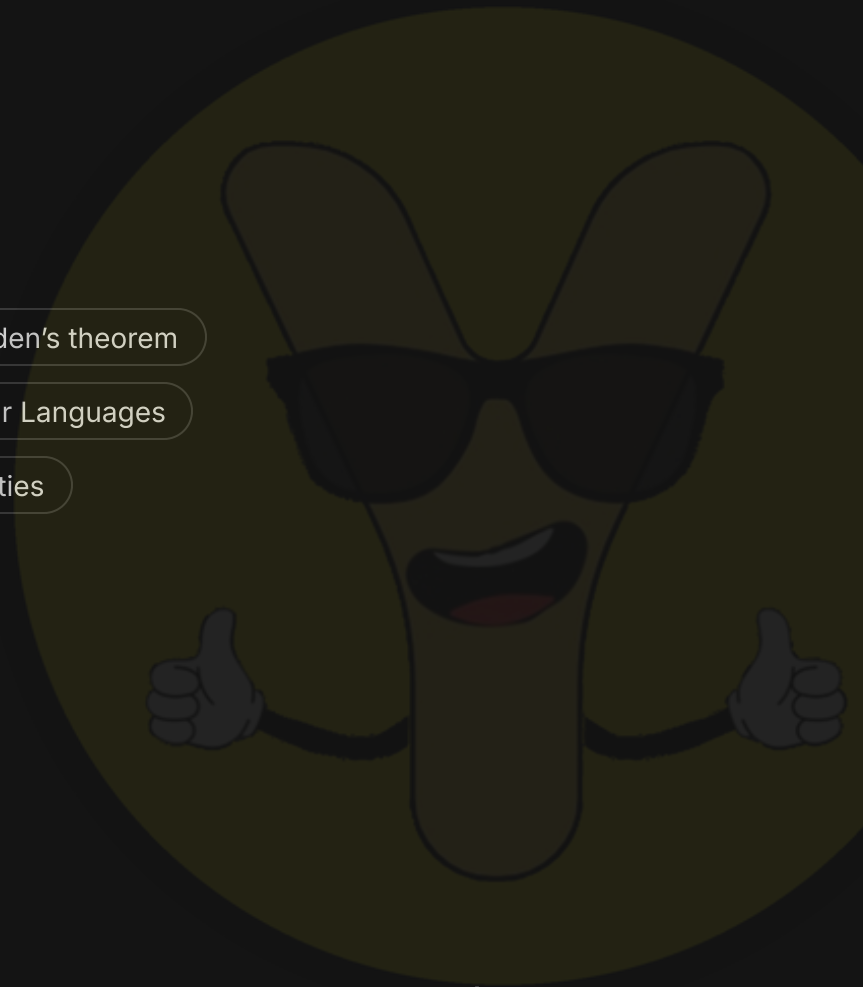
AKTU · 2ND YEAR · SEMESTER 4 · BCS402

Theory of Automata and Formal Languages

Unit 2 — Regular Expressions and Languages

- Regular Expressions
- Transition Graph
- Kleen's Theorem
- Finite Automata and Regular Expression- Arden's theorem
- Algebraic Method Using Arden's Theorem
- Regular and Non-Regular Languages- Closure properties of Regular Languages
- Pigeonhole Principle
- Pumping Lemma
- Application of Pumping Lemma
- Decidability- Decision properties
- Finite Automata and Regular Languages



DECODED BY

Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

Regular Expressions (RE)

REGULAR EXPRESSION

A regular expression is a sequence of characters that defines a search pattern, used to represent regular languages over an alphabet Σ.

