A Turing-Unrecognizable language

A_{TM} it is not decidable

A_{TM} is Turing-recognizable

U is a turing machine that recognizes A_{TM}

U = "On input $\langle M, w \rangle$, where M is TM and w is a string:

1. Simulate M on input w
2. If M ever enters accept state, accept

   If M ever enters reject, reject

$A_{TM} \notin \Sigma_0$ $\quad A_{TM} \in \Sigma_1$

Complement of a language $\ell = \bar{\ell}$

$\bar{\ell} = \{ w \mid w \text{ is not in } \ell \}$

A language is <u>co-turing recognizable</u> if it is the complement of a TM recognizable lang.

A language is decidable iff it is TR and co-TR

$\forall \ell \in ALL.\ \ell \in \Sigma_0 \text{ iff } \ell \in \Sigma_1 \wedge \bar{\ell} \in \Sigma_1$

$(A \text{ iff } B) = (A \rightarrow B) \wedge (B \rightarrow A)$

2 directions

forward: $\underline{l \in \Sigma_0} \rightarrow l \in \Sigma_1 \wedge \bar{l} \in \Sigma_1$

$\underline{l \in \Sigma_1}$ = trivial

$\bar{l} \in \Sigma_1$ = "run $l(w)$, negate answer"

Backwards: $l \in \Sigma_1 \wedge \bar{l} \in \Sigma_1 \longrightarrow l \in \Sigma_0$

let YESA be the recognizer for l

YESA can diverge when $w \notin A$

let NOA be the recognizer for $\bar{l}$

NOA can diverge when $w \in A$

ALWAYSA

M = "On input w:

1. Run both YESA and NOA on input w in <u>parallel</u>
2. If YESA accepts, accept
   if NOA accepts, reject

Parallel = ALWAYSA has 2 tapes

ALWAYSA always halts, it is a decider.

$l \in \Sigma_0$

QED.

$(A \text{ iff } B) \Rightarrow (\neg A \text{ iff } \neg B)$

$\ell \notin \Sigma_0 \text{ iff } \neg(\ell \in \Sigma_1 \wedge \bar{\ell} \in \Sigma_1)$

$\underline{\neg(A \wedge B) = \neg A \vee \neg B}$

$\Downarrow$

$\ell \notin \Sigma_1 \text{ or } \bar{\ell} \notin \Sigma_1$

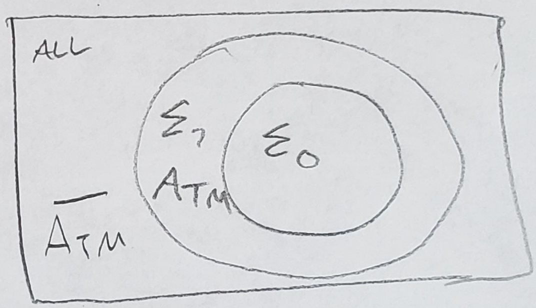
Is $\overline{A_{TM}}$ TR? No.

$A_{TM} \notin \Sigma_0 \rightarrow \underline{A_{TM} \notin \Sigma_1} \vee \overline{A_{TM}} \notin \Sigma_1$

We already know $A_{TM} \in \Sigma_1$

must be true.

QED.



Reducibility

"primary method for proving that problems are computationally unsolvable"

reduction - a way of converting one problem into a second problem such that a solution to the second problem can be used to solve the first problem.

$A \leq_m B$ $\quad \exists f. \forall w \in \Sigma^*. w \in A$ iff $f(w) \in B$

Ex. Problem of finding your way in NYC <u>Reduces</u> to the problem of finding a map.

If A $\leq$ is reducible to B, and B is decidable, A is also decidable.

if $A \leq_m B$ and $B \in \Sigma_0$, then $A \in \Sigma_0$

If A is undecidable and reducible to B, B is also undecidable

if $A \notin \Sigma_0$ $A \leq_m B$, then $B \notin \Sigma_0$

A is "Big" B is "small"

A can't work so B can't work

Method to prove something undecidable is to show that some other problem already known to be undecidable reduces to it.

Halting Problem.

$A_{TM} \notin \Sigma_0$ $\quad A_{TM} = \{ \langle M, w \rangle \mid M \in TM,\ M \text{ accepts } w \}$

$HALT_{TM} \notin \Sigma_0$ $\quad HALT_{TM} = \{ \langle M, w \rangle \mid M \in TM,\ M \text{ halts on } w \}$

Proof by contradiction.

let R be a TM that <u>decides</u> $HALT_{TM}$

Construct a TM S that <u>decides</u> A_{TM}

S = " On input $\langle M, w \rangle$
1. Run R on input $\langle M, w \rangle$
2. If R rejects, reject
3. If R accepts, simulate M on w until it Halts.
4. If M accepts, accept
   If M rejects, reject

But we know that $A_{TM} \notin \Sigma_0$

Therefore, by contradiction, $HALT_{TM} \notin \Sigma_0$

QED.