CPSC 421/501 Sept 22, 2023

1. Fine point about complement:

$$\text{ACCEPTANCE} = \{ \text{prog}, i \mid p \text{ accepts } i \}$$

$$\text{NON-ACCEPTANCE} = \{ \text{" } \text{ " } p \text{ does not accept } i \text{ "} \}$$

$$\Sigma_{\text{ASCII}}^{*} \setminus \text{ACCEPTANCE} = \text{???}$$

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

3. Start regular languages.

$$\Sigma^*_{\mathrm{ASCII}} \setminus \mathrm{ACCEPTANCE} = ???$$

$$= \mathrm{NON\text{-}ACCEPTANCE}$$

$$\cup \underbrace{\left\{ \text{string not of the form } p\,\sigma_c\, i \text{ with } p \text{ valid Python} \right\}}_{\mathrm{NON\_PROG\_PLUS\_INPUT}}$$

$$= \left\{ s \in \Sigma^*_{\mathrm{ASCII}} \;\middle|\; s \text{ does not contain } \sigma_c \right\}$$

$$\cup \left\{ s_1 \sigma_c s_2 \;\middle|\; \begin{array}{l} s_1 \text{ does not contain } \sigma_c, \\ \text{but } s_1 \text{ is not a} \\ \text{valid Python program} \end{array} \right\}$$

NON_PROG_PLUS_INPUT

is <u>decidable</u>

(turns out in poly time)

What about:

$$T = \{ S \text{ is a set} \mid S \notin S \}$$

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

either the set T:

$T \in T \rightarrow$ contradiction

or

$T \notin T \rightarrow$ "

intuitively

{all sets} "<u>too big</u>" to be a set. "class"

Rem: If S is a set, then

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

$\Sigma = \{u, b, c, a\}$

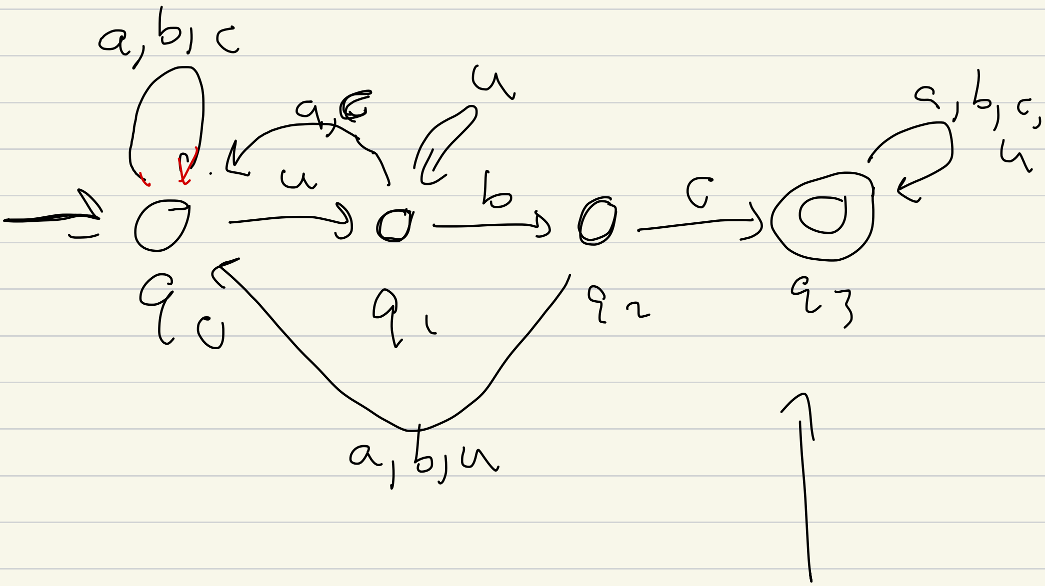
ubc

$\downarrow$

$\{ s \in \{a, b, c, u\}^* \mid s$ is of the form $\underbrace{s_1 ubc\, s_2}_{s_1 \circ (u,b,c) \circ s_2}$, $s_1, s_2 \in \Sigma^* \}$

Build a primitive algorithm called a finite state automaton

DFA – D is for deterministic



accept string

$\Sigma = \{a, b, c, u\}$

So

$\to \bigcirc$ start here

$\circledcirc$ is an "accepting" state

"final" state

$$\delta : Q \times \Sigma \to Q$$

alphabet

$F \subset Q$

accepting states

$$DFA = (Q, \Sigma, \delta, q_0, F)$$

initial state

set of states

in example

$Q = \{q_0, q_1, q_2, q_3\}$

input

$\sigma_1 \sigma_2 \sigma_3 \ldots \sigma_n$

"start"

0^{th} step

$\sigma_1 \sigma_2 \sigma_3 \ldots \sigma_n$

1^{st} step

$\sigma_1 \sigma_2 \sigma_3 \ldots \sigma_n$

2^{nd} step

$\vdots$

end at step n

$\ldots \sigma_{n-1} \sigma_n$

No class Monday