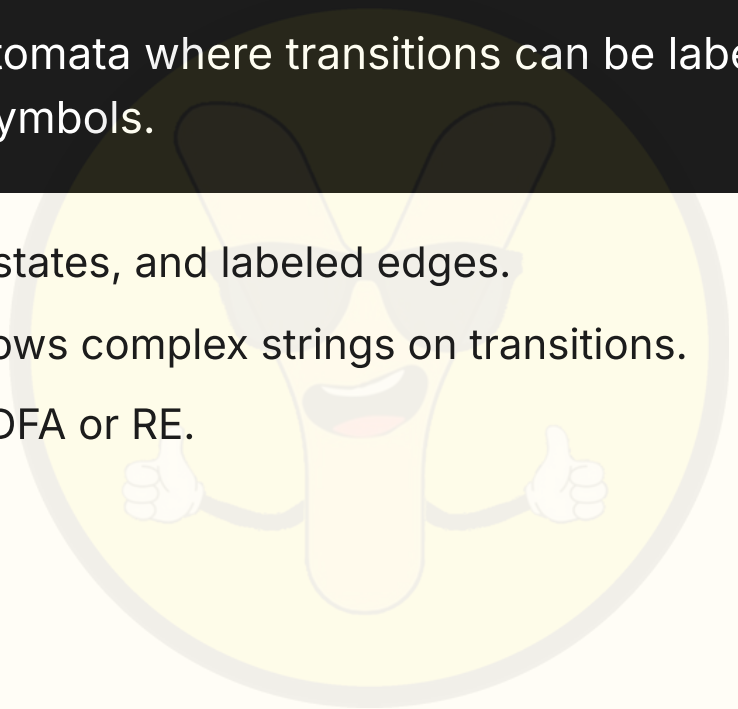
- Base cases: ε, φ, and each symbol 'a' in Σ are REs.
- Recursive steps: If r and s are REs, then (r+s), (rs), and (r*) are also REs.
- Used for pattern matching, lexical analysis, and defining language structure.

Transition Graph (TG)

TRANSITION GRAPH

A generalized version of Finite Automata where transitions can be labeled with any regular expression instead of just single symbols.

- Consists of states, start state, final states, and labeled edges.
- More flexible than DFA/NFA as it allows complex strings on transitions.
- Can be converted to an equivalent DFA or RE.



Kleene’s Theorem

KLEENE'S THEOREM

States that a language is regular if and only if it can be described by a regular expression, a finite automaton, or a transition graph.

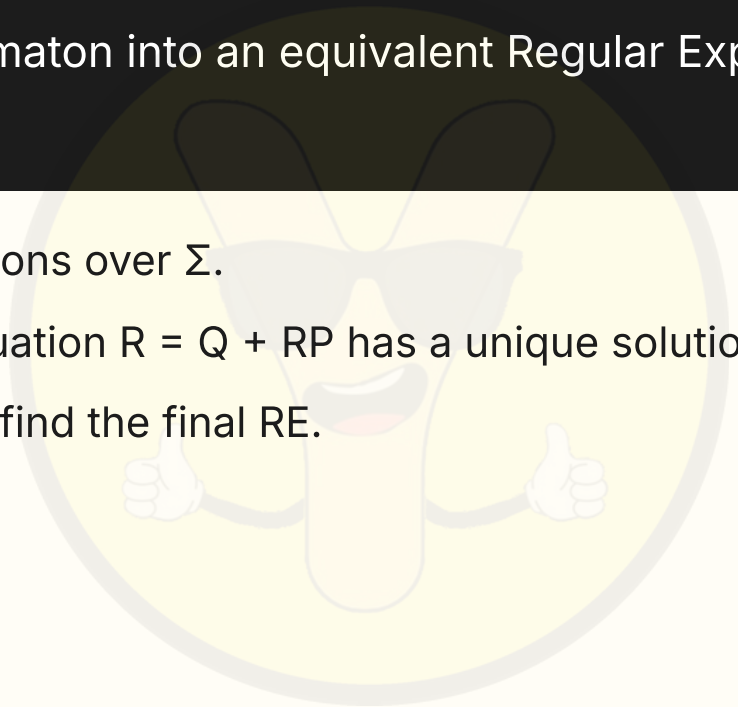
- Part 1: Every language defined by an RE is accepted by some FA.
- Part 2: Every language accepted by an FA can be described by an RE.
- Establishes the equivalence between machine models and algebraic models.

