

# CPSC-406 Report

Savana Chou  
Chapman University

February 21, 2026

## Abstract

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Week by Week</b>	<b>1</b>
2.1	Week 1 . . . . .	1
2.2	Week 2 . . . . .	4
<b>3</b>	<b>Synthesis</b>	<b>8</b>
<b>4</b>	<b>Evidence of Participation</b>	<b>8</b>
<b>5</b>	<b>Conclusion</b>	<b>8</b>

## 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  $b$ :

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

**(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.

### Week 1 Discord Question

Why is it important that the transition function of a DFA be total (defined for every state-symbol pair)? What would break, both formally and conceptually, if we allowed “missing” transitions?

## 2.2 Week 2

### Exercise 1: Product Automata

Let  $\Sigma = \{a, b\}$ .

#### 1. Description of $L(\mathcal{A}^{(1)})$ and $L(\mathcal{A}^{(2)})$

##### Language of $\mathcal{A}^{(1)}$

In  $\mathcal{A}^{(1)}$ , states 2 and 4 are accepting, and state 3 is a trap state with self-loops on both  $a$  and  $b$ . From the transition structure we observe:

- From state 2, reading  $a$  leads to the trap state.
- From state 4, reading  $b$  leads to the trap state.

Thus any occurrence of the substring aa or bb forces the automaton into the trap state.

Therefore,

$$L(\mathcal{A}^{(1)}) = \{w \in \{a, b\}^* \mid w \text{ contains no } aa \text{ and no } bb\}.$$

Equivalently,  $L(\mathcal{A}^{(1)})$  consists of all strings whose symbols strictly alternate.

### Language of $\mathcal{A}^{(2)}$

In  $\mathcal{A}^{(2)}$ , state 2 is the only accepting state and state 3 is a trap.

- From the start state 1, reading  $b$  leads immediately to the trap.
- Thus every accepted string must begin with  $a$ .
- The automaton alternates between states 1 and 2 on every input symbol.

Since state 2 is accepting, the automaton accepts precisely those strings that:

- start with  $a$ , and
- have odd length.

Hence,

$$L(\mathcal{A}^{(2)}) = \{w \in \{a, b\}^* \mid w \text{ starts with } a \text{ and } |w| \text{ is odd}\}.$$

## 2. Construction of the Intersection Automaton

We construct the product automaton

$$\mathcal{A} = \mathcal{A}^{(1)} \times \mathcal{A}^{(2)}.$$

Its components are:

$$\begin{aligned} Q &= Q_1 \times Q_2, \\ q_0 &= (1, 1), \\ F &= F_1 \times F_2. \end{aligned}$$

Since  $F_1 = \{2, 4\}$  and  $F_2 = \{2\}$ , we have

$$F = \{(2, 2), (4, 2)\}.$$

The transition function is defined by

$$\delta((p, q), x) = (\delta_1(p, x), \delta_2(q, x)) \quad \text{for } x \in \{a, b\}.$$

Keeping only reachable states, the automaton includes:

$$(1, 1), (2, 2), (4, 1), (3, 1), (4, 3), (3, 3).$$

Among these, the only reachable accepting state is

$$(2, 2).$$

### 3. Correctness of the Product Construction

By definition of the product construction, a string  $w$  is accepted by  $\mathcal{A}$  if and only if:

- the run of  $w$  in  $\mathcal{A}^{(1)}$  ends in an accepting state, and
- the run of  $w$  in  $\mathcal{A}^{(2)}$  ends in an accepting state.

Thus,

$$L(\mathcal{A}) = L(\mathcal{A}^{(1)}) \cap L(\mathcal{A}^{(2)}).$$

### 4. Construction for the Union

To obtain an automaton  $\mathcal{A}'$  such that

$$L(\mathcal{A}') = L(\mathcal{A}^{(1)}) \cup L(\mathcal{A}^{(2)}),$$

we keep:

- the same state set  $Q_1 \times Q_2$ ,
- the same start state  $(1, 1)$ ,
- the same transition function.

We change only the accepting states to:

$$F' = (F_1 \times Q_2) \cup (Q_1 \times F_2).$$

That is, a state  $(p, q)$  is accepting if either

$$p \in F_1 \quad \text{or} \quad q \in F_2.$$

## Exercise 2: More Automata

Let  $\Sigma = \{a, b\}$ .

