# Lecture 13

# $EQ_{TM}$ is neither recognizable nor co-recognizable

$M_0$: on any input x, reject x. $L(M_0) = \varnothing$

$M_1$: on any input x, accept x. $L(M_1) = \Sigma^*$

For any <M,w>, let h(<M,w>) = $M_2$ be the TM that, on input x,

1. runs M on w
2. if M accepts w, then accept x.

Claim: $L(M_2) = \Sigma^*$ (if $\langle M,w\rangle \in A_{TM}$), else $= \varnothing$ & h computable

Then $\overline{A_{TM}} \leq_m EQ_{TM}$ via $g(\langle M,w\rangle) = \langle M_0,M_2\rangle$

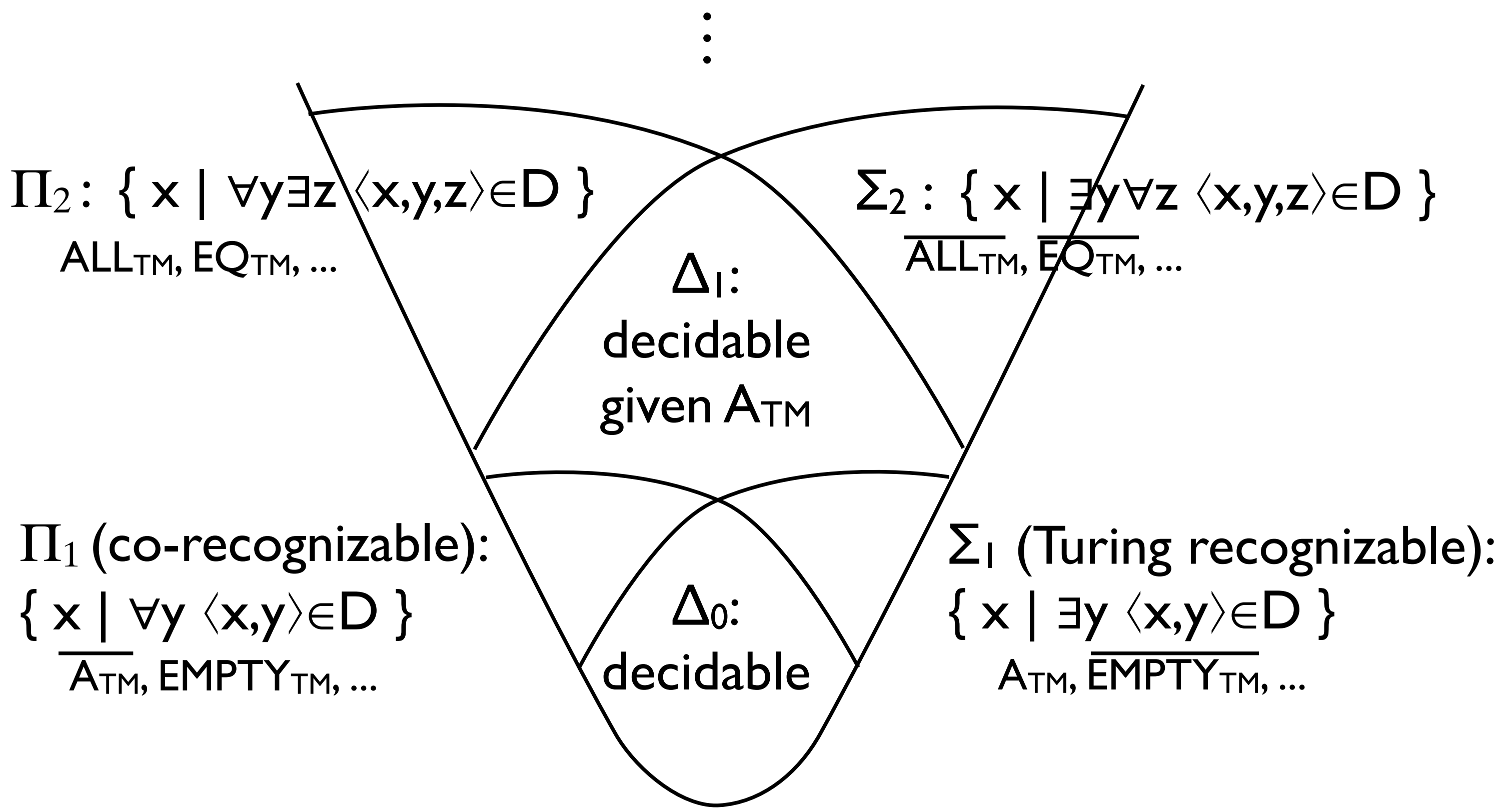
And $\overline{A_{TM}} \leq_m \overline{EQ_{TM}}$ via $f(\langle M,w\rangle) = \langle M_1,M_2\rangle$ (& $A_{TM} \leq_m EQ_{TM}$)

# Defining Inequivalence

"If two TMs are not equivalent, there is some input w where they differ, and if they differ there is some time t such that one accepts within t steps, but the other will not accept no matter how long you run it."