Arden’s Theorem

ARDEN'S THEOREM

A method to convert a Finite Automaton into an equivalent Regular Expression by solving systems of linear equations.

- Let P and Q be two regular expressions over Σ.
- If P does not contain ε, then the equation R = Q + RP has a unique solution R = QP*.
- Used to eliminate states in an FA to find the final RE.

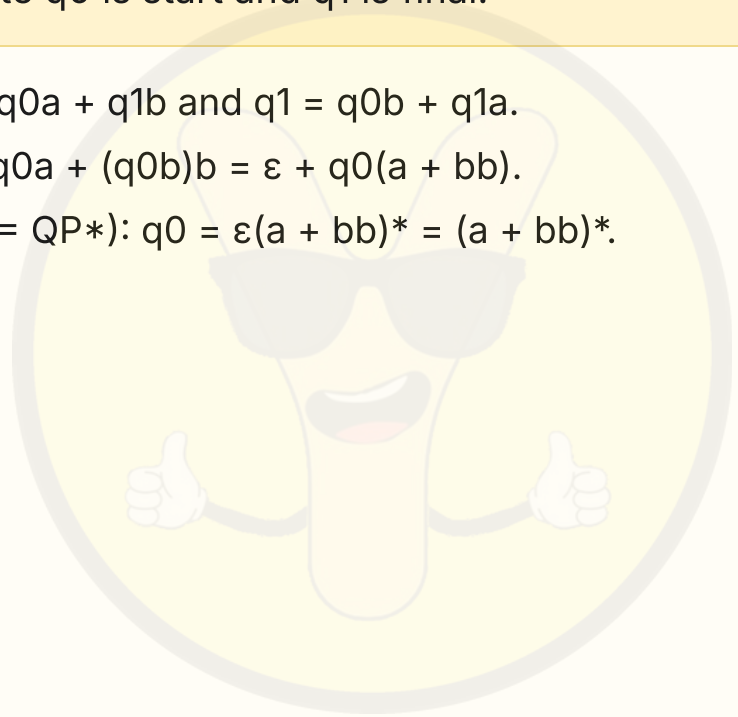


Algebraic Method Using Arden's Theorem

Find the RE for a two-state FA where state q0 is start and q1 is final.

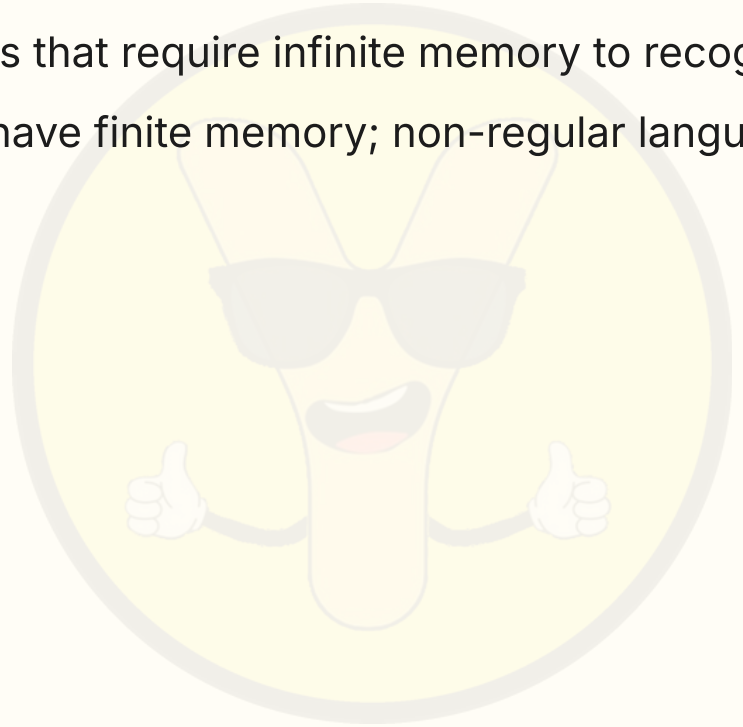
1. Write equations for each state: q0 = ε + q0a + q1b and q1 = q0b + q1a.
2. Substitute q1 into q0 equation: q0 = ε + q0a + (q0b)b = ε + q0(a + bb).
3. Apply Arden's theorem (R = Q + RP → R = QP*): q0 = ε(a + bb)* = (a + bb)*.
4. Calculate q1: q1 = (a + bb)*b.

