① Regular expressions:

- → Symbol
- → Alphabet
- → String
- → Language

} definitions

Operators

Unary
- → Kleene's closure
- → Positive closure
- → Complement

($N_L = N_{\bar{L}}$)
→ no. of states in DFA

Binary
- → Union (+)
- → Concatenation (.)

P.T.O.

→ Prefix problems
→ Suffix "
→ Substring "

} Construct RE & DFA

Finite Automata:

DFA tuple

$$\{q_0, Q, \Sigma, \delta, F\}$$

Transition func:

$$Q \times \Sigma = Q$$

(unique i/p, singular output)

→ All DFA problems

- $|w| \geq 2$
- $|w| == 2$
- $|w| \leq 2$
- $|w| \% 3 == 0$
- $|w| \% 3 == 1$
- $|w|$ = even
- $|w|$ = odd

- ends with b
- ends with aa
- atleast 99 a's
- atmost 99 a's
- exactly 99 a's
- substring
- :
- :

→ Product Automaton

$$N_{ab} = N_a \times N_b$$

→ Product automaton problems

5a & 7b

10a & 12b

binary string x

⇒ x % 3 == 0

⇒ x % 5 == 0

⇒ x % 2 == 0

⇒ x % 6 == 0

⇒ x % 6 == 0
&
x % 12 == 0

⇒ x % 6 == 0
&
x % 8 == 0

→ DFA minimization (just refer)

NFA:

NFA tuple - - - -

Transition function:

$$Q \times \Sigma = 2^Q$$

→ NFA problems all

→ Conversion to DFA (refer)

Identities of RE:

→ Identifiers

→ Annihilators

→ All identities.
Especially $(a+b)^*$

If NFA has n states, DFA has 2^n states

ε-NFA

→ Expressive power

→ ε-closure

→ ε-closure practice problems.

PDA

→ Stack & stack operations

→ PDA Tuple

→ Default PDA is NPDA

→ DPDA ⟶ DCFL
PDA ⟶ CFL

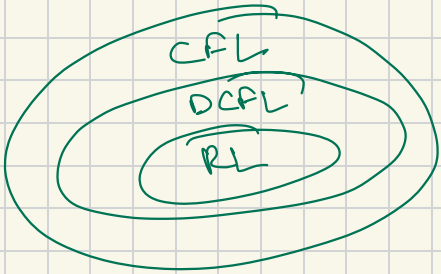
→ Transition Function

→ In PDA string is accepted when:
⇒ string reaches final state
⇒ Stack is empty

→ All PDA and DPDA problems.
(ww^R example for PDA explanation)

→ Expressive power

$$PDA > DPDA > DFA \cong NFA \cong \varepsilon\text{-}NFA$$



Identifying of Languages

→ RL:
- ⇒ Accepted by FA
- ⇒ Finite
- ⇒ Have Es
- ⇒ No dependancies

→ CFL:
- ⇒ 1 dep. b/w 2 parts of the expr. $\{a^n b^n\}$
- ⇒ ≥ 2 dep is "Not CFL".

→ DCFL:
- ⇒ Determinism

Language
- Finite
  - Yes → Regular
  - No → May or may not be regular
    - Dependancy?
      - Yes → Not regular
        - Dep == 1 → CFL / DCFL
        - Dep ≥ 2 → Not CFL
      - Need to check if:
        - → RL
        - → CFL
        - → DCFL
        - → Not CFL

37 Language identification problems:

Grammar

- Terminals, Non-terminals, Productions
- Tuple
- Regular Grammar recognition:
  Either Left-linear Grammar
  (or)
  Right linear rammar
- Left recursive grammar
- Right recursive grammar
- Leftmost Derivation and Rightmost Derivation } Unfurling productions.
- Ambiguous grammar