CPSC-406 Report

Ethan Tapia
Chapman University

February 17, 2025

Abstract

Contents

1 Introduction

2 Week by Week

2.1 Week 1

Lecture Summary

A finite automaton consists of a finite set of **states** (Q), an **alphabet** (Σ), a **transition function** (δ), a **starting state** (q_0), and a set of **accepting states** (F).

It can be formally represented as:

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

where:

- Q is the set of states,
- Σ is the input alphabet,
- $\delta : Q \times \Sigma \to Q$ is the transition function,
- $q_0 \in Q$ is the initial state,

- $F \subseteq Q$ is the set of accepting states.

2.2 Week 2

Homework 1

Problem 1: Characterizing Accepted Sequences

The given problem involves designing a finite automaton that accepts sequences of 5 and 10-cent inputs summing to 25 cents.

Solution: We define the equation:

$$5a + 10b = 25 \tag{1}$$

where a is the number of 5-cent inputs and b is the number of 10-cent inputs. Solving for valid pairs:

- $(a = 5, b = 0) \Rightarrow$ Sequence: $5, 5, 5, 5, 5$
- $(a = 3, b = 1) \Rightarrow$ Sequence: $5, 5, 5, 10$
- $(a = 1, b = 2) \Rightarrow$ Sequence: $5, 10, 10$

These sequences are precisely those accepted by the automaton. The machine accepts a sequence if the total sum equals 25 cents.

Problem 2: Defining Valid Variable Names

A valid variable name must begin with a letter (ℓ) and be followed by any number of letters or digits (d).

Regular Expression:

$$\ell(\ell|d)^* \tag{2}$$

Solution: *Finite Automaton:* - States: q_0 (initial), q_1 (accepting). - Transitions: - $q_0 \to q_1$ on input ℓ - $q_1 \to q_1$ on input ℓ or d

Problem 3: Classification of Words in L_1, L_2, L_3

The given languages are defined as follows:

- $L_1 = \{x0y \mid x, y \in \Sigma^*\}$: The set of words that contain at least one '0'.
- $L_2 = \{w \mid |w| = 2^n \text{ for some } n \in \mathbb{N}\}$: The set of words whose length is a power of 2.
- $L_3 = \{w \mid |w|_0 = |w|_1\}$: The set of words where the number of 0s equals the number of 1s.

Solution: We analyze each word based on these conditions:

	L_1	L_2	L_3
$w_1 = 10011$	✓		
$w_2 = 100$	✓		
$w_3 = 10100100$	✓	✓	
$w_4 = 1010011100$	✓		✓
$w_5 = 11110000$	✓	✓	✓

Table 1: Classification of words into L_1, L_2, L_3

Problem 4: DFA Analysis

Given the DFA with states q_0 (start), q_2, and q_1 (accepting), we determine which words end in the accepting state q_1.

Transitions:

$$\begin{array}{ll} \delta(q_0, 1) = q_0, & \delta(q_0, 0) = q_2 \\ \delta(q_2, 0) = q_2, & \delta(q_2, 1) = q_1 \\ \delta(q_1, 0) = q_1, & \delta(q_1, 1) = q_1 \end{array}$$

Checking Words:

- $w_1 = 0010$: $q_0 \to q_2 \to q_2 \to q_1 \to q_1$ ✓(Accepted)
- $w_2 = 1101$: $q_0 \to q_0 \to q_0 \to q_2 \to q_1$ ✓(Accepted)
- $w_3 = 1100$: $q_0 \to q_0 \to q_0 \to q_2 \to q_2$ (Not Accepted)

Solution:

$w_1 = 0010 \to$ ✓*Accepted*
$\mathrm{w}_2 = 1101 \to$ ✓*Accepted*
$\mathrm{w}_3 = 1100 \to$ *Rejected*

This confirms that w_1 and w_2 end in the accepting state, while w_3 does not.

Chapter 2.1 Report:

Chapter 2.1 discusses the use of finite automata in modeling real-world protocols, particularly in the context of electronic money transactions. The section introduces a three-party system involving a customer, a store, and a bank. The goal is to ensure that digital money is not duplicated or reused fraudulently.

The protocol consists of five primary actions: pay, cancel, ship, redeem, and transfer. Each party's behavior is modeled using finite automata to track transaction states. The section highlights how such models can reveal vulnerabilities—such as a store shipping goods before verifying payment—showcasing the importance of automata in validating protocol security.

Finite automata prove to be useful for detecting logical flaws in transaction systems, ensuring valid sequences of operations. The chapter serves as an introduction to the application of formal computational models in the security and validation of protocols.

3 Synthesis

4 Evidence of Participation

5 Conclusion

References

[BLA] Author, Title, Publisher, Year.