The final RE is (a + bb)*b.



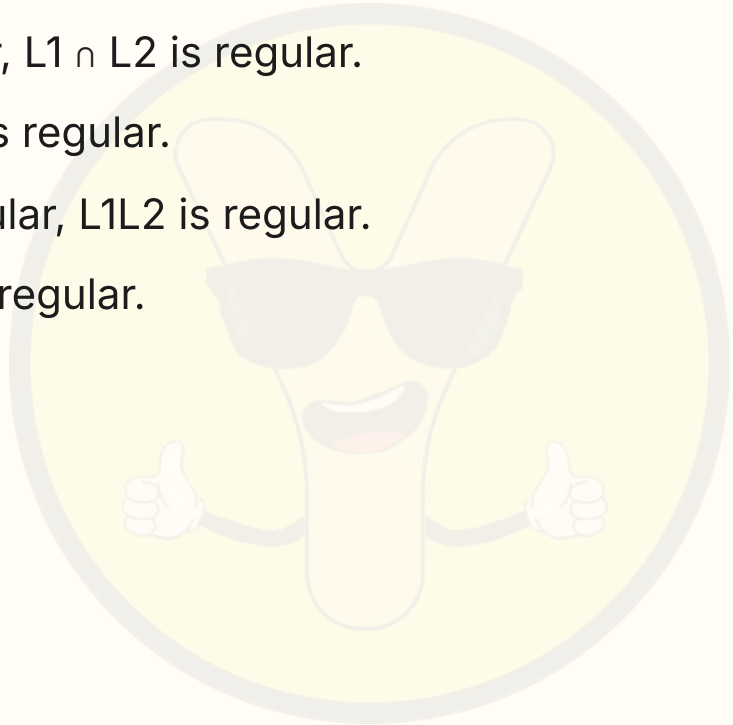
Regular vs. Non-Regular Languages

- Regular Languages: Languages that can be recognized by a DFA, NFA, or represented by an RE.
- Non-Regular Languages: Languages that require infinite memory to recognize, such as {a^n b^n | n ≥ 0}.
- Key difference: Regular languages have finite memory; non-regular languages involve counting or matching dependencies.



Closure Properties of Regular Languages

- Union: If L1 and L2 are regular, L1 ∪ L2 is regular.
- Intersection: If L1 and L2 are regular, L1 ∩ L2 is regular.
- Complement: If L is regular, Σ* - L is regular.
- Concatenation: If L1 and L2 are regular, L1L2 is regular.
- Kleene Closure: If L is regular, L* is regular.



Pigeonhole Principle

PIGEONHOLE PRINCIPLE

If n items are put into m containers, with n > m, then at least one container must contain more than one item.

