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

No. Suppose \( A = \{0^n1^n \mid n \geq 0\} \), \( B = \{0,1\} \).
Consider the following reduction:

\[
\text{If } x \text{ is of the form } 0^n1^n, \text{ output } 0 \\
\text{Else output } 00
\]

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

(5.5) Show that \( \text{Aim} \) is not mapping reducible to \( \text{Eim} \), i.e. no computable function reduces \( \text{Aim} \) to \( \text{Eim} \).

Assume, towards contradiction, that \( \text{Aim} \leq \text{Eim} \).

Via some reduction \( f \).

We already know

(a) \( \text{Eim} \) is co-recognizable (\( \overline{\text{Eim}} \) is recognizable)

(b) \( \text{Aim} \) is recognizable but not decidable.

(since \( \overline{\text{Aim}} \) is not recognizable)
By the definition of mapping reducibility, we also have
\((c) \overline{A_{TM}} \leq_{E_{TM}} \overline{F_{TM}}\) via the same reduction \(f\).

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

But this contradicts \((b)\): If \(A_{TM}\) is recognizable and \(\overline{A_{TM}}\) 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 \(f\) and
\(B \leq_{m} C\) via computable function \(g\).

W.l.o.g. \(\exists\) some computable function \(h\) such that
\(A \leq_{m} C\) via \(h\).

Build a T.M that computes \(H\) as follows:
\[H = \text{"On input } x\]
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, \( x \in A \iff f(x) \in B \)
\( f(x) \in B \iff g(f(x)) \in C \)
\( \therefore x \in A \iff g(f(x)) \in C \) i.e. \( h(x) \in C \)
\( A \leq_m C \) via \( h \).

(5.9) S.T. if \( A \) is Turing-recognizable & \( A \leq_m \overline{A} \), then \( A \) is decidable.

Suppose \( A \leq_m \overline{A} \) via computable function \( f \).
Then, \( \overline{A} \leq_m A \) via the same \( f \).

\( A \) is recognizable, so by Thm 5.28, \( \overline{A} \) is also recognizable.

\( A \) is recognizable & \( \overline{A} \) 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 \( B = \{ \langle M, w \rangle \mid M \text{ is a 2-tape T.M that writes a non-blank symbol on its second tape when it is run on input } w \} \).

To show \( B \) is undecidable, we demonstrate a mapping reduction from \( \text{Afm} \) 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 \( \langle m', w' \rangle \in B \).

Construct T.M \( F \) that computes \( f \) as follows:

\[ F = \text{"On input } x \text{"} \]

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:

   \[ M_x = \text{"On input } y \text{"} \]

   1. Do nothing.

Output \( \langle M_x, \epsilon \rangle \).
2. Construct the two-tape machine \( M_x' \) as follows:

\[ M_x' = \langle \text{On input } y \rangle \]

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 and accept.

Output \( \langle m_x', e \rangle \).

Observe that if \( x = \langle M_1, w \rangle \in \text{Arm} \), then \( f(x) \) is \( \langle m_x', e \rangle \), and \( M_x' \) writes a non-blank symbol on its second tape on input \( e \), because \( M \) accepts \( w \); \( f(x) \in B \).

However, if \( x = \langle M_1, w \rangle \in \text{Arm} \) then \( M \) either loops on \( w \) or rejects \( w \), in which case \( f(x) = \langle m_x', e \rangle \) 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, \( f(x) = \langle m_x, e \rangle \notin B \) because \( M_x \) does nothing and so does not write a non-blank symbol on its second tape.
Arm ≤n B via f. Using corollary 5.23, since Arm is undecidable, B is undecidable.

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

For each k ∈ 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 TMs.

Show that BB is not computable.

Idea: Proof by contradiction. If BB is computable then Arm is decidable. Suppose that some Tm F computes BB.

Build a decider D for Arm as follows:

D = "on input \langle M, w \rangle"

1. Construct Tm Mw as follows:

   Mw = "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 $BB(k)$, $k = \# \text{states in } M_w$.
3. Run $M$ on $w$ for $BB(k)$ steps.
4. If $M$ accepts, accept; else reject.

Proof that $D$ decides $Atm$:

1) Suppose input $\langle m, w \rangle \in Atm$:
   (a) This means $M$ accepts $w$ in some $\#$ steps (say $c$).
   (b) So, $M_w$ halts when started with a blank tape, & writes $c$ 1's on its tape.
   (c) $BB(k) \geq c$ by definition, since $k = \# \text{states of } M_w$.
   (d) $D$ runs $M$ on $w$ for $BB(k)$ steps, & $M$ accepts $w$ in $c (\leq BB(k))$ steps, \implies $D$ sees this & accepts $\langle m, w \rangle$.

2) Suppose input $\langle m, w \rangle \notin Atm$
   (a) Irrespective of $BB(k)$, there is no $c$ such that $M$ accepts $w$ in $c$ steps.
   (b) So, $D$ rejects $\langle m, w \rangle$.

\implies $D$ decides $Atm$, which is a contradiction.
\implies BB is not computable.