San José State University
Department of Computer Science

Ahmad Yazdankhah
ahmad.yazdankhah@sjsu.edu
www.cs.sjsu.edu/~yazdankhah

Deterministic Finite Automata

(Part 2)

Lecture 07

Day 07/31

CS 154

Formal Languages and Computability

Spring 2019

Agenda of Day 07

- Collecting HW1
- Summary of Lecture 06
- Quiz 2
- Lecture 07: Teaching ...
  - Deterministic Finite Automata (Part 2)

Summary of Lecture 07: We learned ...

Automata

- Formal languages are mathematical model of all languages.
- So, we'd need to construct some machines to understand these languages.
- We call these machines automata.
- Automaton is ...
- ... a mathematical model of a computing device.
- We'll construct several classes of machines in this course.

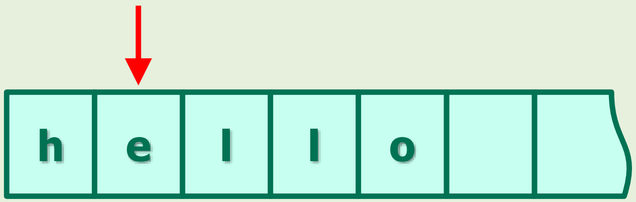
DFAs

- DFAs are the simplest ones.
- DFA stands for ...
- ... Deterministic Finite Automata
- Its building blocks contains ...
- ... Input tape, Control unit, Output

Any question?

Summary of Lecture 07: We learned ...

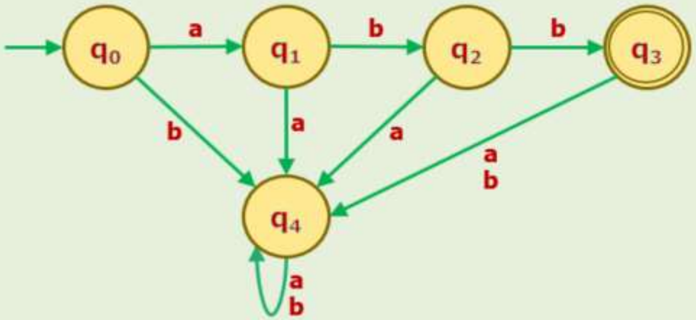
Input Tape



- The input tape is read-only.
- The read-head moves from left-to-right.
  - We cannot move the head back.
- Consuming a symbol = reading the symbol + moving the read-head to the right

Control Unit

- Its decision making part is represented by a transition graph.



- There is only one initial state.
- There can be zero or more accepting state (aka final state).
- The number of states is finite.
- That's why we call this class Deterministic Finite Automata.

Any question?

Summary of Lecture 07: We learned ...

Output

Output

Accept
or
Reject

- The output has two messages:
  - Accept
    (aka: understood, recognized, Yes)
  - Reject
    (aka: not understood, not recognized, No)

Any question?

NAME	Alan M. Turing		
SUBJECT	CS 154	TEST NO.	2
DATE	02/14/2019	PERIOD	1 / 2 / 3

TEST RECORD	
PART 1	123
PART 2	
TOTAL	

Your list # goes here!

Take-Home Exam!

Quiz 2

Use Scantron

4. How DFAs Work

4. How DFAs Work

- To understand how DFAs work, we should clearly respond to the following questions:

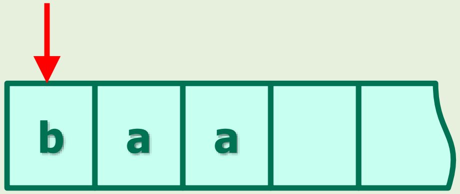
1. What is the "starting configuration"?
2. What would happen during a timeframe?
3. When would the machine halt (stop)?
4. How would a string be Accepted/Rejected?

- We'll answer the first two questions right now and postpone the last two because we need some practices first.

4.1. DFAs Starting Configuration

Clock

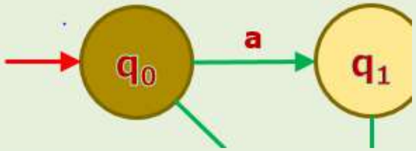
- The clock is set at timeframe 0.



Input Tape

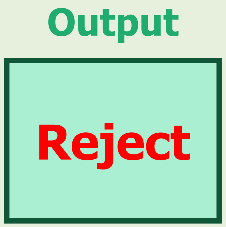
- The input string has already been written on the tape.
- The read-head is pointing to the left-most symbol.

Control Unit



- The control unit is set to initial state.

Output



- The output shows "Reject".

What is Configuration?

Definition

- DFAs' configuration is the combination of the following data:

1. Input string + Position of the read-head
2. Current state of the transition graph
3. Timeframe number on the clock

- In fact, it is a snapshot of the machine's status in one timeframe.

4.2. What Happens During a Timeframe

- During a timeframe,
  the machine "transits" (aka "moves")
  from one configuration to another.

- Several tasks happen during a timeframe.
- The combination of these tasks is called a "transition".

- Let's first visualize these tasks through some examples.
- Then, we'll summarize them in one slide.

4.2. What Happens During a Timeframe

Transition Examples

Transition Examples

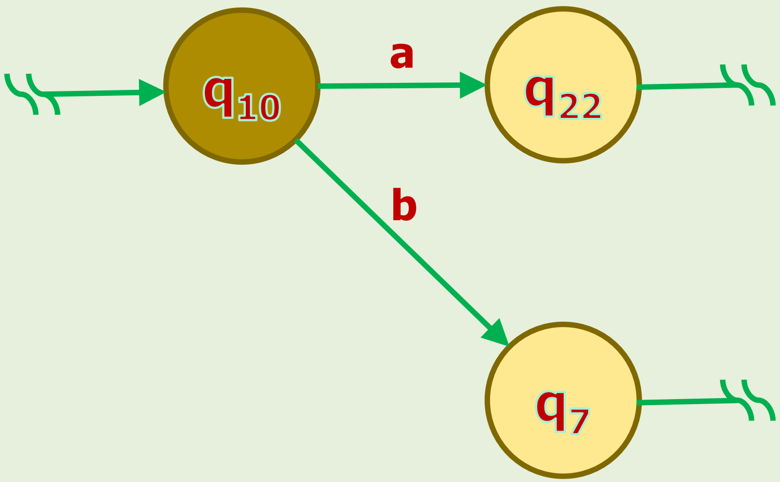
- The next examples will show:
  - a partial transition graph
  - an input tape
  - a clock

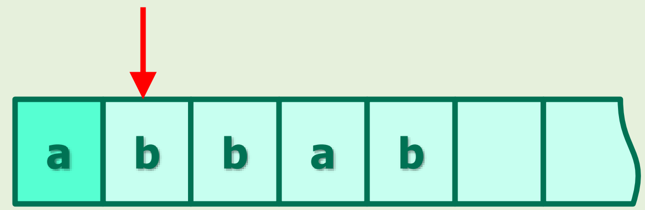
- We assume that the machine is in the middle of its operation at timeframe n.
- The question is: in what configuration would the machine be at timeframe n+1?

Transition Examples

Example 2: Timeframe 1

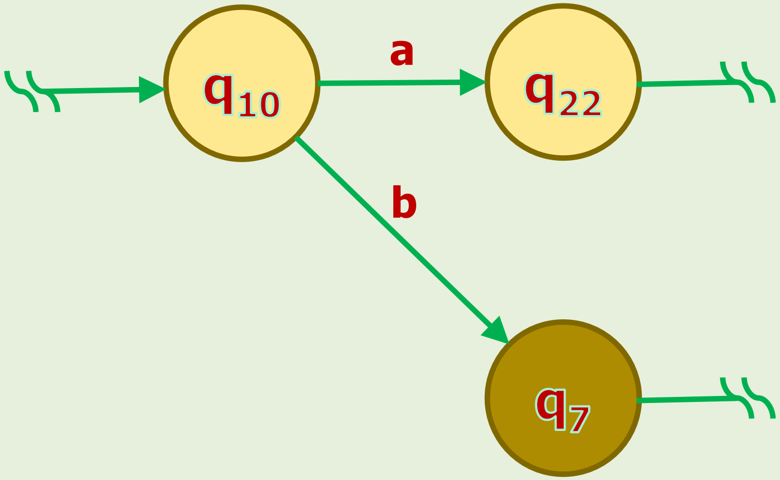
- What would be the DFA's configuration after the next timeframe?