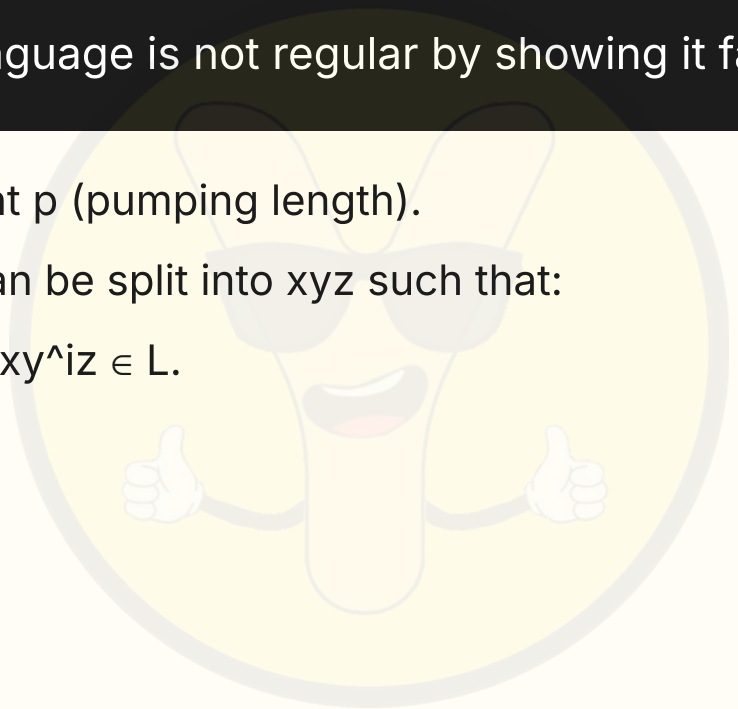
- Fundamental tool in proving non-regularity.
- In FAs, if a string has length ≥ number of states, a state must be repeated.
- This repetition creates a cycle in the path, allowing for pumping.

Pumping Lemma for Regular Languages

PUMPING LEMMA

A theorem used to prove that a language is not regular by showing it fails the 'pumping' condition.

- If L is regular, there exists a constant p (pumping length).
- For any string $s \in L$ with $|s| \geq p$, s can be split into xyz such that:
- 1. $|xy| \leq p$, 2. $|y| > 0$, 3. For all $i \geq 0$, $xy^iz \in L$.

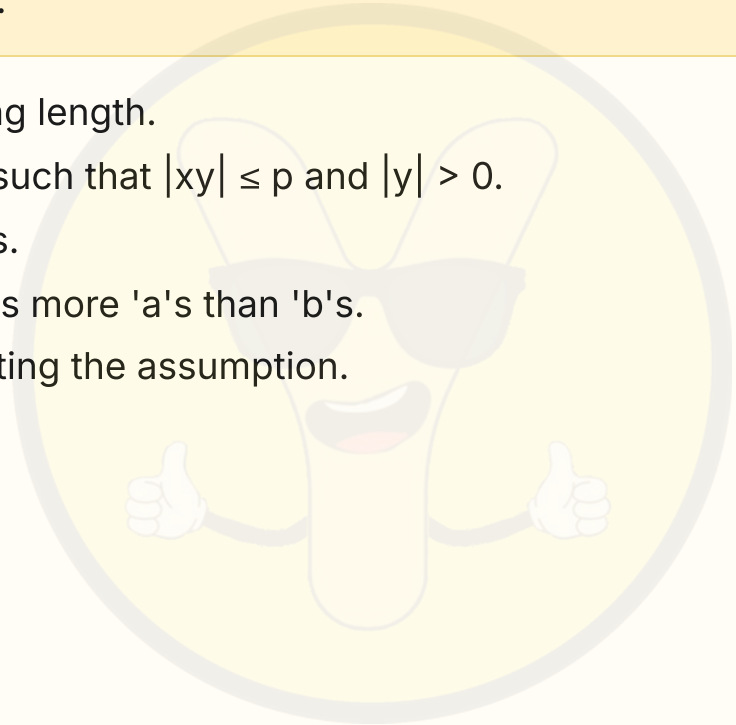


Application of Pumping Lemma

Prove L = {a^n b^n | n ≥ 0} is not regular.

