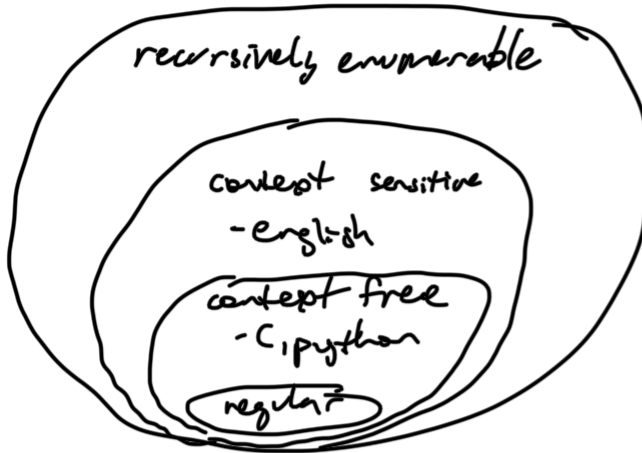


# Regular Expressions 3/6/2023

regex  
- awk, grep  
lexical analysis

## Theory: Finite Automata

hierarchy of languages:



DFA Deterministic  
- single transition per letter

NFA Nondeterministic  
- many transitions per letter

Push down automata  
- stack  
- context free languages

Linear bounded  
- context sensitive  
- tape

our computers  $\neq$   
- needs infinite  
memory

Turing machine  
- everything  
- finite state control  
- infinitely long tape  
- can compute all computable functions  
- mathematical concept

A Grammar - (type 3)

$\langle \overset{R}{\text{rule}}, \overset{N}{\text{non terminal}}, \overset{T}{\text{terminal}} \rangle$

$N = A$

$T = a, b$

Finite automata finds expression

DFA:  $\langle \Sigma, S, S_0, \delta, F \rangle$

$E = \text{input}$

$S = \text{finite set of states}$

$S_0 = \text{initial state}$

$\delta = \text{transition}$

$F = \text{set of final states}$

State machine:

- vending machines
- robotics

String:  $a \in \Sigma$   $a = \text{letter}$   
 $x \in \Sigma^*$   $x = \text{string}$   
 $\lambda \in \Sigma^0$  empty string

$$\Sigma^* = \cup$$

Non deterministic - all choices are made at the same time  
- np complete: can be solved by non deterministic Turing machine, cannot be solved by deterministic

equivalent deterministic machine of Non deterministic  
is exponentially bigger

deterministic finite machines can be optimized

regex comp - compiler for regular expression