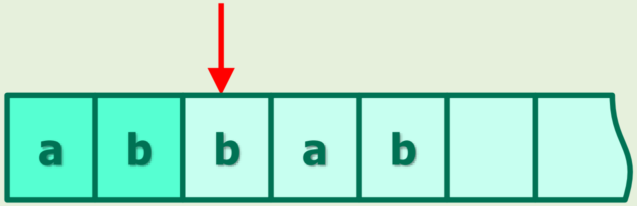




Transition Examples

Example 2: Timeframe 2

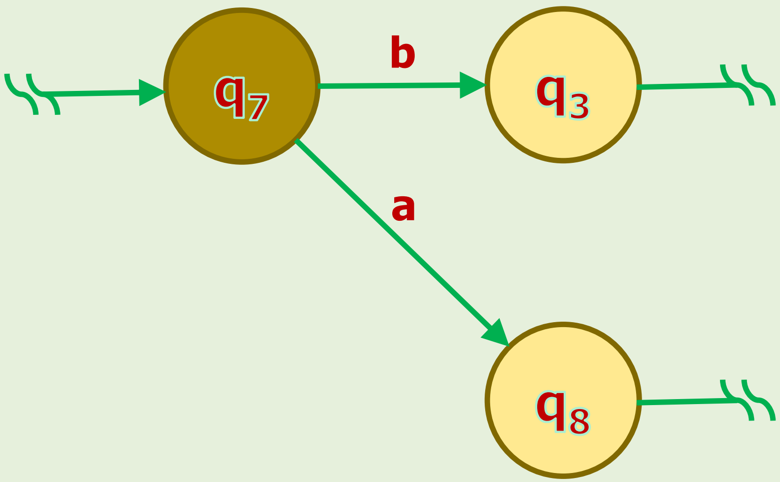


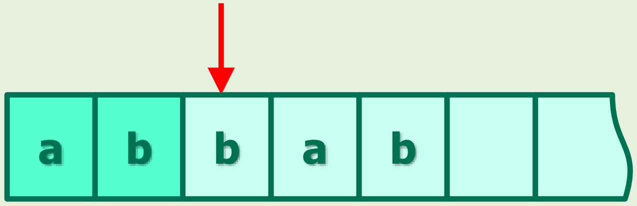


Transition Examples

Example 3: Timeframe 2

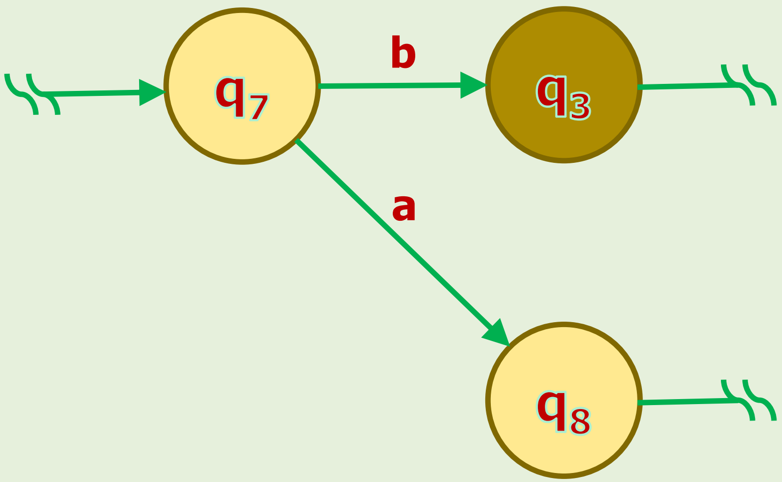
- What would be the DFA's configuration after the next timeframe?

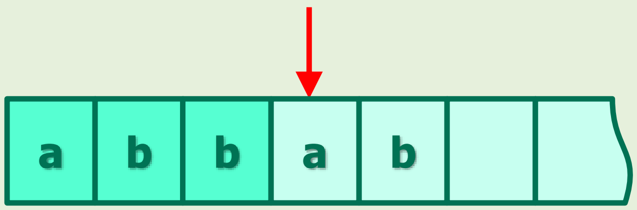




Transition Examples

Example 3: Timeframe 3

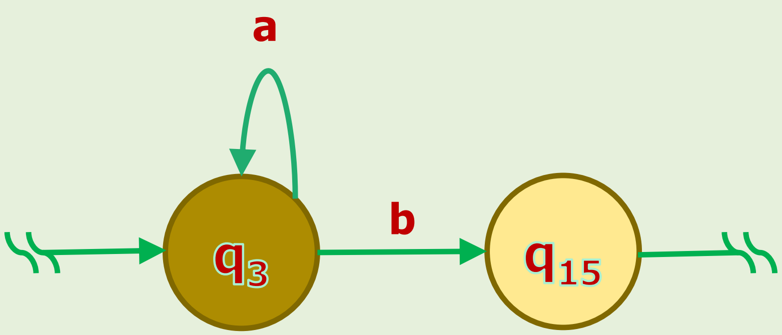


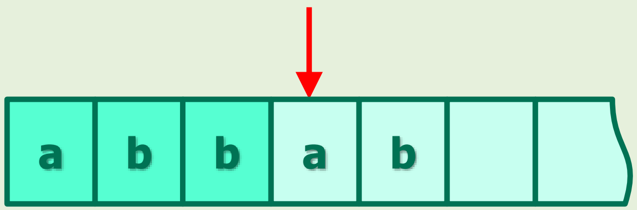


Transition Examples

Example 4: Timeframe 3

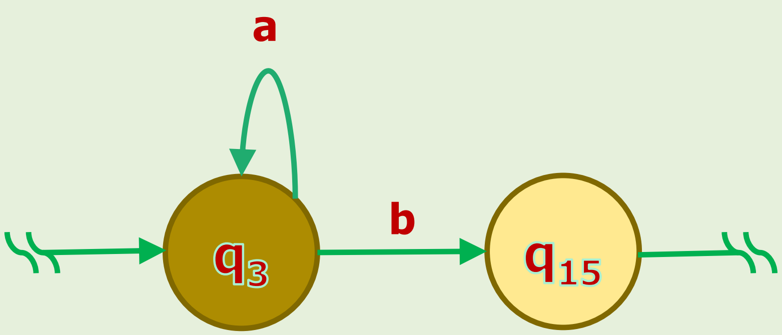
- What would be the DFA's configuration after the next timeframe?

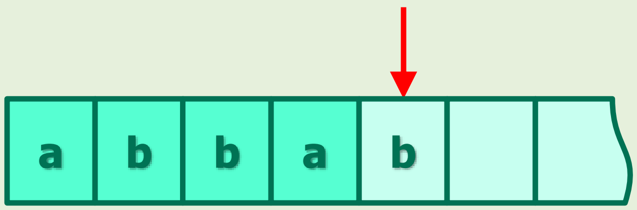




Transition Examples

Example 4: Timeframe 4





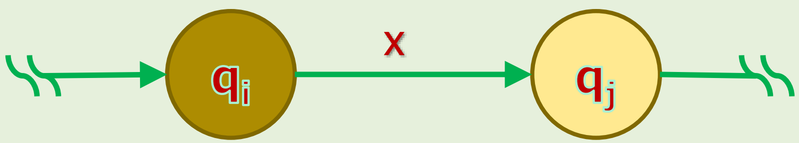
4.2. What Happens During a Timeframe

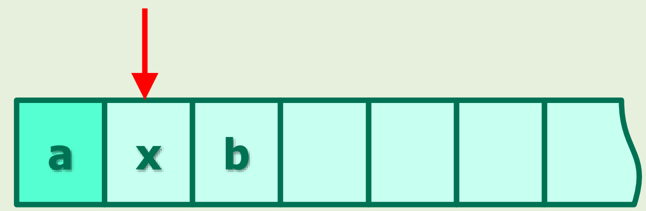
Summary of Transition

1. The symbol at which the read-head is pointing is read and is sent to the control unit.
2. The control unit makes its move based on the "logic of the transition". (See next slide!)
3. The control unit commands the tape to move the read-head one cell to the right.
   - Recall that:
     reading a symbol + moving the head = consuming a symbol

- Now let's see what the "logic of the transition" is?

DFAs' Logic of Transitions





If (Condition)

the input symbol is 'x'

Then (Operation)

transit to q_j

move the head to the right

Definition

- The logic of the transition is the "decision" that the control unit makes during every timeframe.

4. How DFAs Work

- At this moment, our knowledge is enough to work with DFAs.
- Now let's see DFAs in Action!
- We'll analyze the behavior of some DFAs.
- We'll put some strings on the tape and
  will follow the machine's behavior after each tic of the clock.

5. DFAs in Action

5. DFAs in Action

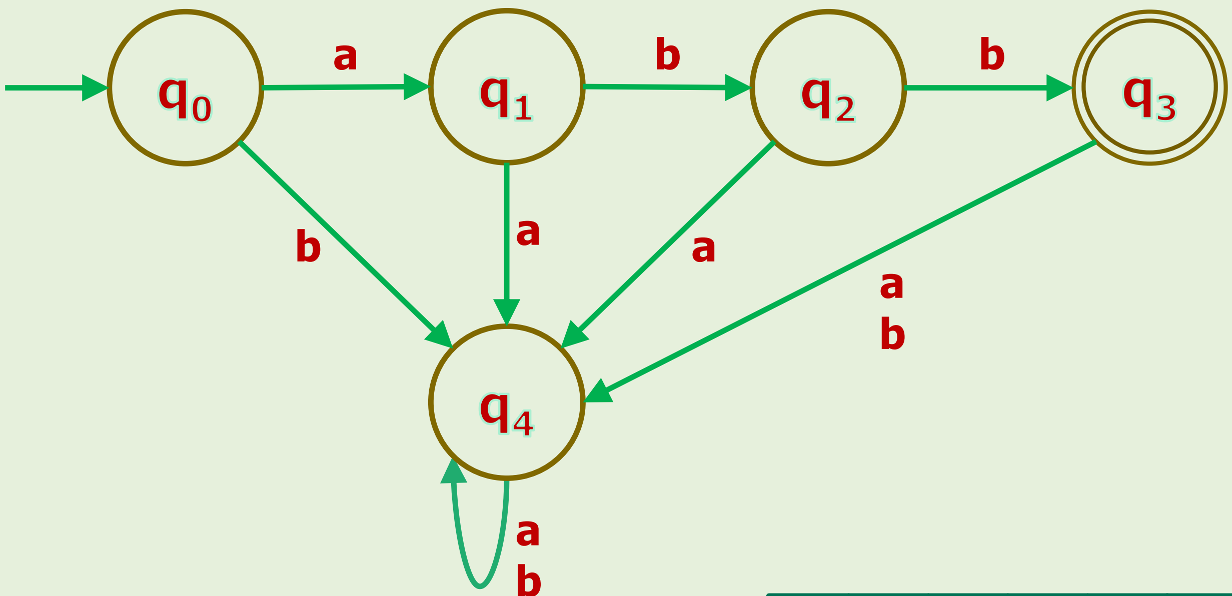
Example 5

The machine is off!

