1. Fine point about complement:

$$\text{ACCEPTANCE} = \left\{ \sigma \in \mathcal{L} \mid \text{accepts } \sigma \right\}$$

$$\text{NON-ACCEPTANCE} = \left\{ \sigma \in \mathcal{L} \mid \text{does not accept } \sigma \right\}$$

$$\sum_{\text{ASCII}} \setminus \text{ACCEPTANCE} = ???$$

2. Russell’s Paradox explained ... (?)

3. Start regular languages.
\[ \sum_{\text{ASCII}} \setminus \text{ACCEPTANCE} = \text{NON-ACCEPTANCE} \]

\[ u \{ \text{string not of the form } p \sigma_0 \text{ with } p \text{ valid Python} \} \]

\[ \text{NON_PROG_PLUS_INPUT} \]

\[ = \{ S \in \Sigma^+_{\text{ascii}} \mid S \text{ does not contain } \sigma_0 \} \]

\[ u \{ S_1 \sigma_0 S_2 \mid S_1 \text{ does not contain } \sigma_0, \text{ but } S_1 \text{ is not a valid Python program} \} \]
\text{decidable}

(turns out in \text{poly} time)

What about:

\[ T = \{ S \mid S \notin S \} \]

\[ = \{ \text{all sets } S \text{ s.t. } S \notin S \} \]

either the set \( T \):

\[ T \in T \rightarrow \text{contradiction} \]

or

\[ T \notin T \rightarrow \text{contradiction} \]

\[ \{ \text{all sets} \} \] `too big` to be a set, "class"
Rem: If $S$ is a set, then $\text{Power}(S)$...

If not, Cantor's Thm -- 😞
could not work well...
Chapter 1 [Sip] Regular Languages, Finite Automata

Regular languages are the same set as those described by regular expressions

* vacation *

* uboc *

\[
\Sigma = \{a, b, c, a\}
\]

\[
\downarrow
\]

\[
\left\{ s \in \{a, b, c, u\}^* \mid s \text{ is of the form } s_1 \text{uboc}s_2, \ s_1, s_2 \in \Sigma^* \right\}
\]

\[
S_1 \text{ o (}u,b,c\text{) o } S_2
\]
Build a primitive algorithm called a finite state automaton

DFA - D is for deterministic

\[ \Sigma = \{a, b, c, \mu\} \]
So to start here

\[ Q \] is an "accepting" state "final" state

\[ \Sigma \]

\[ \delta : Q \times \Sigma \rightarrow Q \]

\[ \begin{array}{c}
\text{alphabet} \\

\downarrow
\end{array} \]

\[ \begin{array}{c}
\text{accepting state} \\

\downarrow
\end{array} \]

DFA = \( (Q, \Sigma, \delta, q_0, \mathcal{F}) \)

\[ \begin{array}{c}
\text{initial state} \\

\downarrow
\end{array} \]

\[ \begin{array}{c}
\text{set of in example states} \\

\downarrow
\end{array} \]

\[ Q = \{q_0, q_1, q_2, q_3\} \]
input \( \sigma_1 \sigma_2 \sigma_3 \cdots \sigma_n \)

Out step

1st step

2nd step

end at step n

\( \sigma_{h_{n-1}} \sigma_n \)

---

No class Monday