Today: more examples of "unsolvable" problems; using reductions.

"Problem" means a language (i.e. a "decision problem") (i.e. an $L \subseteq \Sigma^*$, some $\Sigma$)

"Unsolvable" means either

1. undecidable

or

2. unrecognizable
Last time:

\[
GROUCHO มาในSELF
\]

= 

\[
\text{NON}\_\text{SELF}_\text{-ACCEPTING}
\]

= 

\[
\left\{ p \in \Sigma^+ \mid p \in \text{Language recBy}(p) \right\}
\]

\[
\text{Lang recBy} : \Sigma^* \to \text{Power}(\Sigma^*)
\]

\[
\text{Lang recBy}(p) = \left\{ \{ i \mid p \text{ accepts } i \} \right\} \text{ if } \{ p \} \text{ is valid}
\]
Canter's Thm \(\Rightarrow\)
\[
\{ p \mid p \notin \text{Lang RecBy}(p) \} = \top
\]

is unrecognizable.

We used a "reduction" to show that

**Non-Acceptance**

\[
= \{ \sigma_0 \mid p \text{ does not accept } i \}
\]

Our convention of describing a string \(p\), followed by another string \(i\), \(\sigma_0 = \langle \text{BELL} \rangle\)
is really some way to express \( \langle p, i \rangle \), i.e. a description of a pair \((p, i)\) where we anticipate:

\[ p = \text{Python program} \in \Sigma^+_{\text{ASCII}} \]

\[ i = \text{some input to } p \in \Sigma^+_{\text{ASCII}} \]

We proved that

\[
\begin{cases}
    \text{if NON-ACCEPTANCE were recognizable} \\
    \text{then } T = \text{GROUCHO-MARKX-SELF or NON-SELF-ACCEPTANT}
\end{cases}
\]

would also be recognizable.
Decidable vs. recognizable

- We say Python program, p, is a \textbf{decider} if on any $i \in \Sigma^*$, p on input $i$ is either

- "yes" — p accepts $i$
- "no" — p rejects $i$

(other possibilities are called "loops,"
i.e. p loops on $i$)

\L \in \Sigma^*$ is decidable if some
decider, p, recognizes \L,
i.e.,

$$L = \text{LanguageRecBy}(p)$$

$$= \{ i \mid p \text{ accepts } i \}$$

$$= \{ \sigma_0 i \mid p \text{ accepts } i \}$$

e.g.,

$$\text{ACCEPTANCE} = \{ \sigma_0 i \mid p \text{ accepts } i \}$$

and

$$\text{SELF-ACCEPTANCE}$$

$$= \{ p \mid p \in \text{LanguageRecBy}(p) \}$$

are recognizable (!!!)
What is \( \text{LangRecBy} \) (verySilly-py)?

\[
\text{LangRecBy} = \{ \langle n \rangle \mid n \in \mathbb{Z}, n \leq 5 \}
\]

= \{ \text{strings that represent integers } \leq 5 \}

verySilly-py is not a decider.
But:

\[ \text{ACCEPTANCE} = \{ p \in \Sigma^* \mid p \text{ accepts } i \} \]

\[ \text{is} \quad \{ \text{undecidable} \}
\quad \{ \text{recognizable} \} \]

Why recognizable?

Just:

1. simulate \( p \) on input \( i \)
2. run Python -m pdb \( p \)
3. run debugger -.
4. also called \textbf{universal} Python prog
Proposition: If \( L \) is decidable, then \( \Sigma^* \setminus L \) is decidable.

- "complement of \( L \)"

\[ A \setminus B = \{ a \in A \mid a \notin B \} \]

Proof: If \( p \) decides \( L \)

\[ p = \begin{cases} 
\text{return ("no")}, & \text{if } p \text{ decides } L \\
\text{return ("no")}, & \text{return ("yes")}, \\
\text{return ("yes")}, & \end{cases} \]
Hence $p$ says "yes" iff $q$ says "no".

hence

$\text{LongRecBy}(q) = \Sigma^* \setminus \text{LongRecBy}(p)$

Prep: If $L$ is decidable (by $p$) then $L$ is recognizable. (by $p$)

\[ \text{PROC THAT ACCEPT AT LEAST ONE INPUT} \]

\[ = \{ p \mid p \text{ accepts some } x \in \Sigma^* \} \]