OFF

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

$w = abb$



Output

Reject

5. DFAs in Action

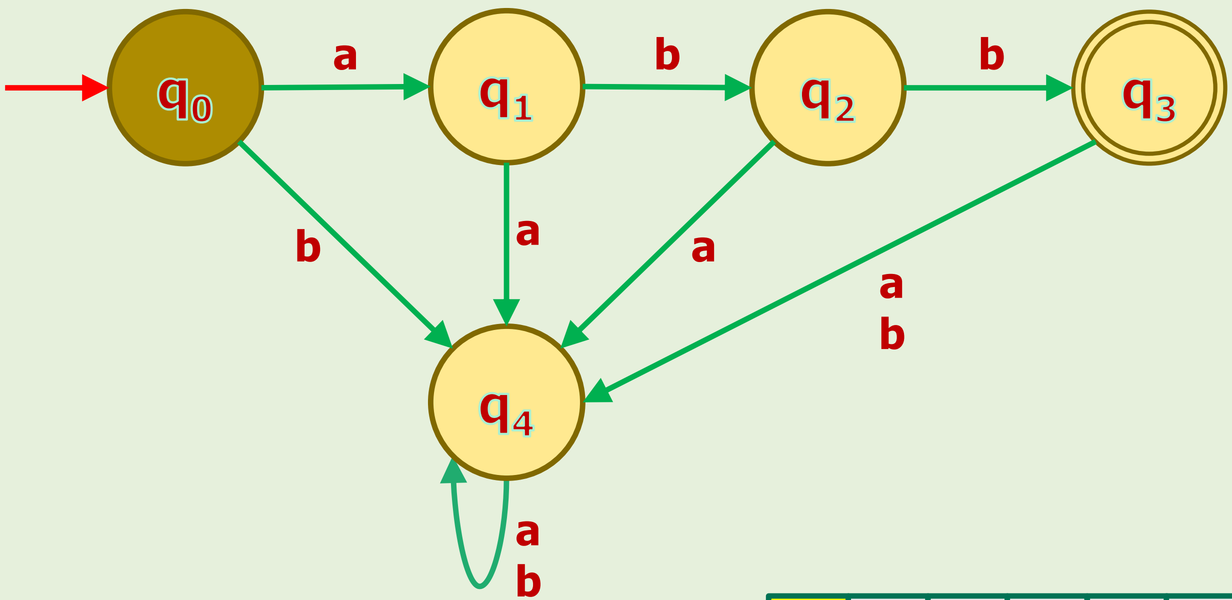
Example 5

The machine is in **starting configuration.**

ON

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

$w = abb$



Output

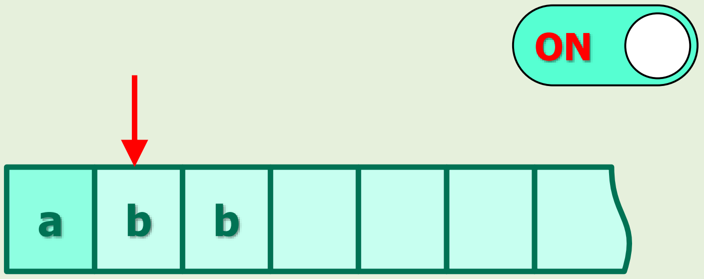
Reject

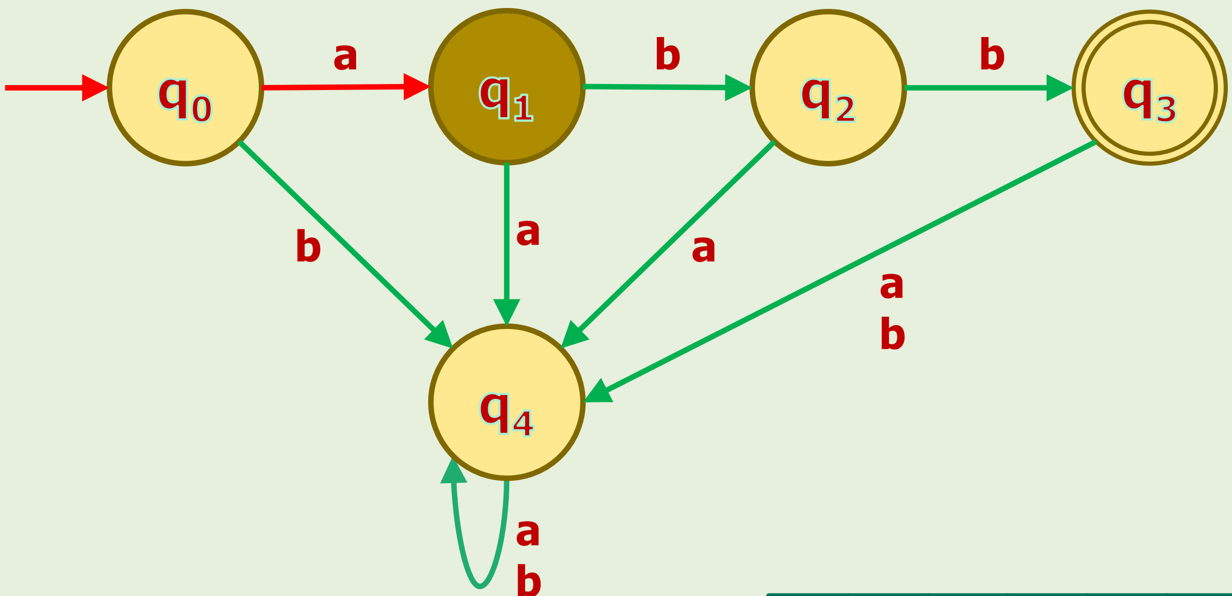
5. DFAs in Action

Example 5

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

$w = abb$





Output

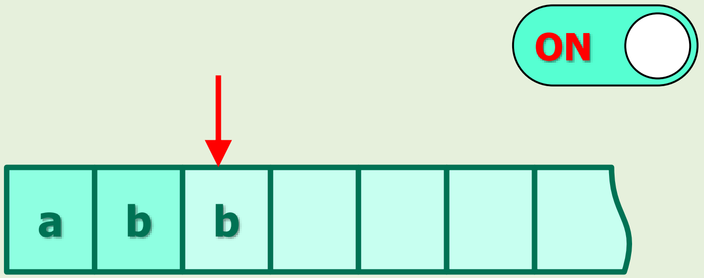
Reject

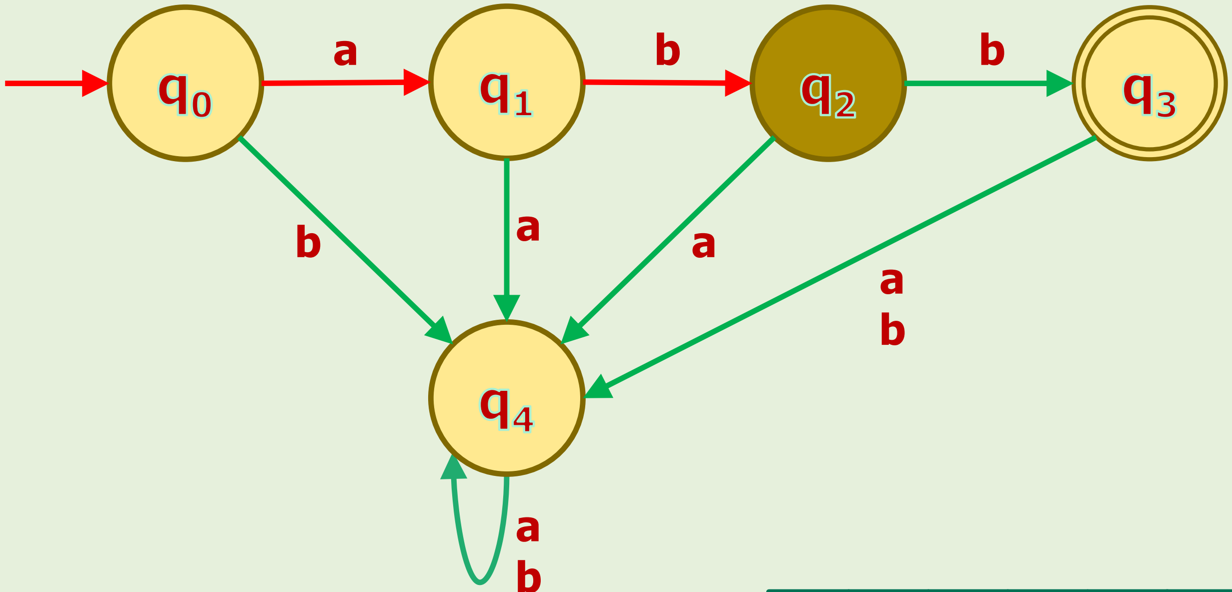
5. DFAs in Action

Example 5

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

$w = abb$





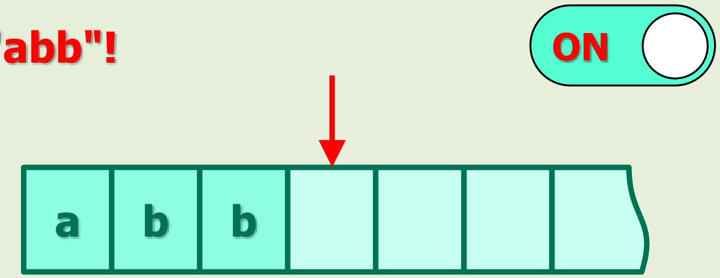
Output

Reject

5. DFAs in Action

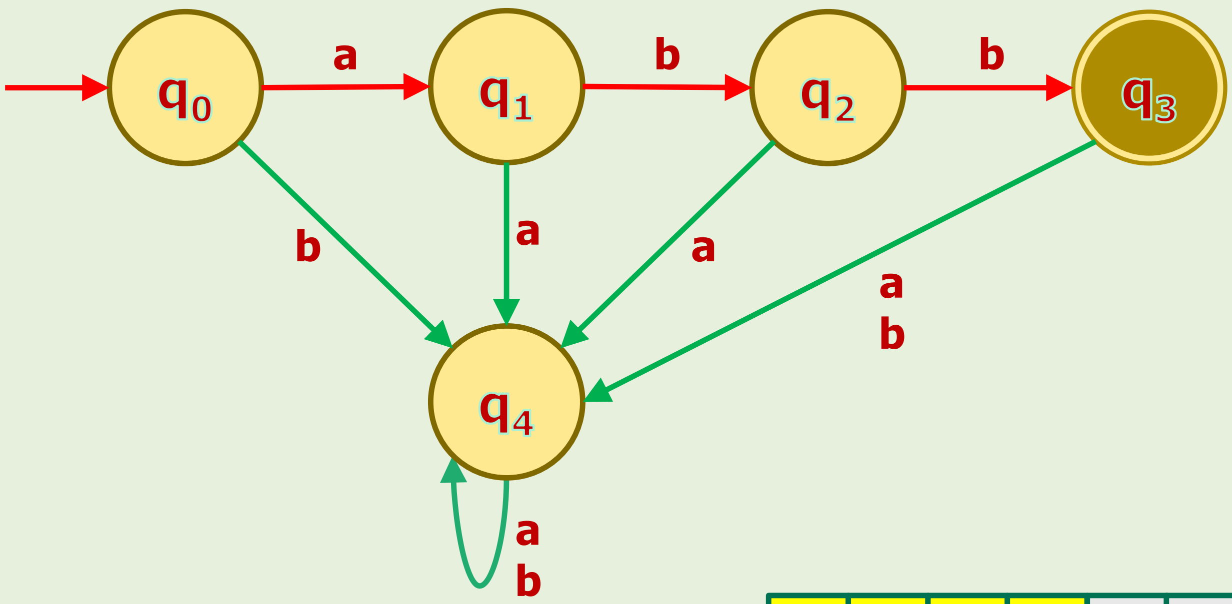
Example 5

The machine understood "abb"!



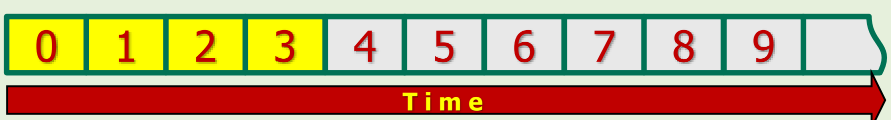
$\Sigma = \{a , b\}$

$w = abb$



Output

Accept



4.3. When DFAs Halt

- A DFA halts iff all input symbols are consumed.

- In other words, for DFAs, the following logical statement is true:
- (All input symbols are consumed.) ↔ (The DFA halts.)

Recap: Biconditional

$p \leftrightarrow q \equiv$ if and only if

Logical Representation of Halting

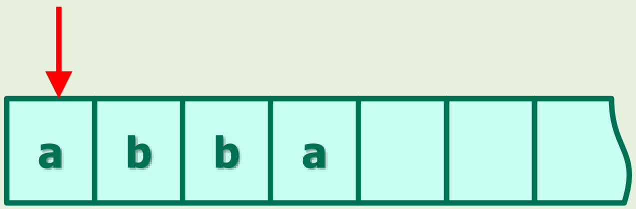
$$\left.\begin{array}{l}\text{The DFA halts.} \equiv \mathbf{h} \\ \text{IFF} \\ \text{All input symbols are consumed.} \equiv \mathbf{c}\end{array}\right\} \quad \mathbf{c} \leftrightarrow \mathbf{h}$$

