CPSC-406 Report

Your Name
Chapman University

February 11, 2026

Abstract

Contents

1 Introduction

2 Week by Week

2.1 Week 1

Exercise 1

1. Word acceptance table

w	accepted by A_1?	accepted by A_2?
aaa	No	Yes
aab	Yes	No
aba	No	No
abb	No	No
baa	No	Yes
bab	No	No
bba	No	No
bbb	No	No

2. Description of the languages

Language of A_1. In A_1, any word beginning with b immediately goes to a trap state. If the word begins with a, the automaton stays in state 2 while reading a's, and moves to the accepting state 3 upon reading a single b. Any further symbol leaves the accepting state.

Therefore, a word is accepted if and only if it consists of one or more a's followed by exactly one b.

$$L(A_1) = \{a^n b \mid n \geq 1\}$$

Equivalently, as a regular expression:

$$L(A_1) = a^+ b$$

Language of A_2. In A_2, the only accepting state is 3. The automaton reaches state 3 precisely after reading two consecutive a's. Any b resets the automaton to state 1.

Thus, a word is accepted if and only if it ends with at least two consecutive a's.

$$L(A_2) = \{w \in \{a,b\}^* \mid w \text{ ends with } aa\}$$

Equivalently, as a regular expression:

$$L(A_2) = (a|b)^* aa$$

Exercise 2

Alphabet: $\Sigma = \{a, b\}$.

1. DFAs for the given languages

(1) All words that end with `ab`. Let $M_1 = (Q, \Sigma, \delta, q_0, F)$ where

$$Q = \{q_0, q_1, q_2\}, \quad q_0 \text{ start}, \quad F = \{q_2\}.$$

Transitions:

	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_0

State q_2 represents that the word currently ends in `ab`.

(2) All words that contain `aba`. Let $M_2 = (Q, \Sigma, \delta, q_0, F)$ where

$$Q = \{q_0, q_1, q_2, q_3\}, \quad q_0 \text{ start}, \quad F = \{q_3\}.$$

Transitions:

	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_3	q_0
q_3	q_3	q_3

State q_3 means the substring `aba` has been seen.

(3) Odd number of `a`'s and odd number of `b`'s. Use four states representing parity:

$$Q = \{(E, E), (E, O), (O, E), (O, O)\}$$

(start state (E, E)). Accepting state: (O, O).

Reading a toggles the first component; reading b toggles the second component.

(4) Even number of `a`'s and odd number of `b`'s. Same construction as (3).

Start state: (E, E). Accepting state: (E, O).

Transitions again toggle parity accordingly.

(5) Any three consecutive characters contain at least one `a`. This is equivalent to saying the word does *not* contain `bbb`.

Use states counting consecutive b's:

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

- q_0: no recent b
- q_1: one consecutive b
- q_2: two consecutive b's
- q_3: three consecutive b's (trap)

Start: q_0.

Accepting states: $\{q_0, q_1, q_2\}$.

Transitions:

- On a: go to q_0 from any state except q_3 - On b:

$$q_0 \to q_1, \quad q_1 \to q_2, \quad q_2 \to q_3, \quad q_3 \to q_3$$

(6) All words that contain `bbb`. Same structure as (5), but now:

Accepting state: $\{q_3\}$ only.

Once q_3 is reached, stay there.

2. Observations

Several patterns appear:

- Languages involving substrings (e.g. `aba`, `bbb`) use states that track partial progress toward matching the substring.
- Parity conditions use a product construction (Cartesian product) of two 2-state automata, giving 4 states.
- Conditions about avoiding a pattern (problem 5) are closely related to conditions about containing that pattern (problem 6); they differ mainly in which states are accepting.
- Many DFAs follow a systematic design pattern: track exactly the minimal information necessary to decide acceptance.

3 Synthesis

4 Evidence of Participation

5 Conclusion

References

[BLA] Author, Title, Publisher, Year.