### 1. Description of the Languages

#### Language of $\mathcal{B}^{(1)}$

The automaton  $\mathcal{B}^{(1)}$  has states  $\{p_0, p_1, p_2\}$  where  $p_0$  is the start and only accepting state. Each  $b$  leaves the state unchanged, while each  $a$  moves cyclically:

$$p_0 \xrightarrow{a} p_1 \xrightarrow{a} p_2 \xrightarrow{a} p_0.$$

Thus the automaton counts the number of  $a$ 's modulo 3. Since only  $p_0$  is accepting, we obtain

$$L(\mathcal{B}^{(1)}) = \{ w \in \{a, b\}^* \mid \#_a(w) \equiv 0 \pmod{3} \}.$$

In words: all strings whose number of  $a$ 's is divisible by 3.

### Language of $\mathcal{B}^{(2)}$

The automaton  $\mathcal{B}^{(2)}$  has states  $\{q_0, q_1, q_2\}$  where  $q_0$  is the start and only accepting state, and  $q_2$  is a trap state with loops on both  $a$  and  $b$ .

The transitions show:

$$q_0 \xrightarrow{a} q_1, \quad q_1 \xrightarrow{a} q_2.$$

Thus any occurrence of two consecutive  $a$ 's sends the automaton to the trap state  $q_2$ . Therefore,

$$L(\mathcal{B}^{(2)}) = \{ w \in \{a, b\}^* \mid w \text{ does not contain } aa \}.$$

## 2. Construction of the Intersection Automaton

We construct the product automaton

$$\mathcal{B} = \mathcal{B}^{(1)} \times \mathcal{B}^{(2)}.$$

Its components are:

$$\begin{aligned} Q &= \{p_0, p_1, p_2\} \times \{q_0, q_1, q_2\}, \\ q_0^{\mathcal{B}} &= (p_0, q_0), \\ F &= \{(p_0, q_0)\}. \end{aligned}$$

The transition function is defined by

$$\delta((p_i, q_j), x) = (\delta_1(p_i, x), \delta_2(q_j, x)) \quad \text{for } x \in \{a, b\}.$$

Since both automata must accept, the only accepting state is  $(p_0, q_0)$ .

## 3. Correctness of the Construction

By definition of the product construction, a string  $w$  is accepted by  $\mathcal{B}$  if and only if:

- the run of  $w$  in  $\mathcal{B}^{(1)}$  ends in  $p_0$ , and
- the run of  $w$  in  $\mathcal{B}^{(2)}$  ends in  $q_0$ .

Hence,

$$L(\mathcal{B}) = L(\mathcal{B}^{(1)}) \cap L(\mathcal{B}^{(2)}).$$

## 4. Construction of $\mathcal{B}'$ for the Union with a Complement

We want an automaton  $\mathcal{B}'$  such that

$$L(\mathcal{B}') = L(\mathcal{B}^{(1)}) \cup \overline{L(\mathcal{B}^{(2)})}.$$

**Step 1: Complement  $\mathcal{B}^{(2)}$ .** Since  $\mathcal{B}^{(2)}$  is a complete DFA, we obtain  $\overline{\mathcal{B}^{(2)}}$  by swapping accepting and non-accepting states. Thus  $q_1$  and  $q_2$  become accepting, and  $q_0$  becomes rejecting.

**Step 2: Product Construction for the Union.** We keep:

- the same state set  $Q_1 \times Q_2$ ,
- the same start state  $(p_0, q_0)$ ,
- the same transition function.

We change the accepting states to:

$$F' = (F_1 \times Q_2) \cup (Q_1 \times F_2).$$

That is, a state  $(p, q)$  is accepting if either

$$p = p_0 \quad \text{or} \quad q \in \{q_1, q_2\}.$$

## Week 2 Discord Question

In both exercises, we constructed product automata to recognize intersections and modified acceptance conditions to recognize unions or complements. Suppose instead we were given only one of the product automata (without being told it was constructed from two smaller DFAs). How could we determine whether this automaton can be decomposed into a product of two smaller DFAs?

## 3 Synthesis

## 4 Evidence of Participation

## 5 Conclusion

## References

[BLA] Author, [Title](#), Publisher, Year.