23 April 2018 – Reducing from the co-halting problem to show sets are not r.e.
– Rice's Theorem

$\overline{H} = \{ x;y \mid U(x;y) = \nearrow \} \cup \{$ strings that don't contain exactly one ';' $\}$

$=$ "the co-halting problem" $= \Sigma^* \setminus H$.

We've seen that H is r.e. but not decidable.
Recall: a set is decidable $\iff$ it and its complement are r.e.

$\therefore$ $\overline{H}$ is not r.e.

This makes $\overline{H}$ a useful starting point for reductions showing other sets are not r.e.

__Example.__ Let $L = \{ x \mid x$ is a description of a Turing machine M and $|L(M)| = \infty \}$.
Prove that L is not r.e.

(We should suspect L is not r.e. because how would you give a __finite__ amount of evidence that proves that M accepts ∞ many strings?)

__Proof.__ To prove L is not r.e. we reduce from $\overline{H}$.
In other words given a hypothetical machine M_L that accepts strings iff they belong to L, we construct a machine $M_{\overline{H}}$ that accepts strings if and only if they belong to $\overline{H}$, using M_L as a subroutine.

Goal of the reduction should be: given $x;y$ transform it to description, z, of a machine $M'_{x;y}$ such that

$$x;y \in \overline{H} \iff z \in L$$

If we can do this, $M_{\overline{H}}$ can operate as follows.
1. Transform $x;y$ into z.
2. Execute M_L on input z.

As long as step 1 is guaranteed to run in finite time,

$$x;y \in \overline{H} \iff z \in L \iff M_L \text{ accepts } z \iff M_{\overline{H}} \text{ accepts } x;y$$

$\therefore$ $\overline{H} = L(M_{\overline{H}})$ $\lightning$

Goal of the reduction should be: given $x;y$ transform it to description, z, of a machine $M'_{x;y}$ such that

$$x;y \in \overline{H} \iff z \in L$$

Let M_x denote the machine described by x.

M_x runs forever on input y $\iff$ $M'_{x;y}$ accepts infinitely many inputs

$\Updownarrow$

$M_x(y)$ has infinite running time

<u>Idea:</u> $M'_{x,y}$ takes an input string, w, and checks if $\exists t \geq 0$ if w represents the first t transitions of M_x running on input y.

In other words $M'_{x;y}$ checks if w has the form

$$w = w_0 \# w_1 \# w_2 \# \cdots \# w_t$$

where each w_i has the form (x, q, k) s.t. $x \in \Sigma^*$, $q \in K(M_x)$, $k \in \mathbb{N}$

and checks that $w_{i-1} \xrightarrow{M_x} w_i$ holds for $i = 1, \ldots, t$.

and checks that $w_0 = (y, s, \emptyset)$.

This works because the # of strings that $M'_{x;y}$ accepts is identical to the # of steps in the computation $M_x(y)$.

<u>Rice's Theorem.</u> (informally) Any property of a Turing machine that depends <u>only</u> on the set of strings it accepts is either trivial or undecidable.

<u>Recall.</u> For $x \in \Sigma^*$ we defined $L(x) = \begin{cases} L(M) & \text{if } x \text{ is a description of } M \\ \emptyset & \text{if } x \text{ is not a TM descrip} \end{cases}$

RICE'S THM

Let P be a set of strings, $P \subseteq \Sigma^*$, such that
$$\forall x,y \quad (L(x) = L(y)) \Longrightarrow (x \in P \Leftrightarrow y \in P).$$

Then either
① $P = \emptyset$
② $P = \Sigma^*$ } Trivial
③ P is undecidable. } Non-trivial.

Proof. See lecture notes.

Applications. All of the following are undecidable.

(1) Given x is $|L(x)| = \infty$? (We already showed this isn't r.e.) (Undecidable is a weaker conclusion.)

(2) Is $L(x) \neq \emptyset$? i.e. does a specified Turing machine accept at least one string?

(3) Does $L(x)$ contain the empty string?

(4) Does $L(x)$ contain at least one string of length k for every $k \in \mathbb{N}$?

Examples of problems not covered by Rice's Theorem:

(A) Does the machine described by x run for more than 100 steps on every input string?