San José State University
Department of Computer Science

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

Nondeterministic Finite Automata

(Part 2)

Lecture 10
Day 10/31

CS 154
Formal Languages and Computability
Fall 2019

Agenda of Day 10

- About Midterm 1
- Feedback and Solution of Quiz 3
- Summary of Lecture 09
- Lecture 10: Teaching ...
  - Nondeterministic Finite Automata (Part 2)

About Midterm 1

Reminder 2

- Midterm #1 (aka Quiz+)
  - Date: Thursday 09/26
  - Value: 10%
  - Topics: Everything covered from the beginning of the semester
  - Type: Closed $y \in$ Material

    Material = {Book, Notes, Electronic Devices, Chat, ... }

- The cutoff for this midterm is the end of lecture 09.

Study Guide

- I've given you a study guide about the type and number of questions via Canvas.

Solution and Feedback of Quiz 3 (Out of 22)

Section	Average	High Score	Low Score
01 (TR 3:00 PM)	17.65	22	13
02 (TR 4:30 PM)	17.31	22	11
03 (TR 6:00 PM)	19.21	22	14

Summary of Lecture 09: We learned ...

NFAs

- Two violations in DFAs were introduced...
- Violation #1
  During a timeframe, the machine has no (zero) transition.
  - The transition function is partial function.
- Violation #2
  During a timeframe, the machine has more than one transition.
  - The transition function is a multifunction.

- We relaxed DFAs constraint and introduced a new class:

  Nondeterministic Finite Automata

- The same building blocks
- Simpler transition graphs
- Their behavior are similar to DFAs except for those two violations.

1. When NFAs have zero transition, ...

   ... they halt.

2. When there are more than one transition, ...

   ... they start parallel processing.

Any question?

Summary of Lecture 09: We learned ...

NFAs

- NFAs halt when ...
  - All input symbols are consumed. $\equiv$ **c**

  OR

  - They have zero transition. $\equiv$ **z**

$$(\mathbf{c} \lor \mathbf{z}) \leftrightarrow \mathbf{h}$$

- A string is accepted iff ...

  ... at least one process accepts it.

- A string is rejected iff ...

  ... all processes reject it.

- Logical statement of accepting a string for one process ...

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

  - Recall that for DFAs, we changed $(\mathbf{h} \land \mathbf{c} \land \mathbf{f}) \leftrightarrow \mathbf{a}$ to $(\mathbf{c} \land \mathbf{f}) \leftrightarrow a$ because h and c have the same value.
  - But for NFAs, **h** and **c** might have different values.

- Logical statement of rejecting a string for one process ...

$$(\sim\mathbf{h} \lor \sim\mathbf{c} \lor \sim\mathbf{f}) \leftrightarrow \sim\mathbf{a}$$

Any question?

