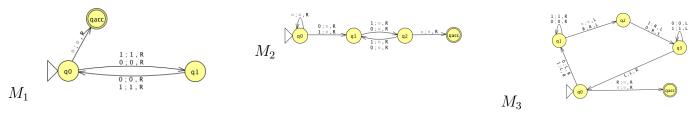
Week8 monday

Acceptance problem $\{\langle M, w \rangle \mid M \text{ is a Turing machine that accepts input string } w\}$ for Turing machines A_{TM} Language emptiness testing $\{\langle M \rangle \mid M \text{ is a Turing machine and } L(M) = \emptyset\}$ for Turing machines E_{TM} Language equality testing

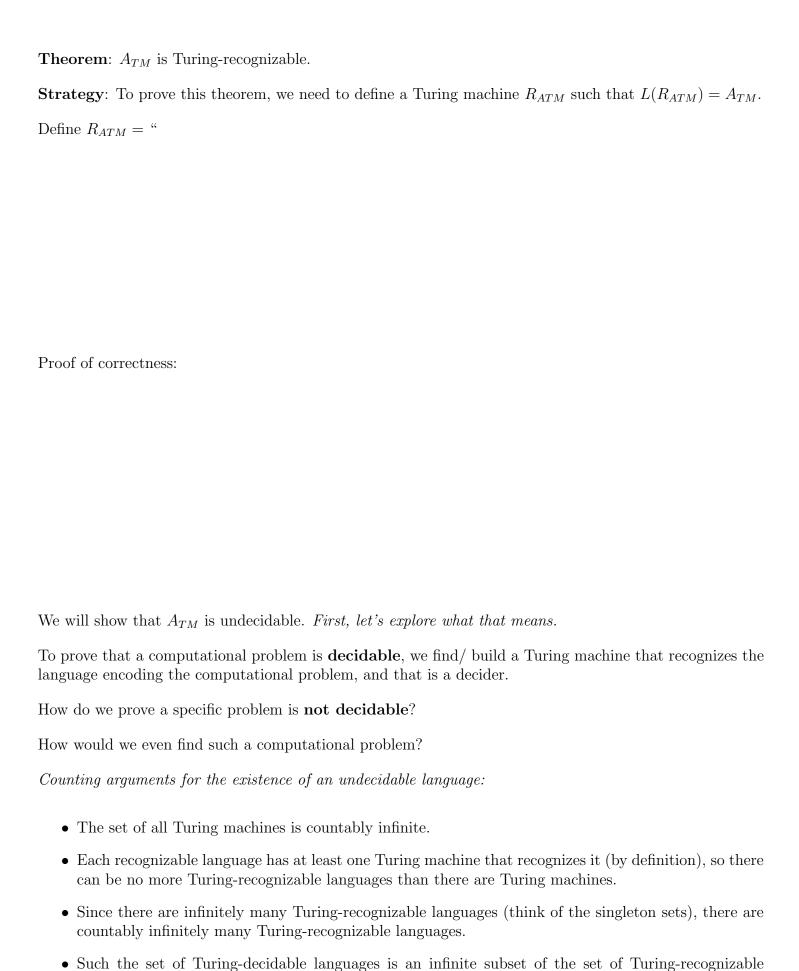
for Turing machines $\{\langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are Turing machines and } L(M_1) = L(M_2)\}$ EQ_{TM}



Example strings in A_{TM}

Example strings in E_{TM}

Example strings in EQ_{TM}



languages, the set of Turing-decidable languages is also countably infinite.

Since there are uncountably many languages (because $\mathcal{P}(\Sigma^*)$ is uncountable), there are uncountably many unrecognizable languages and there are uncountably many undecidable languages.
Thus, there's at least one undecidable language!
What's a specific example of a language that is unrecognizable or undecidable?
To prove that a language is undecidable, we need to prove that there is no Turing machine that decides it.
Key idea : proof by contradiction relying on self-referential disagreement.
Theorem : A_{TM} is not Turing-decidable.
Proof : Suppose towards a contradiction that there is a Turing machine that decides A_{TM} . We call this presumed machine M_{ATM} .
By assumption, for every Turing machine M and every string w
• If $w \in L(M)$, then the computation of M_{ATM} on $\langle M, w \rangle$
• If $w \notin L(M)$, then the computation of M_{ATM} on $\langle M, w \rangle$
Define a new Turing machine using the high-level description:
$D=$ " On input $\langle M \rangle$, where M is a Turing machine:
1. Run M_{ATM} on $\langle M, \langle M \rangle \rangle$.
2. If M_{ATM} accepts, reject; if M_{ATM} rejects, accept."
Is D a Turing machine?
Is D a decider?
What is the result of the computation of D on $\langle D \rangle$?