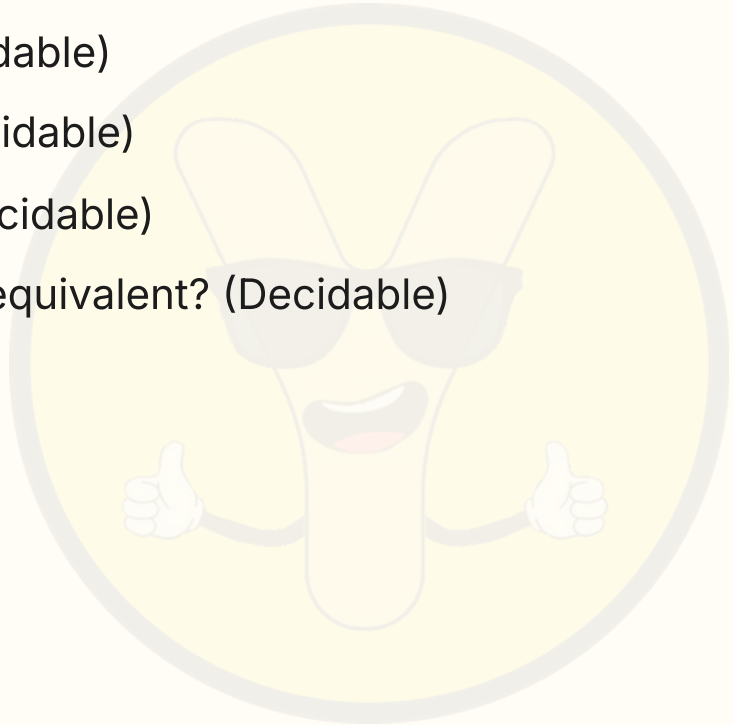
1. Assume L is regular. Let p be the pumping length.
2. Choose string s = a^p b^p. Split s = xyz such that |xy| ≤ p and |y| > 0.
3. Since |xy| ≤ p, y must consist only of 'a's.
4. Pump y: let i=2. The new string xy^2z has more 'a's than 'b's.
5. The resulting string is not in L, contradicting the assumption.

L is not regular.

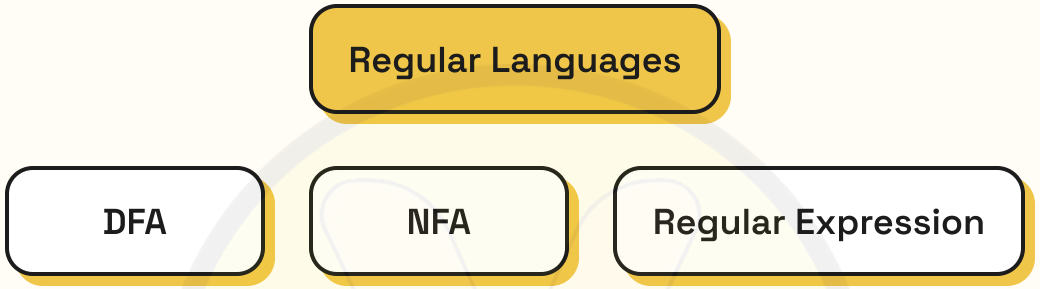


Decidability and Decision Properties

- Decidability: A problem is decidable if there exists an algorithm that provides a Yes/No answer in finite time.
- Emptiness Problem: Is $L = \varphi$? (Decidable)
- Finiteness Problem: Is L finite? (Decidable)
- Membership Problem: Is $w \in L$? (Decidable)
- Equivalence Problem: Are two FAs equivalent? (Decidable)



Finite Automata and Regular Languages Hierarchy



All these models are equivalent in power for defining Regular Languages.

Exam corner

- **Q1** State and prove Kleene's Theorem.
- **Q2** Explain Arden's Theorem with a suitable example.
- **Q3** Prove that the language L = {0^n 1^n | n ≥ 1} is not regular using Pumping Lemma.
- **Q4** List and explain the closure properties of regular languages.
- **Q5** Differentiate between DFA, NFA, and Transition Graphs.
- **Q6** Define the Pumping Lemma and its significance in automata theory.
- **Q7** Discuss the decision properties of regular languages.

Unit 2 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 3 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

