$\Rightarrow$ M accepts $f(w)$
$\Rightarrow$ N correctly accepts

2) If $w \notin A$, then $f(w) \notin B$
$\Rightarrow$ M either rejects or loops on $f(w)$
$\Rightarrow$ N correctly either rejects or loops

Unrecognizability

Theorem: If $\overbrace{A \leq_{\mathrm{m}} B}^{f}$ and $\overbrace{B \text{ is recognizable}}^{\text{TM } M}$, then $\overbrace{A \text{ is also}}^{\text{TM } N}$ 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

Recognizability and A_{TM}

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

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

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

Recognizability and A_{TM}

"L is no harder than A_{TM}"

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

Proof: ($\Leftarrow$) Follows from A_{TM} recognizable

"A_{TM} is the hardest recognizable language"

($\Rightarrow$) let L be recognizable by TM M. Goal: Construct mapping reduction f from L to A_{TM}. The following TM N computes f:

TM N

On input w: (instance of L)

Output $\langle M, w \rangle$

"A_{TM} is complete for RE = {recognizable languages} under mapping reductions"

Analysis: WTS that $w \in L \iff f(w) \in A_{TM}$

1) $w \in L \Rightarrow M$ accepts on input w [M recognizes L]
$\Rightarrow f(w) = \langle M, w \rangle \in A_{TM}$ [Def'n of N]

2) $w \notin L \Rightarrow M$ does not accept w
$\Rightarrow f(w) = \langle M, w \rangle \notin A_{TM}$