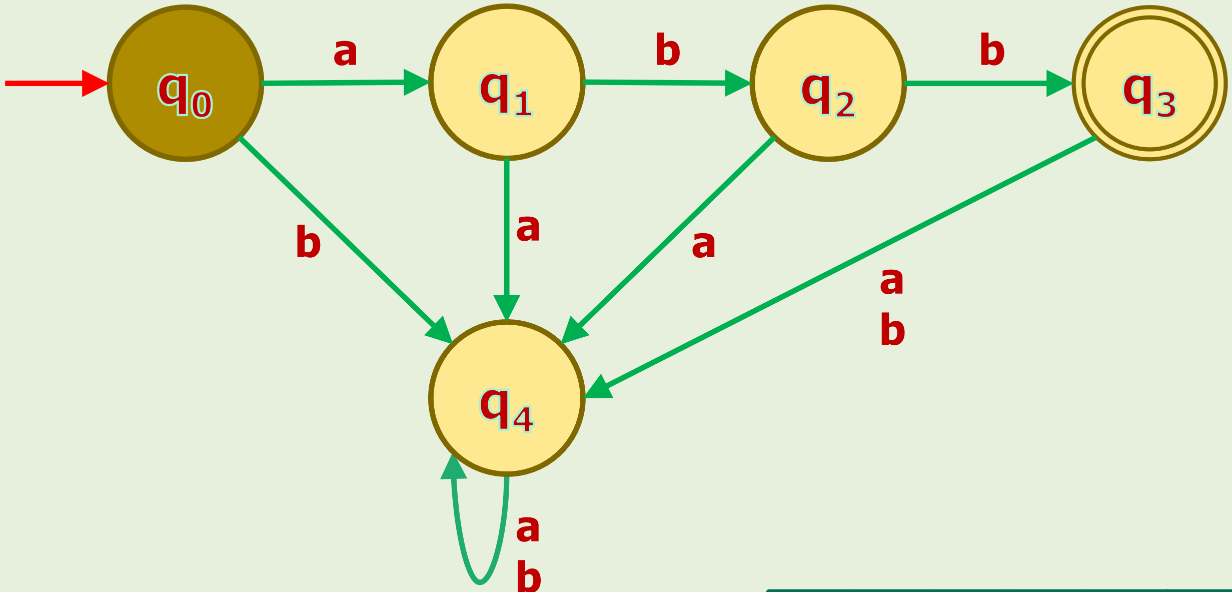
5. DFAs in Action

Example 6

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

$w = abba$





Output

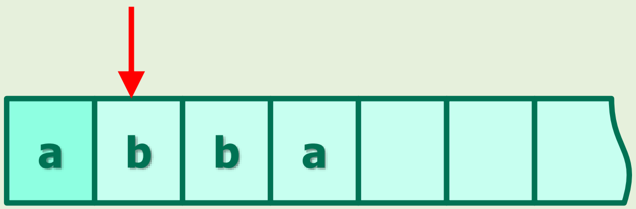
Reject

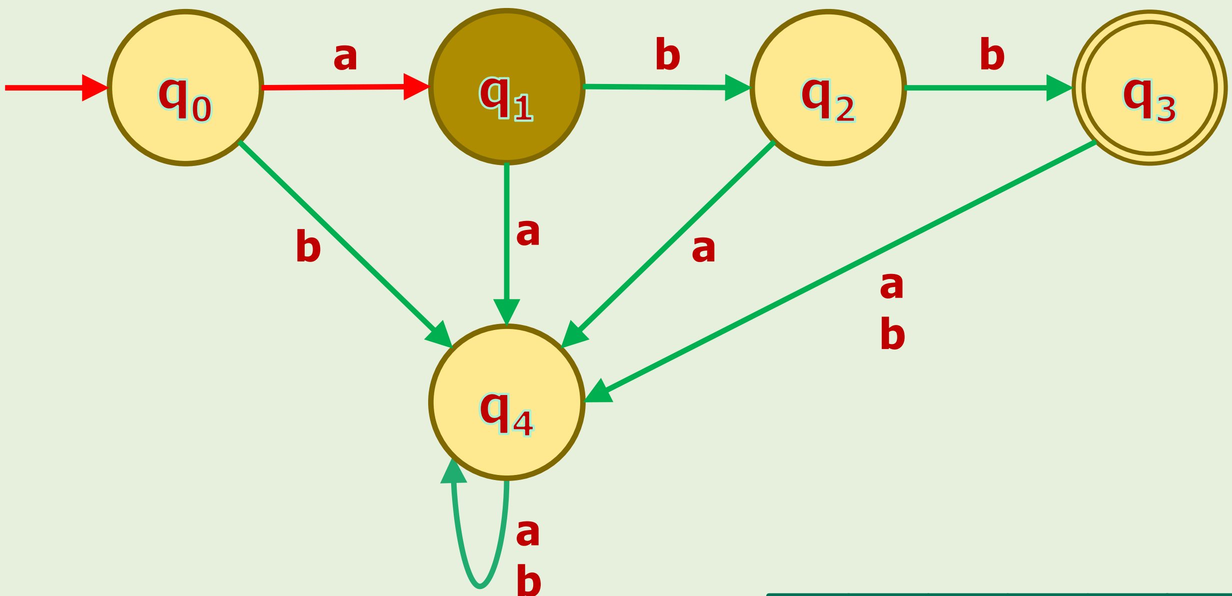
5. DFAs in Action

Example 6

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

w = abba





Output

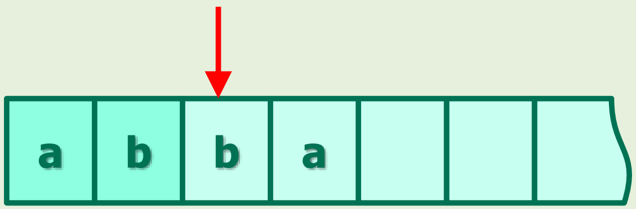
Reject

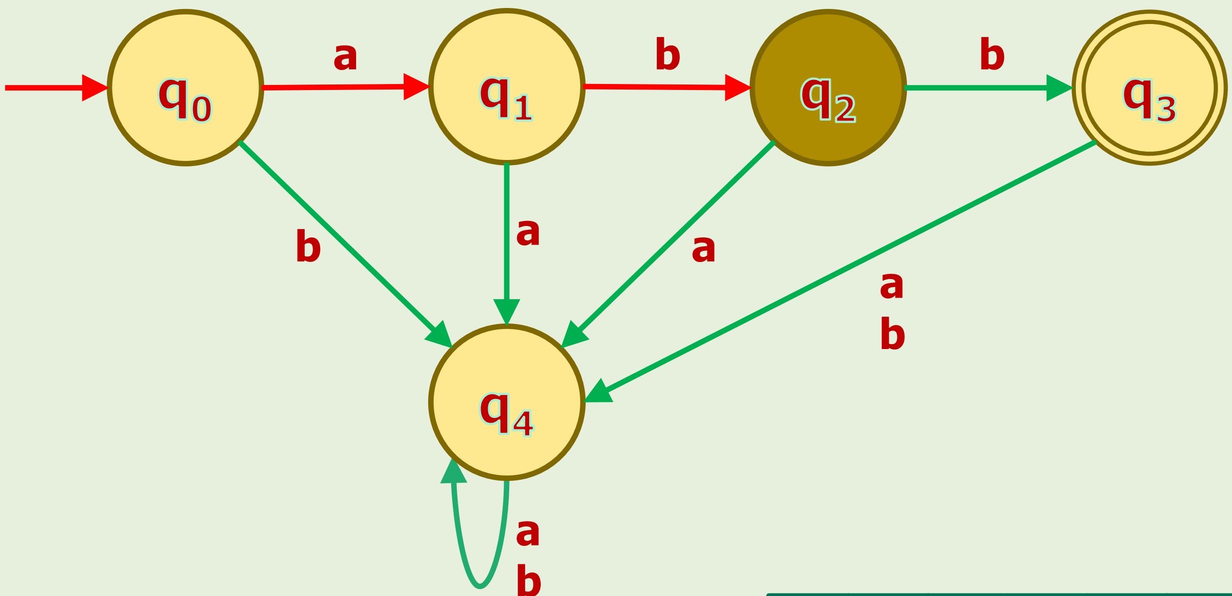
5. DFAs in Action

Example 6

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

w = abba





Output

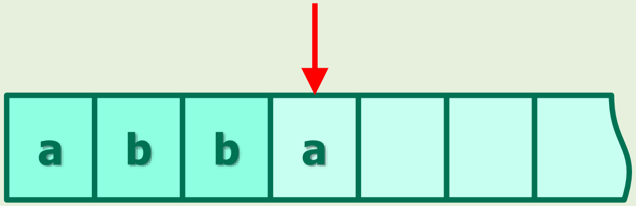
Reject

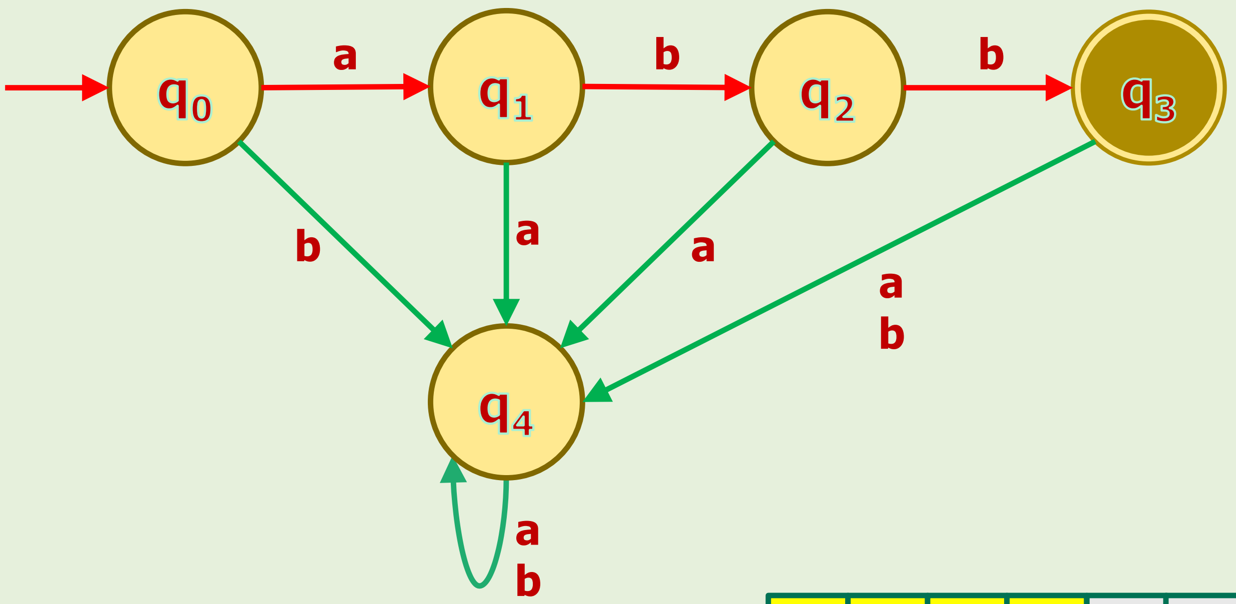
5. DFAs in Action

Example 6

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

$w = abba$





Output

Reject

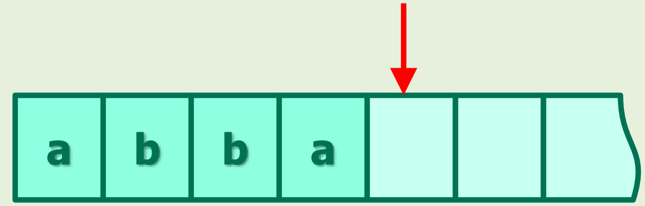
5. DFAs in Action

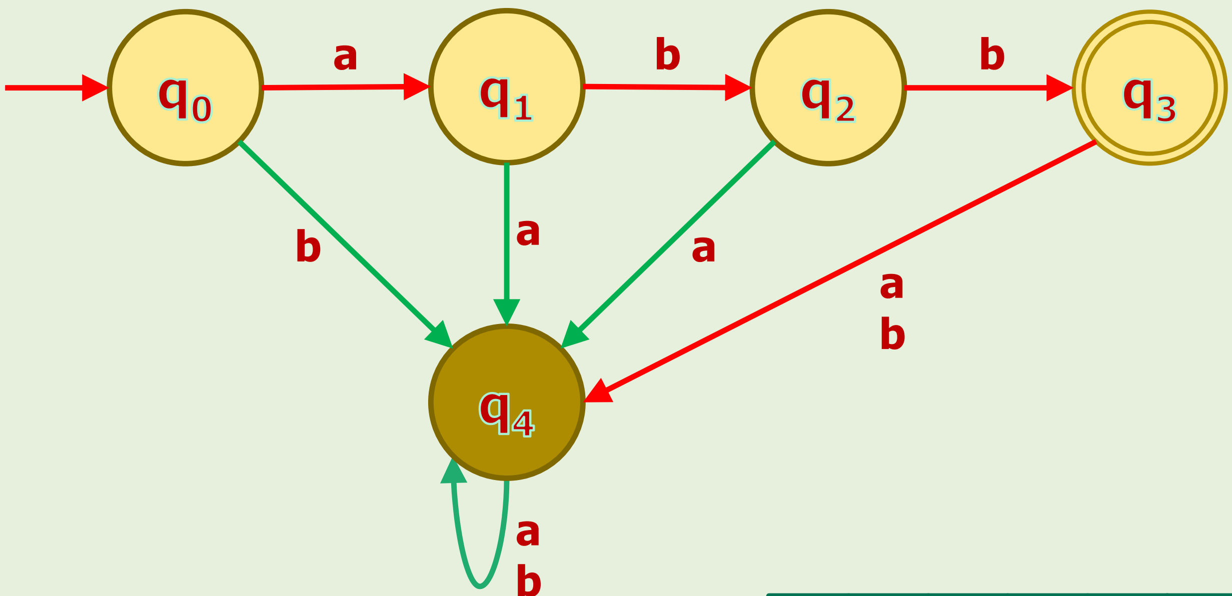
Example 6

The machine did not understand "abba"!

