DISCUSSION WEEK 9 CSE 105 SPRING '23

Agenda: 5.4, 5.5, 5.6, 5.7, 5.10, 5.16 (Sipser)

(5.4) If $A \leq_m B$ & B is regular, does that imply A is regular? Why / why not?

No. Suppose $A = \{0^n 1^n \mid n \geq 0\}$, $B = \{0, 1\}$.
Consider the following reduction:

f = "On input x

1. If x is of the form $0^n 1^n$, output 0
2. Else output 00"

F defines a computable function, & for any x, $f(x) \in B$ iff $x \in A$. But A is not regular.

(5.5) Show that A_{TM} is not mapping reducible to E_{TM}; i.e. no computable function reduces A_{TM} to E_{TM}.

Assume, towards contradiction, that $A_{TM} \leq E_{TM}$ via some reduction f.

We already know

(a) E_{TM} is co-recognizable ($\overline{E_{TM}}$ is recognizable)

(b) A_{TM} is recognizable but not decidable (since $\overline{A_{TM}}$ is not recognizable)

By the definition of mapping reducibility, we also have

(c) $\overline{A_{TM}} \leq \overline{E_{TM}}$ via the same reduction f.

From (c) & (a), we get $\overline{A_{TM}}$ is recognizable (Th. 5.28 Sipser)

But this contradicts (b): If A_{TM} is recognizable & $\overline{A_{TM}}$ is recognizable, A_{TM} is decidable (which we know is not true).

$\therefore$ A_{TM} is not mapping reducible to E_{TM}.

(5.6) Show that $\leq_m$ is a transitive relation.

Suppose there exist A, B, C such that

$A \leq_m B$ via computable function f &

$B \leq_m C$ via computable function g.

W.T.S $\exists$ some computable function h such that

$A \leq_m C$ via h.

Build a T.M that computes H as follows:

H = "On input x

1. Simulate a T.M for f on input x. Call the output y ($y = f(x)$)
2. Simulate a T.M for g on input y.
3. Output $g(y)$ (i.e $g(f(x))$)"

Clearly $h(x) = g(f(x))$, & h is a computable fn.

Also, $x \in A \leftrightarrow f(x) \in B$

$f(x) \in B \leftrightarrow g(f(x)) \in C$

$\therefore x \in A \leftrightarrow g(f(x)) \in C$ i.e $h(x) \in C$

$\therefore A \leq_m C$ via h.

(5.7) S.T if A is Turing-recognizable & $A \leq_m \bar{A}$, then A is decidable.

Suppose $A \leq_m \bar{A}$ via computable function f.

Then, $\bar{A} \leq_m A$ via the same f.

A is recognizable, so by Thm 5.28, $\bar{A}$ is also recognizable.

A is recognizable & $\bar{A}$ is recognizable, so A is decidable.

(5.10) Consider the problem of determining whether a 2-tape T.M ever writes a non-blank symbol on its second tape when run on input w. Formulate this problem as a language & show that it is undecidable.

Let $B = \{ \langle M, w \rangle \mid M$ is a 2-tape T.M that writes a non-blank symbol on its second tape when it is run on input $w \}$

To show B is undecidable, we demonstrate a mapping reduction from A_{TM} to B.

Want a computable function f such that on input x:

(a) If x is of the form $\langle M, w \rangle$ where M is a T.M that accepts w, then $f(x) = \langle M', w' \rangle$ such that $\langle M', w' \rangle \in B$.

(b) If x is of the form $\langle M, w \rangle$ & M is a T.M that does not accept w, OR if x is not of the form $\langle M, w \rangle$, then $f(x) = \langle M', w' \rangle$ such that $\langle M', w' \rangle \notin B$.

Construct T.M F that computes f as follows:

$F =$ "On input x

1. Type check whether $x = \langle M, w \rangle$ for some T.M M & string w. If so, go to step 2. If not, construct a 2-tape T.M M_x as follows:

   $M_x =$ " On input y

   1. Do nothing".

   Output $\langle M_x, \epsilon \rangle$.

2. Construct the 2-tape machine M_x' as follows:

M_x' = " On input y

1. Ignore y.
2. Simulate M on input w using the first tape.
3. If M accepts, write a non-blank symbol onto the second tape & accept."

Output $\langle M_x', \epsilon \rangle$."

Observe that if $x = \langle M, w \rangle \in A_{TM}$, then $f(x)$ is $\langle M_x', \epsilon \rangle$, and M_x' writes a non-blank symbol on its second tape on input ϵ, because M accepts w; $\therefore f(x) \in B$.

However, if $x = \langle M, w \rangle \notin A_{TM}$ then M either loops on or rejects w, in which case $f(x) = \langle M_x', \epsilon \rangle$ where M_x' does not write a non-blank symbol on its second tape, so $f(x) \notin B$.

When x doesn't type check, $f(x) = \langle M_x, \epsilon \rangle \notin B$ because M_x does nothing & so does not write a non-blank symbol on its second tape.

$\therefore A_{TM} \leq_m B$ via f. Using corollary 5.23, since A_{TM} is undecidable, B is undecidable.

(5.16) Let $\Gamma = \{0, 1, \sqcup\}$ be the tape alphabet for all TMs in this problem. Define the busy beaver fn $BB: \mathcal{N} \to \mathcal{N}$ as follows:

For each $k \in \mathcal{N}$, consider all k-state TMs that halt when started with a blank tape.

$BB(k)$ is the max. # 1s that remain on the tape among all such T.Ms.

Show that BB is not computable.

Idea: Proof by contradiction. If BB is computable then A_{TM} is decidable. Suppose that some T.M F computes BB.

Build a decider D for A_{TM} as follows:

D = " On input $\langle M, w \rangle$

1. Construct T.M M_w as follows:

   M_w = " On input x

   1. Ignore x.
   2. Simulate M on w, keep a count (c) of the # steps used in the simulation
   3. If & when M halts, write c 1s on the tape, & halt."

2. Use F to compute $BB(k)$, $k = \#$ states in M_w.

3. Run M on w for $BB(k)$ steps.

4. If M accepts, accept, else reject."

Proof that D decides A_{TM}:

(1) Suppose input $\langle M, w \rangle \in A_{TM}$:

(a) This means M accepts w in some # steps (say 'c').

(b) So, M_w halts when started with a blank tape, & writes c 1s on its tape.

(c) $BB(k) \geq c$ by definition, since $k = \#$ states of M_w.

(d) D runs M on w for $BB(k)$ steps, & M accepts w in c $(\leq BB(k))$ steps, $\therefore$ D sees this & accepts $\langle M, w \rangle$.

(2) Suppose input $\langle M, w \rangle \notin A_{TM}$

(a) Irrespective of $BB(k)$, there is no c such that M accepts w in c steps.

(b) So, D rejects $\langle M, w \rangle$.

$\therefore$ D decides A_{TM}, which is a contradiction.

$\therefore$ BB is not computable.
