Mapping Reduction

CSE 105 Week 9 Discussion

Deadlines and Logistics

- Test 2 next week (week 10)
- Do review quizzes on PrairieLearn
- HW 6 due Thursday 3/13/25 at 5pm (week 10)
- Project due Wednesday 3/19/25 at 8am (final week) (NO EXTENSION)

Mapping reduction & computable functions

Definition: A is **mapping reducible to** B means there is a computable function $f : \Sigma^* \to \Sigma^*$ such that *for all* strings x in Σ^*,

$$x \in A \qquad \text{if and only if} \qquad f(x) \in B.$$

Notation: when A is mapping reducible to B, we write $A \leq_m B$.

Intuition: $A \leq_m B$ means A is no harder than B, i.e. that the level of difficulty of A is less than or equal the level of difficulty of B. **"Can convert questions about membership in A to questions about membership in B"**

Computable functions

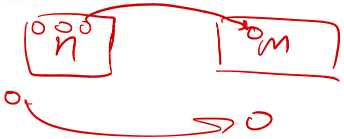
Definition: A function $f : \Sigma^* \to \Sigma^*$ is a **computable function** means there is some Turing machine such that, for each x, on input x the Turing machine halts with exactly $f(x)$ followed by all blanks on the tape

Warm up: If A is mapping reducible to B then the complement of A is mapping reducible to the complement of B.

Theorems 5.22, 5.28: If A is mapping reducible to B…
… and B is decidable, then A is decidable.
… and A is undecidable, then B is undecidable.
… and B is recognizable, then A is recognizable.
… and A is unrecognizable, then B is unrecognizable.

Recall that **mapping reduction** is defined in section 5.3: For languages A and B over Σ, we say that the problem A mapping reduces to B means there is a computable function $f : \Sigma^* \to \Sigma^*$ such that for all $x \in \Sigma^*$, $x \in A$ iff $f(x) \in B$. A computable function that makes the iff true is said to witness the mapping reduction from A to B.

$n, m > 0$



Fix $\Sigma = \{0, 1\}$ throughout this question.

Is each of the stated mapping reductions witnessed by the given function?

$g(0) = 00$ $g(1) = 10$

$\boxed{\{0,1\}} \leq_m \{00, 10\}$ is witnessed by the computable function $g : \Sigma^* \to \Sigma^*$ given by $g(x) = x0$ for all x

$\{00, 10\} \leq_m \{0, 1\}$ is witnessed by the computable function $f : \Sigma^* \to \Sigma^*$ given by

$$f = \begin{cases} 0 & \text{if } x = 00 \\ 0 & \text{if } x = 10 \\ 00 & \text{otherwise} \end{cases}$$

$f(00) = 0$

$f(10) = 0$

$$f(x) = \begin{cases} 0 & \text{if } x = y0 \text{ for some } y \in \{0,1\} \\ 00 & \text{otherwise} \end{cases}$$

$f(110) = 0$

$\{0, 1\} \leq_m \{00, 10\}$ is witnessed by the computable function $f : \Sigma^* \to \Sigma^*$ given by

$f(0) = 0$

$$f(x) = \begin{cases} 0 & \text{if } x = y0 \text{ for some } y \in \{0,1\} \\ 00 & \text{otherwise} \end{cases}$$

1. The mapping reduction relationship is not true.
2. The mapping reduction relationship is true but the given function does not witness this mapping reduction.
3. This mapping reduction is witnessed by this computable function.

Mapping reduction practice

RQ8.10. Properties of mapping reductions

Recall that **mapping reduction** is defined in section 5.3: For languages A and B over Σ, we say that the problem A mapping reduces to B means there is a computable function $f : \Sigma^* \to \Sigma^*$ such that for all $x \in \Sigma^*$, $x \in A$ iff $f(x) \in B$. A computable function that makes the iff true is said to witness the mapping reduction from A to B.

Select all and only the true statements below. $\phi \leq_m \{1\}$ $\quad$ $\{1\} \leq_m \phi$ $\times$

- [ ] For all languages A and B, if A mapping reduces to B then B mapping reduces to A. $\leftarrow$
- [ ] Every language mapping reduces to its complement. $\phi \leq_m \Sigma^*$ $\times$
- [ ] Σ^* mapping reduces to every nonempty language over Σ. a
- [ ] Every decidable language mapping reduces to $\emptyset$.
- [ ] $\emptyset$ mapping reduces to every nonempty language over Σ.
- [ ] For all languages A and B and C, if A mapping reduces to B and B mapping reduces to C then A mapping reduces to C.

Proof: $\leq_m$ is transitive

Suppose $A \leq_m B$, $B \leq_m C$.

$\exists$ f: $x \in A \iff f(x) \in B$. — M_f

g: $x \in B \iff g(x) \in C$. — M_g

M_h:

simulate M_f on x. output is y.

Simulate M_g on y. output z.

$h = g(f(x))$

$x \in A \iff f(x) \in B \iff g(f(x)) \in C$

h. $x \in A \iff h(x) \in C$.

WTS $A \leq_m C$.

Sipser Exercise 5.6

[A]**5.7** Show that if A is Turing-recognizable and $A \leq_m \overline{A}$, then A is decidable.