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

$w = abba$





Output

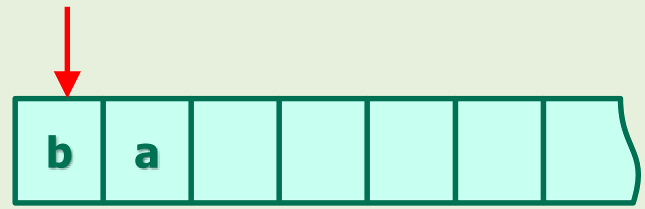
Reject

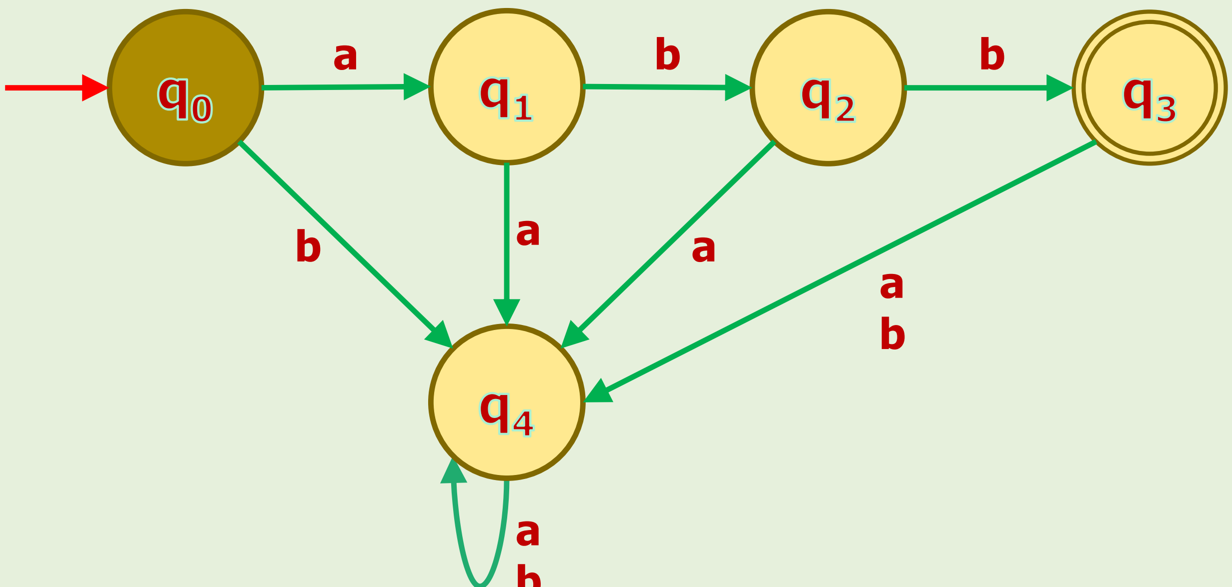
5. DFAs in Action

Example 7

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

$w = ba$





Output

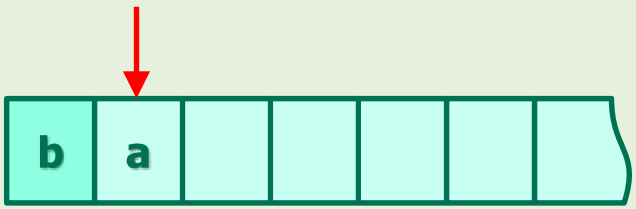
Reject

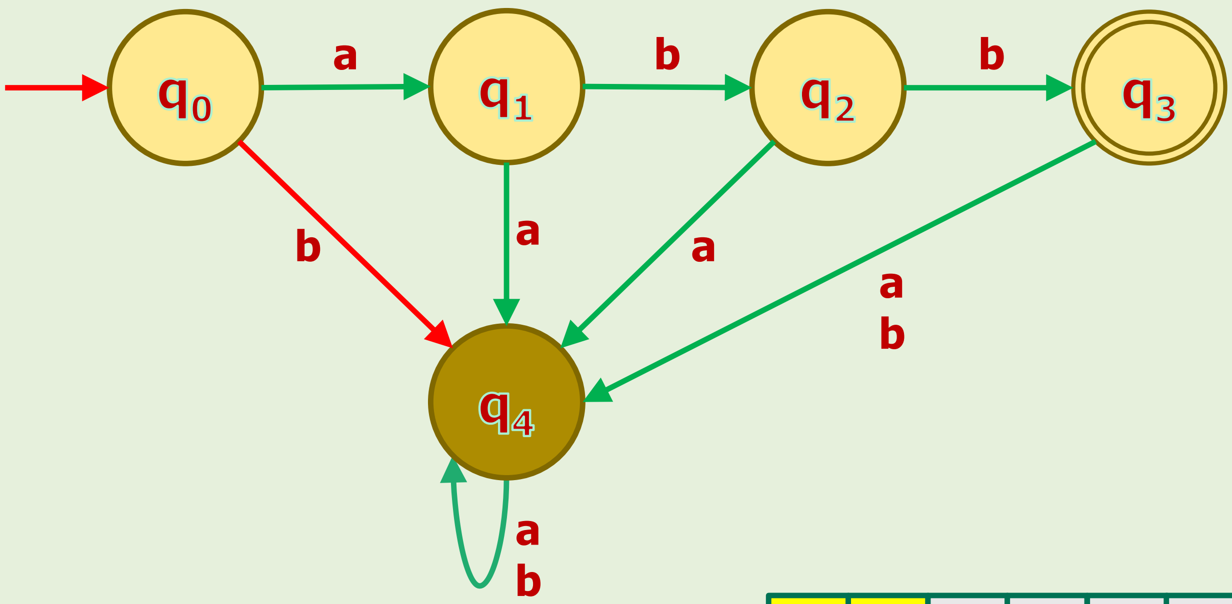
5. DFAs in Action

Example 7

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

$w = ba$





Output

Reject

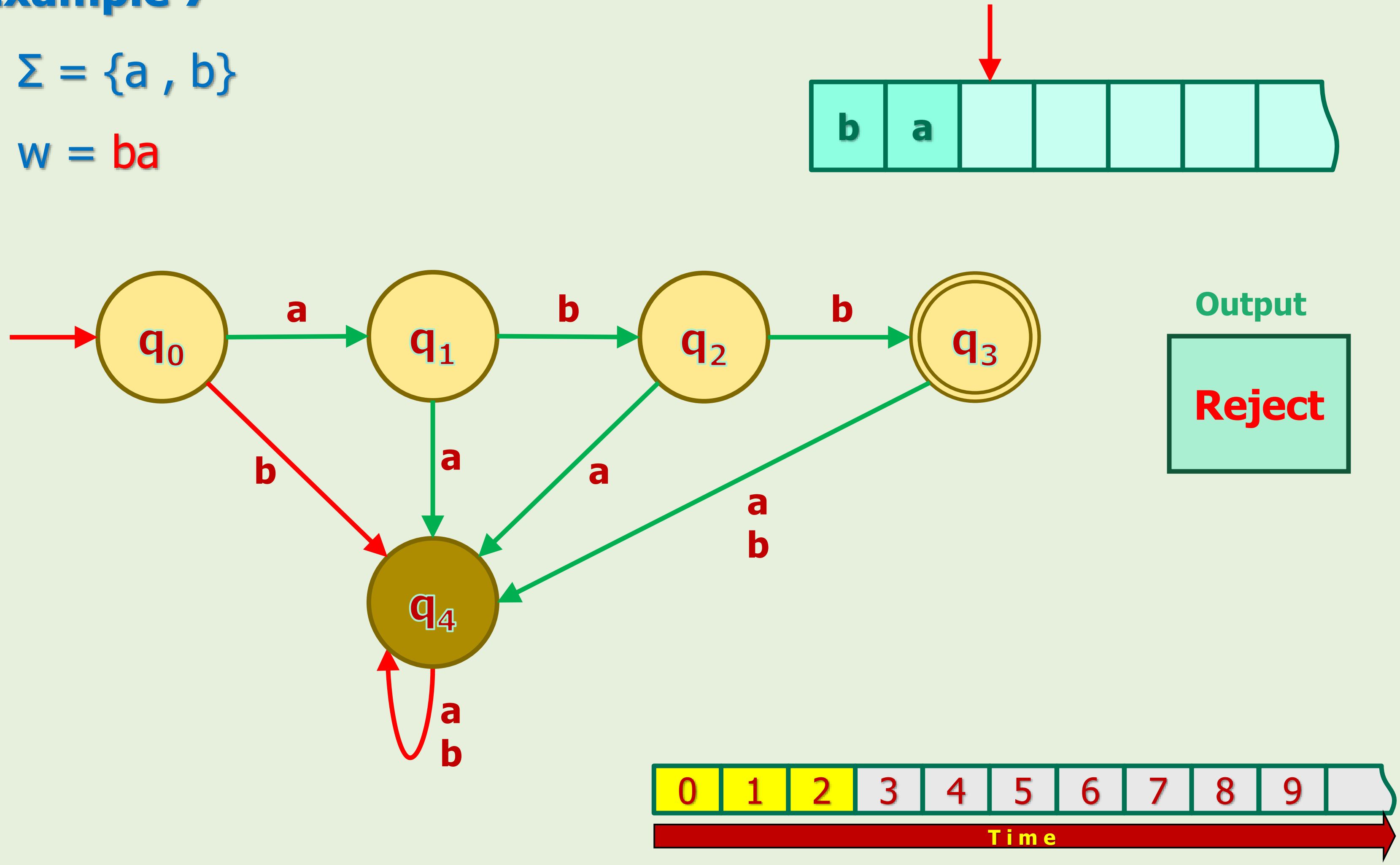
5. DFAs in Action

Example 7

The machine did not understand "ba"!

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

w = ba



5. DFAs in Action

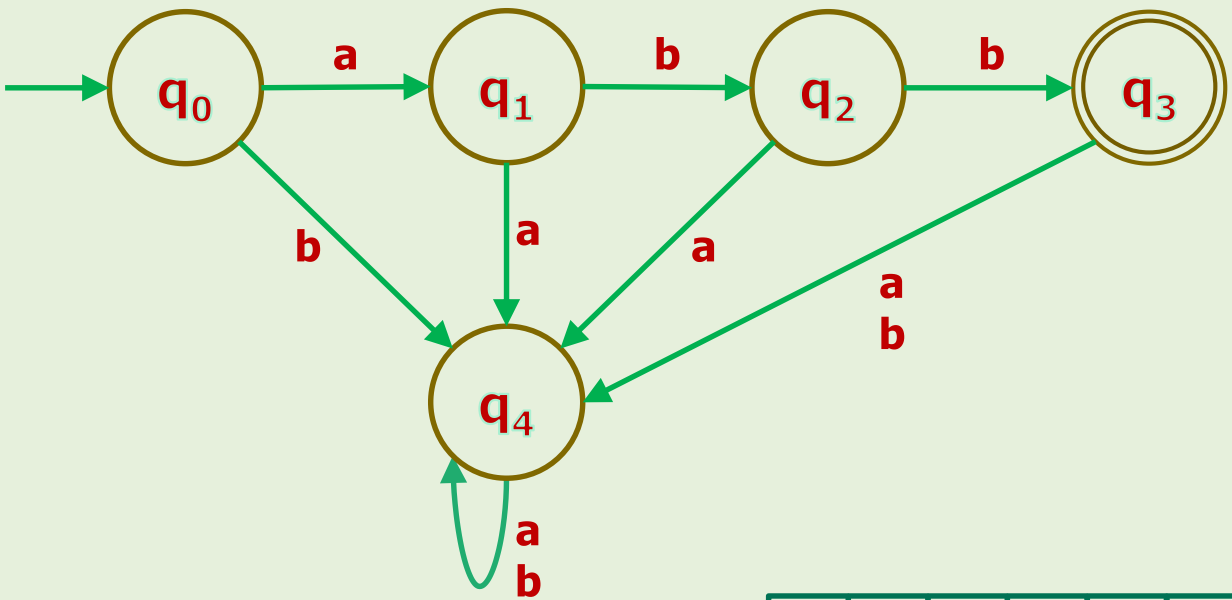
Example 8

The machine is off!