$\overline{EQ_{TM}} = \{ x \mid \exists y \, \forall z \, \langle x,y,z \rangle \in D \}$ where the decidable set D = { triples $\langle x,y,z \rangle$ such that x is a pair of TMs, y is a pair w,t, and one machine accepts w within t steps but the other has not accepted w within z steps }

# The “Arithmetical Hierarchy”




Potential Utility: It is often easy to give such a quantifier-based characterization of a language; doing so suggests (but doesn’t prove) whether it is decidable, recognizable, etc. and suggests candidates for reducing to it.

“The human mind seems limited in its ability to understand and visualize beyond four or five alternations of quantifier. Indeed, it can be argued that the inventions, subtheories, and central lemmas of various parts of mathematics are devices for assisting the mind in dealing with one or two additional alternations of quantifier.”

H. Rogers, *The Theory of Recursive Functions and Effective Computability,* McGraw-Hill, 1967, pp 322-323.

# Decidability Questions

Questions about a single TM:

- Detail questions: about *operation* or *structure* of a TM
  - useless state, does head move left, does it take >100 steps, ...
- Bottom-line questions: ask about a TM's *language*
  - Is L(M) empty? Infinite? Is 42 in L(M)? ...
  - About L(M), not M, *per se*. *Same answer for M' if L(M)=L(M')*

Other: Questions about ⟨M,w⟩, 2 TMs, grammars, ...

# Language Properties

We formalize language *properties* simply as sets of languages

E.g., the "infiniteness" property is just the set of infinite languages.

A property is *non-trivial* if there is at least one language with the property and one without.

E.g., "emptiness" is nontrivial: $L_1=\varnothing$ has it; $L_2=\{42\}$ doesn't.

E.g., "countable" is trivial: every subset of $\Sigma^*$ is countable

# Rice's Theorem

Theorem:

For every nontrivial property $\mathcal{P}$ of the Turing recognizable languages, it is undecidable whether a TM recognizes a language having property $\mathcal{P}$. I.e.,

$$\mathcal{P}_{TM} = \{ \langle M \rangle \mid L(M) \in \mathcal{P} \}$$

is undecidable.

Corr:

$EMPTY_{TM}$, $INFINITE_{TM}$, $REGULAR_{TM}$, ... all undecidable

# Rice's Theorem

$\mathcal{P}_{TM}$ = { <M> | M is a TM s.t. L(M) $\in \mathcal{P}$ }

M' on input x:
1. save x
2. write w
3. run M on w
4. if M accepts w, then run $M_1$ on x

L(M') = ?

# Rice's Theorem

$\mathcal{P}_{TM}$ = { <M> | M is a TM s.t. $L(M) \in \mathcal{P}$ }

M' on input x:

1. save x
2. write w
3. run M on w
4. if M accepts w, then run $M_1$ on x

$$L(M') = \begin{cases} L(M_1), & \text{if M accepts w} \\ \emptyset, & \text{if M rejects w} \end{cases}$$

# Rice's Theorem

$\mathcal{P}_{TM}$ = { <M> | M is a TM s.t. $L(M) \in \mathcal{P}$ }

Pf: To show: $A_{TM} \leq_T \mathcal{P}_{TM}$. WLOG, $\emptyset \notin \mathcal{P}$; $M_1$ is a TM s.t. $L(M_1) \in \mathcal{P}$

On input <M,w> build M' :

M' on input x:
1. save x
2. write w
3. run M on w
4. if M accepts w, then run $M_1$ on x

Do *not* run M or M'. (That whole "halting thing" means we might not learn much if we did.) But note that L(M') is/is not in $\mathcal{P}$ exactly when M does/does not accept w, so knowing whether $L(M') \in \mathcal{P}$ answers whether <M,w> is in $A_{TM}$.

I.e., $A_{TM} \leq_T EMPTY_{TM}$

$$L(M') = \begin{cases} L(M_1), & \text{if M accepts w} \\ \emptyset, & \text{if M rejects w} \end{cases}$$

NB: it shows $A_{TM} \leq_m \mathcal{P}_{TM}$ or $\overline{\mathcal{P}_{TM}}$

Programs, in general, are opaque, inscrutable, confusing, complex, obscure, and generally yucky...

(If you've been a 142 TA, you might have observed this yourself...)

# Decidability Questions

Questions about a single TM:

- Detail questions: about *operation* or *structure* of a TM
  - useless state, does head move left, does it take >100 steps, ...
- Bottom-line questions: ask about a TM's *language*
  - Is L(M) empty? Infinite? Is 42 in L(M)? ...
  - About L(M), not M, *per se*. *Same answer for M' if L(M)=L(M')*

Other: Questions about ⟨M,w⟩, 2 TMs, grammars, ...

Rice's theorem doesn't (directly) answer these

But it says all these are undecidable (or trivial)