$A \leq_m \overline{A}$

$\exists f$ $\begin{cases} x \in A & f(x) \in \overline{A} \\ \underline{x \in \overline{A}} & \underline{f(x) \in A} \end{cases}$

$\overline{A} \leq_m \underline{A} \Rightarrow \overline{A}$ recognizable

A decidable

Halting Problem

Prove : $A_{TM} \leq_m HALT_{TM}$

$\langle M, w\rangle$ acc $\Rightarrow$ $\langle M', w'\rangle$ halt

$\langle M, w\rangle$ rej/loop $\Rightarrow$ bad format

$\langle M', w'\rangle$ loops

bad $\Rightarrow$ bad

M': on input y:

run M on y, if acc $\Rightarrow$ acc

rej $\Rightarrow$ loop.

$HALT_{TM} \leq_m A_{TM}$

M': on y:

run M on w

$\langle M', w'\rangle$

$\langle M', \langle M'\rangle\rangle$

$\langle M, w\rangle$ halt $\rightarrow$ $\langle M', w'\rangle$ acc

loop $\rightarrow$ $\langle M', w'\rangle$ rej/loop

f:

M': run M on w.

$\langle M', w\rangle$

Language Emptiness Problem

E_{TM} co-rec: $E_{TM} \leq_m \overline{A_{TM}}$

undecidable

unrec.

Prove : $A_{TM} \leq_m \overline{E_{TM}}$ $\overline{A_{TM}} \leq_m E_{TM}$

$\langle M, w \rangle$ acc $\Rightarrow$ $\langle M' \rangle$ nonempty

rej/loop $\Rightarrow$ empty

M': on input y
run M on w.

acc $\to$ acc y
rej $\to$ reject y

$\langle M' \rangle$

Language Equality Problem

EQ_{TM} not co-rec

Prove : $HALT_{TM} \leq_m EQ_{TM}$ $\{1\}$ Σ^*

$\langle M, w\rangle$ halt $\Rightarrow$ $\langle M_1, M_2\rangle$ equal

$\langle M, w\rangle$ loop $\Rightarrow$ $\langle M_1, M_2\rangle$ unequal

M_1: on input y: $\leftarrow$ if $y \neq 1$, rej.

Run M on w.

if acc/rej, accept y.

halt $\Rightarrow$ $L(M_1) = \boxed{\Sigma^*}$ $\{1\}$

loop $\Rightarrow$ $L(M_1) = \phi$

$\langle M_1, M_2\rangle$

EQ_{TM} not rec

$\overline{EQ_{TM}}$ not co-reco

$HALT_{TM} \leq_m \overline{EQ_{TM}}$ ϕ

$\langle M, w\rangle$ halt $\Rightarrow$ $\langle M_1, M_2\rangle$ unequal

$\langle M, w\rangle$ loop $\Rightarrow$ $\langle M_1, M_2\rangle$ equal

Summary

Computable Problems	Recognizable	Co-recognizable	Decidable
A_{TM}	✅	❌	❌
$\overline{A_{TM}}$	❌	✅	❌
$HALT_{TM}$	✅	❌	❌
$\overline{HALT_{TM}}$	❌	✅	❌
E_{TM}	❌	✅	❌
$\overline{E_{TM}}$	✅	❌	❌
EQ_{TM}	❌	❌	❌
$\overline{EQ_{TM}}$	❌	❌	❌

Equally Expressive Models

- Deterministic Turing Machines
- May-stay Machines (Head can move L, R, Stay)
- Multitape Turing Machines
- Enumerators
- Nondeterministic Turing Machines

Bonus!

Rice's theorem. Let P be any nontrivial property of the language of a Turing machine. Prove that the problem of determining whether a given Turing machine's language has property P is undecidable.

In more formal terms, let P be a language consisting of Turing machine descriptions where P fulfills two conditions. First, P is nontrivial—it contains some, but not all, TM descriptions. Second, P is a property of the TM's language—whenever $L(M_1) = L(M_2)$, we have $\langle M_1 \rangle \in P$ iff $\langle M_2 \rangle \in P$. Here, M_1 and M_2 are any TMs. Prove that P is an undecidable language.

Sipser Exercise 5.28

Rice's Theorem

Assume for the sake of contradiction that P is a decidable language satisfying the properties and let R_P be a TM that decides P. We show how to decide A_{TM} using R_P by constructing TM S. First, let $T_\emptyset$ be a TM that always rejects, so $L(T_\emptyset) = \emptyset$. You may assume that $\langle T_\emptyset \rangle \notin P$ without loss of generality because you could proceed with $\overline{P}$ instead of P if $\langle T_\emptyset \rangle \in P$. Because P is not trivial, there exists a TM T with $\langle T \rangle \in P$. Design S to decide A_{TM} using R_P's ability to distinguish between $T_\emptyset$ and T.

$S =$ "On input $\langle M, w \rangle$:

1. Use M and w to construct the following TM M_w.
   $M_w =$ "On input x:
   1. Simulate M on w. If it halts and rejects, *reject*. If it accepts, proceed to stage 2.
   2. Simulate T on x. If it accepts, *accept*."
2. Use TM R_P to determine whether $\langle M_w \rangle \in P$. If YES, *accept*. If NO, *reject*."

TM M_w simulates T if M accepts w. Hence $L(M_w)$ equals $L(T)$ if M accepts w and $\emptyset$ otherwise. Therefore, $\langle M_w \rangle \in P$ iff M accepts w.

if $\exists$ $\underline{P}$ decidable

$\Rightarrow$ A_{TM} decidable

$\exists \langle M_\emptyset \rangle \in P$

$\exists \langle M \rangle \in \overline{P}$

$\langle M, w \rangle \in A_{TM}$

$\notin A_{TM}$