OFF

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

$w = \lambda$



Output

Reject

- How'd it work?

5. DFAs in Action

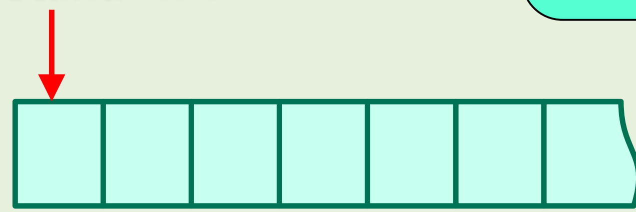
Example 8

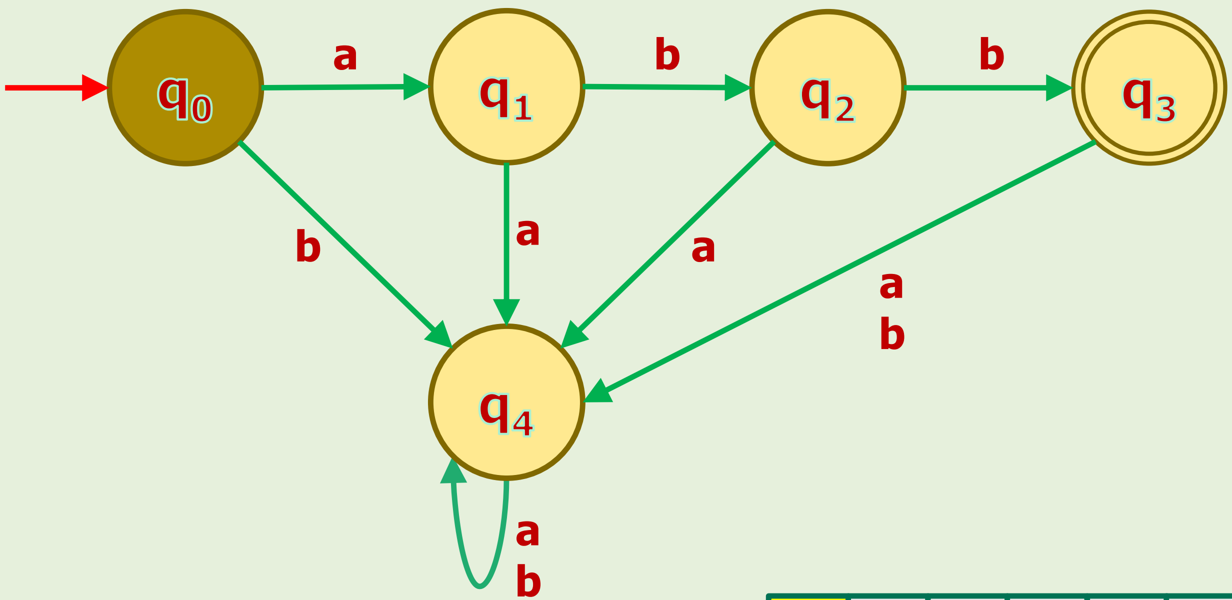
The machine did not understand "λ"!

ON

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

$w = \lambda$





Output

Reject

4.4. How DFAs Accept/Reject Strings

Logical Representation of Accepting Strings

The string w is accepted. ≡ **a**

IFF

The DFA halts. ≡ **h**

AND

All symbols of w are consumed. ≡ **c**

AND

The DFA is in an accepting (final) state. ≡ **f**

$\Bigg\}$ $(\mathbf{h} \wedge \mathbf{c} \wedge \mathbf{f}) \leftrightarrow \mathbf{a}$

- Note that, for DFAs, h and c have the same value. (both T or both F)
  - But it might NOT be true for other classes of automata.

- So, for DFAs, we can simplify the accepting logic as: $(\mathbf{c} \wedge \mathbf{f}) \leftrightarrow \mathbf{a}$

4.4. How DFAs Accept/Reject Strings

Logical Representation of Rejecting Strings

Recap: Math 42

$$p \leftrightarrow q \equiv {\sim}p \leftrightarrow {\sim}q$$

$${\sim}(p \wedge q) \equiv ({\sim}p \vee {\sim}q)$$

- We know the following equivalencies:
- Let's apply these rules to the logical representation of accepting strings:

$$(c \wedge f) \leftrightarrow a$$

$$\equiv {\sim}(c \wedge f) \leftrightarrow {\sim}a$$

- Applying DeMorgan's rule on the left side:

$$\equiv ({\sim}c \vee {\sim}f) \leftrightarrow {\sim}a$$

- What's the translation of this logical statement in plain English?

