$REGULAR_{TM} = \{\langle M \rangle \mid M$ is a TM and $L(M)$ is regular $\}$

$A_{TM} \leq REGULAR_{TM}$

$\langle M, w \rangle$ $\langle M_1 \rangle$

I am going to create special TM M_1

M_1 = "On input $\langle x \rangle$, M and w are encoded in this TM

1. If $x \in 0^n 1^n$, accept
2. If $x \notin 0^n 1^n$, run M on w and accept x if M accepts w.

M accepts w $\Longrightarrow$ $L(M_1) = \Sigma^*$

$L(M_1)$ is regular.

M does not accept w $\Longrightarrow$ $L(M_1) = \{0^n 1^n \mid n \geq 0\}$

not a regular language.

Let's create the decider for A_{TM}.

Assume R is the decider for $REGULAR_{TM}$.

Let's create the decider S for A_{TM}

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

1. construct TM M_1 by encoding $\langle M, w \rangle$ in its constrouction
2. Run R on $\langle M_1 \rangle$
3. If R accept $\langle M_1 \rangle$, accept
   If R rejects $\langle M_1 \rangle$, reject

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

$$A_{TM} \leq EQ_{TM}$$

$\langle M, w \rangle$ $\qquad$ $\langle M_1, M_2 \rangle$

$$E_{TM} \leq EQ_{TM}$$

Assume EQ_{TM} is decidable, and TM R decides it.

Then we construct TM S to decide E_{TM}.

S = on input $\langle M \rangle$, where M is a TM

1. Run R on $\langle M, T \rangle$ where TM T does not accept any string.
2. If R accepts, accept
   If R rejects, reject.

Q: what is $\overline{A_{TM}}$?

$A_{TM} = \{ \langle M, w \rangle \mid M$ is a TM and M accepts $w \}$

$\overline{A_{TM}} = \{ \langle M, w \rangle \mid M$ does not accept $w \}$

Turing-unrecognizable languages

Any language that is not turing-recognizable is called turing-unrecognizable.

Def

A language is co-turing recognizable if it is the complement of a turing recognizable language

$\overline{A_{TM}}$ is co-Turing recognizable.

Theorem 4.22: A language is decidable if and only if it is both Turing recognizable and co-Turing recognizable.

Proof: Suppose A is recognized by TM M_1 and $\overline{A}$ is recognized by TM M_2, then we construct TM M which is a decider for A.

$M =$ " on input w:

1. Run M_1 and M_2 on w in parallel.

2. If M_1 accept, accept; If M_2 accept, reject.