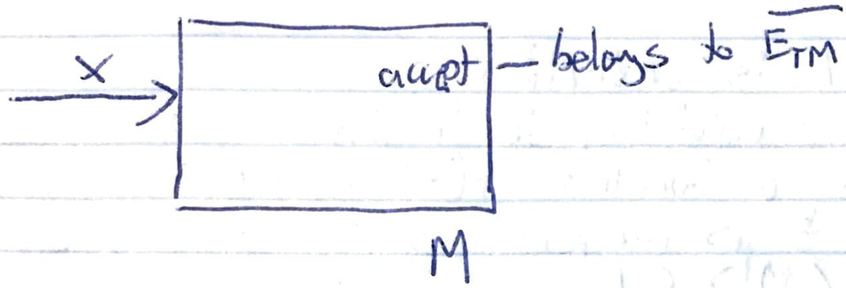
1. Let x be a string that is run on Turing Machine M. x is any string.



- If M accepts x this means that $L(M) \neq \emptyset$ and is part of $\overline{E_{TM}}$
- Since we know that $\langle M \rangle \in \overline{E_{TM}}$ if it accepts any string we know that $\overline{E_{TM}}$ is recognizable.
- E_{TM} is also reccognizable since if any string is input it will be either rejected or accepted. If the list of strings is infinite it will stay in a loop state, but that still means it is recognizable.

$\therefore$ Therefore E_{TM} and $\overline{E_{TM}}$ are co-recognizable.

2. $L = \{ \langle M \rangle \mid$ M when started on the blank tape, eventually writes a \$ somewhere on the tape $\}$

- Prove L is undecidable
- Use undecidability to do this of A_{TM}
  - ↳ Give a function $f(\langle M, w \rangle) = \langle M' \rangle$ for some TM M',
- Prove f is computable so that $\langle M, w \rangle \in A_{TM}$ iff $\langle M' \rangle \in L$

- We assume L is decidable
- Run input $\langle M, w \rangle$ on Turing machine M'

  1. If w is accepted as a string then M' accepts, else it rejects
  2. If M' is accepted then feed that to M, if M' accepts then M accepts, otherwise it rejects.

- This creates a decider for A_{TM} which is a contradiction. Therefore we have proven that by contradiction L is undecidable.

3. $L = \{\langle M_1, M_2\rangle \mid L(M_1) \cap L(M_2) = \emptyset\}$

- Prove L is not recognizable
- Use E_{TM}.

- We assume that L is recognizable
- Let M_1' and M_2' be a turing machine(s) that recognizes input $\langle M_1, w\rangle$ and $\langle M_2, w\rangle$

↳ If w is a string and is accepted by M' that means $L(M_1') \neq \emptyset$, if it doesn't accept then reject

↳ If M_1' rejects then M_1 accepts

↳ Then run $\langle M_2, w\rangle$ on M_2', if w is accepted by M_2' that means $L(M_2') \neq 0$ else wise it rejects.

↳ If M_2' rejects then M_2 accepts

↳ Thus if both M_1 and M_2 accept L is decidable

↳ However L can't be decidable since by definition E_{TM} isn't decidable.

↳ If L isn't decidable is is proven that it isn't recognizable.

4. Prove that the language is undecidable.

$L = \{ \langle M \rangle \mid M$ is a TM that accepts w^R whenever it accepts $w \}$

- w^R denotes the reverse of the string w.
- Assume we have $w_1 w_2 = w$ and $w_2 w_1 = w^R$
- Let us construct TM M'
  - ↳ We run M' on $w_2 w_1$
    - if M' accepts $w_2 w_1$ then feed that to M and M will ~~reject~~ ~~accept~~ reject
    - if M' rejects $w_2 w_1$ then feed that to M and M will ~~accept~~. ~~reject~~ accept
  - ↳ If M' accepts $w_2 w_1$ and rejects everything else then that can be fed to M, implying that M will accept the strings M' didn't, meaning $w_1 w_2$. else wise this rejects.

∴ This thus has created a decider for all languages which is A_{TM}, which we know to be impossible. Therefore we have proven L isn't decidable.

5. Show NP is closed under union and concatenation.

(a) <u>union</u>: TM M_1 and TM M_2 we
we to show NP decider for $TM_{M_1 \cup M_2}$.

- Let A be a polynomial time verifier by definition 7.18
- Run both M_1 and M_2 on A, if either accept then accept, else reject.
- the time for each would be $T(M_1) + T(M_2)$ therefore the total time is $T(M_1 + M_2)$.

$\therefore$ This shows that NP is closed under union.

(b) <u>Concatenation</u>:

- Let A be a polynomial time verifier by definition 7.18
- Need to show a definition by $TM_{M_1 \circ M_2}$

1 ⇒ Split input A into $A_1 A_2$. Run TM_1 on A_1, if it rejects then reject.

2 ⇒ Then run TM_2 on A_2, if it rejects then reject.

3 ⇒ If neither reject then accp.

is linear $T(M_1 + M_2)$ therefore
decider.

6. • $G = (V, E)$ is an undirected graph.

• A set $S \subseteq V$ is a dominating set if every vertex in $V \setminus S$ is adjacent to some node in S.

• DOMINATING-SET = $\{\langle G, k\rangle \mid G$ has a dominating set of size $k\}$

• Prove that DOMINATING-SET is NP-Complete, by giving a reduction from VERTEX-COVER.

↳ I wasn't sure at all how to do this.