4.4. How DFAs Accept/Reject Strings

Logical Representation of Rejecting Strings

$$(\sim c \lor \sim f) \leftrightarrow \sim a$$

Translation

The string w is rejected. $\equiv \sim a$

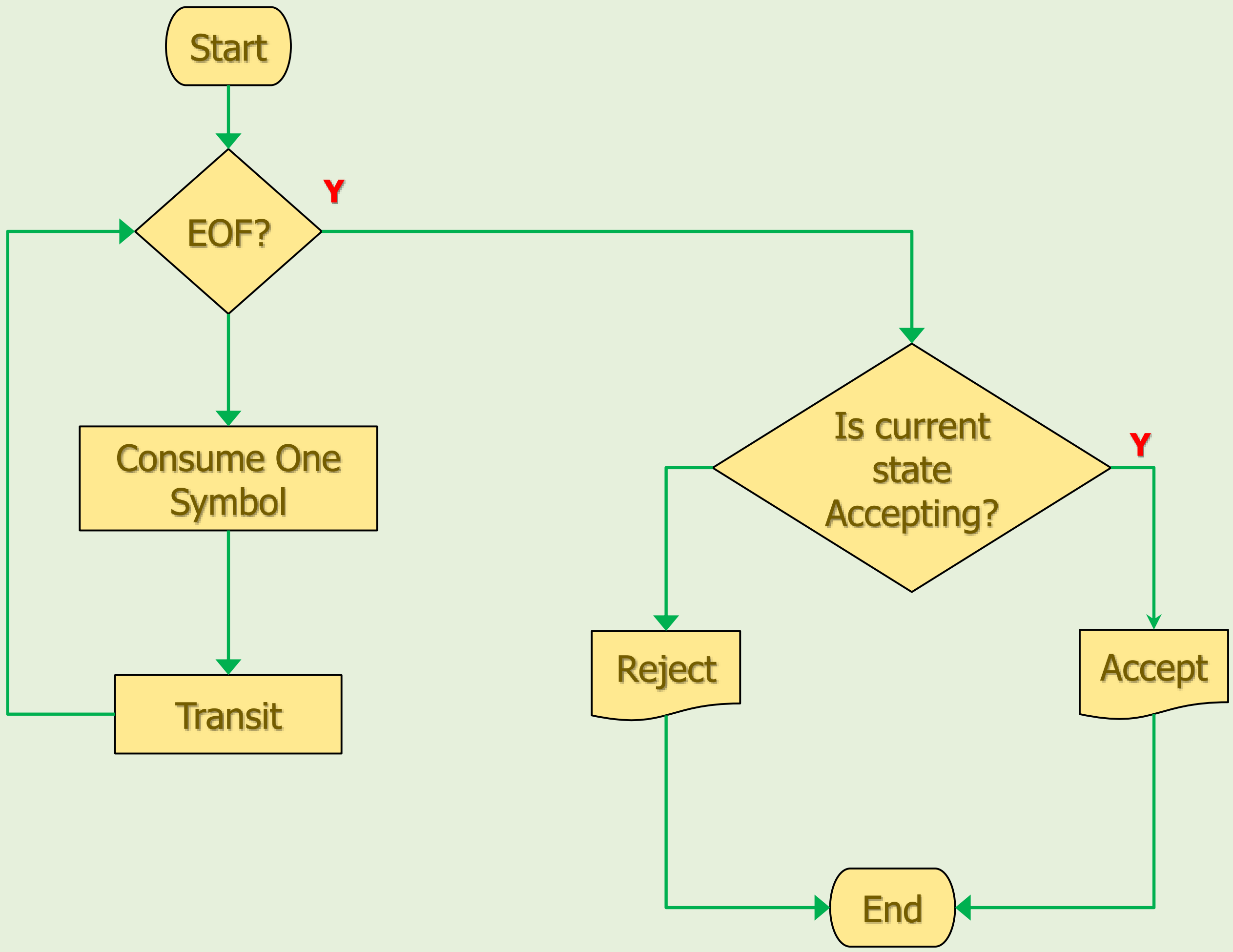
IFF

At least one symbol of w is NOT consumed. $\equiv \sim c$

OR

The DFA is NOT in an accepting (final) state. $\equiv \sim f$

DFAs Operation Flowchart



DFAs Operation Pseudo Code

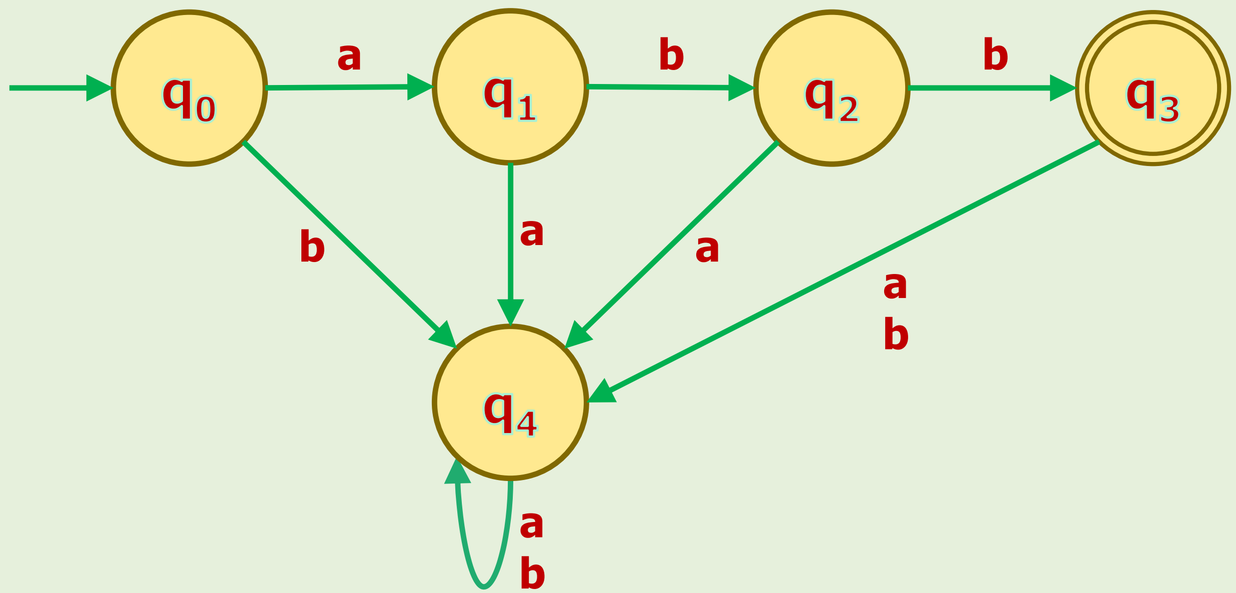
1. Go to step 5 if EOF.
2. Consume a symbol.
3. Transit based on the logic of the current state.
4. Go to step 1
5. If the current state is "accepting state", change the output to "Accept".

Analysis Examples

Analysis Examples

Example 9

- What language does the following DFA accept?
- What is Σ?

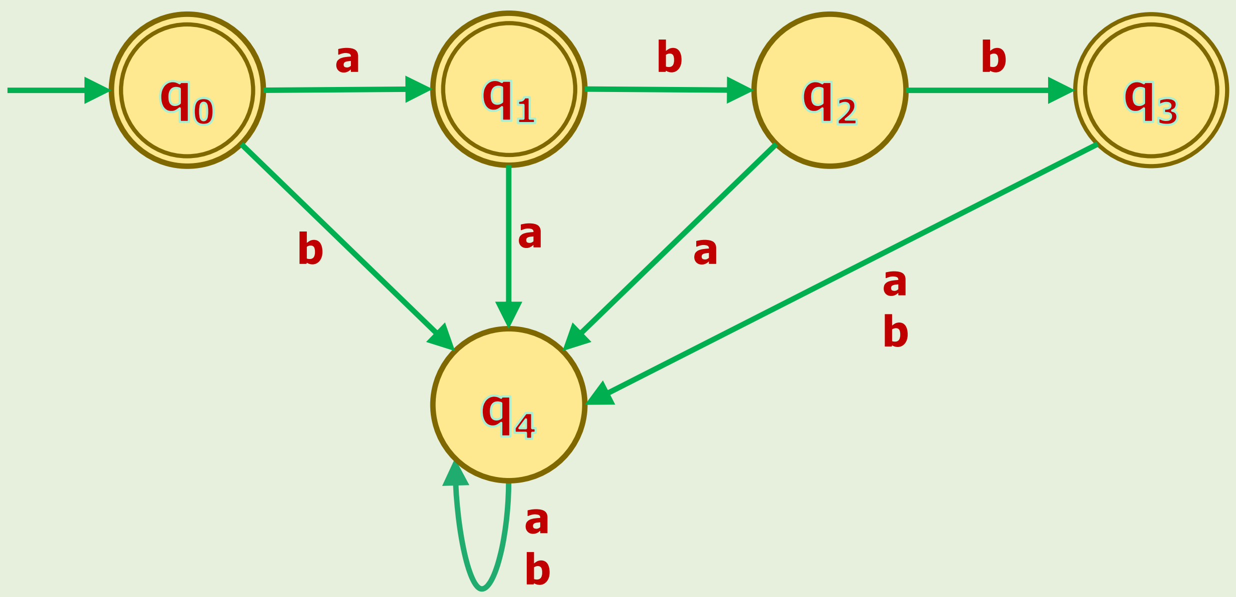


- $L = \{abb\}$ over $\Sigma = \{a , b\}$

Analysis Examples

Example 10

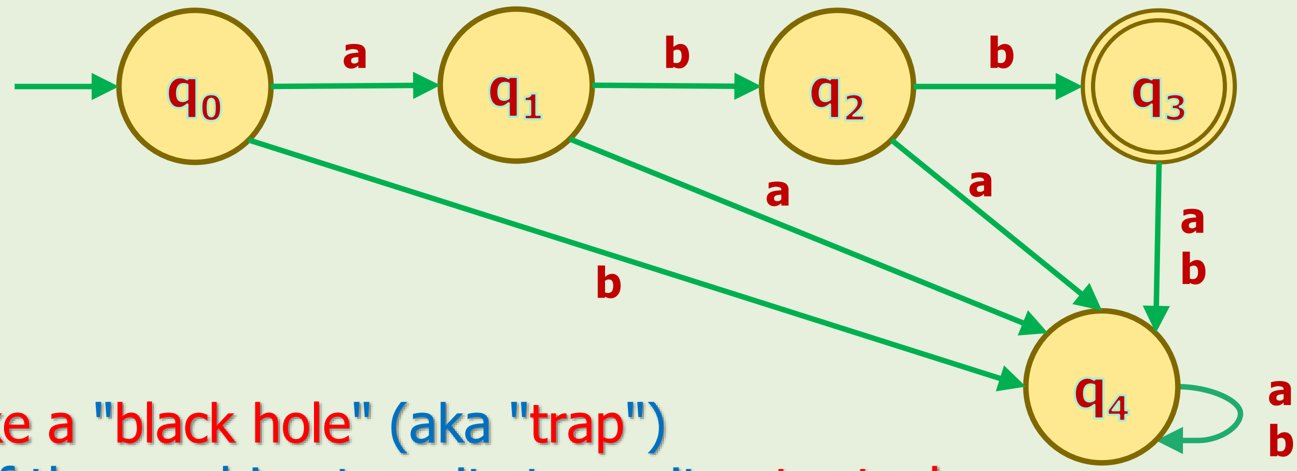
- What language does the following DFA accept?



- $L = \{\lambda, a, abb\}$ over $\Sigma = \{a , b\}$

Analysis Examples: Notes

1. We don't need to show the "output" and the "clock" any longer!
2. The role of q_4 in the previous examples:

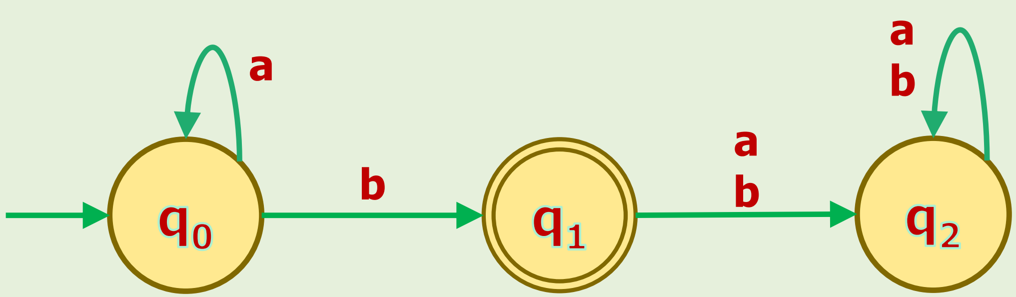


- q_4 acts like a "black hole" (aka "trap") because if the machine transits to q_4, it gets stuck in it and would not be able to transit to any other states.
- We use it to reject majority strings of $\overline{L}$.
- Sometimes I call it "hell"!
- Now we understand the CS equivalent of "Go to hell!"!

Analysis Examples

Example 11

- What language does the following DFA accept?
  - Represent the language by set builder method.



Design Examples

Design Examples

Example 12

- Let L = {ad , ada , adam} over Σ = {a , d , m}.
- Design a DFA to accept L.

Design Important Notes

1. The machine should be designed in such a way that:

It accepts all strings of **L**.

AND

It rejects all strings of $\overline{L}$.

Design Important Notes

2. To test your machine, all accepted strings and rejected strings should be picked from Σ^*.
   - We are not allowed to input strings from outside of Σ^*.

A million dollar question!

- Can you ever claim that your design (code) works perfect?
- Never, because Σ^* is infinite
  and you cannot test your code with infinite test cases.
- So, theoretically every design (code) has potential bugs!


This is me!
A MILLION DOLLAR QUESTION!!
Dec6.2017

Design Examples

Example 13

- Let L be the set of strings starting with prefix ab over $\Sigma = \{a , b\}$.

a. Write a set-builder for L.

b. Design a DFA to accept L.

Homework: DFA Design

- For each of the following languages over $\Sigma = \{a , b\}$:
  a. Write a set-builder to represent the language.
  b. Design a DFA to accept the language.

1. The set of strings that contains exactly one 'a'
2. The set of strings that contains at least one 'a'
3. The set of strings ending with suffix ab

References

1. Linz, Peter, "An Introduction to Formal Languages and Automata, 5th ed.," Jones & Bartlett Learning, LLC, Canada, 2012
2. Kenneth H. Rosen, "Discrete Mathematics and Its Applications, 7th ed.," McGraw Hill, New York, United States, 2012
3. Michael Sipser, "Introduction to the Theory of Computation, 3rd ed.," CENGAGE Learning, United States, 2013
ISBN-13: 978-1133187790