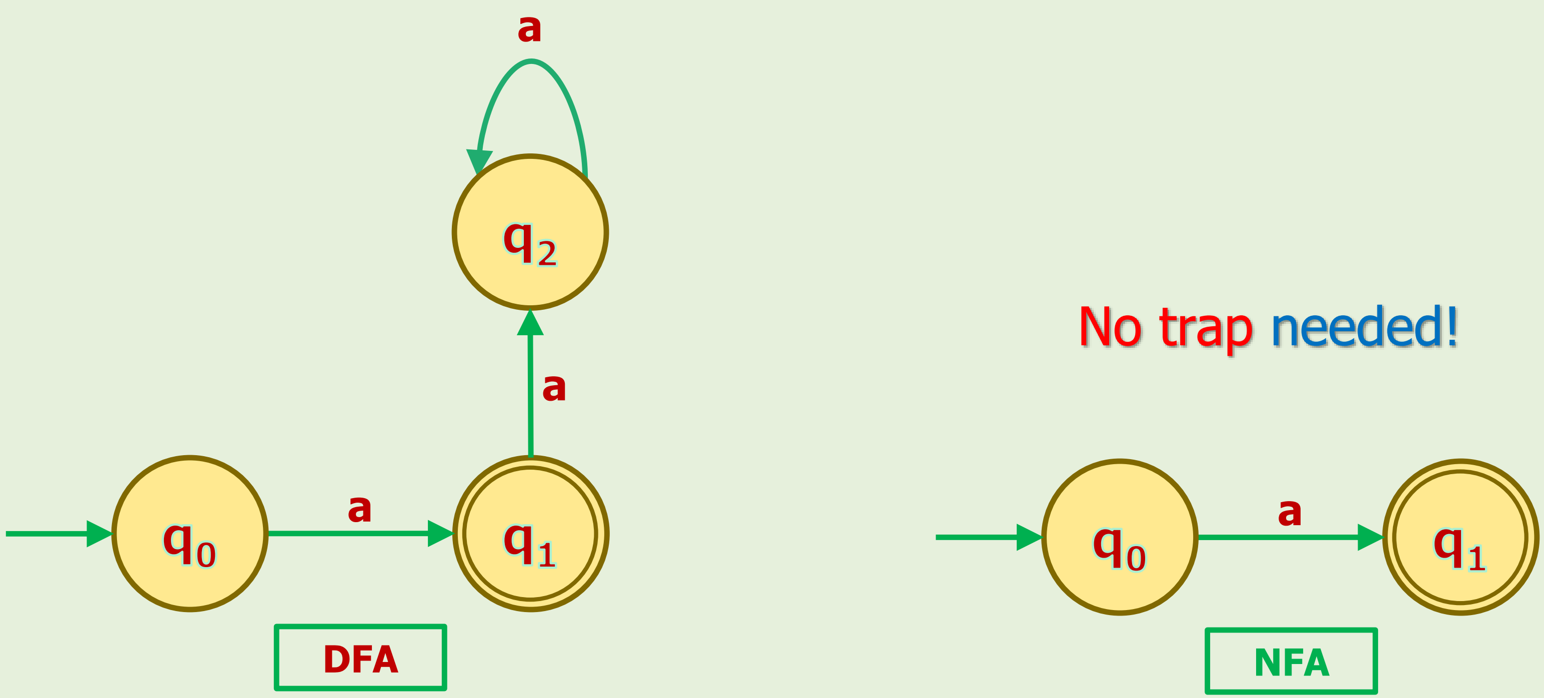
1. Why do We Need a New Class?

Revisited Question

- We introduced NFAs because their transition graphs are simpler.

Example 7

Design a DFA and an NFA for L = {a} over Σ = {a}

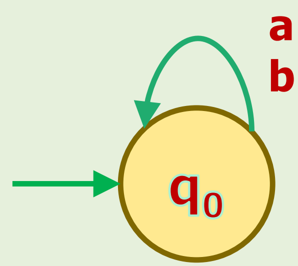


1. Why do We Need a New Class?

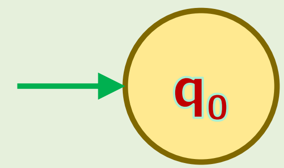
Revisited Question

Example 8: Empty Language

L = { } over Σ = {a, b}



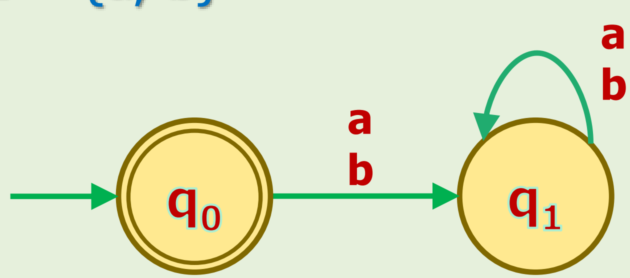
DFA



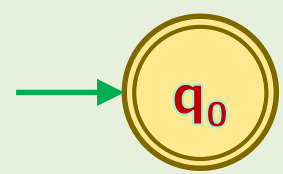
NFA

Example 9: Empty String Language

L = {λ} over Σ = {a, b}



DFA

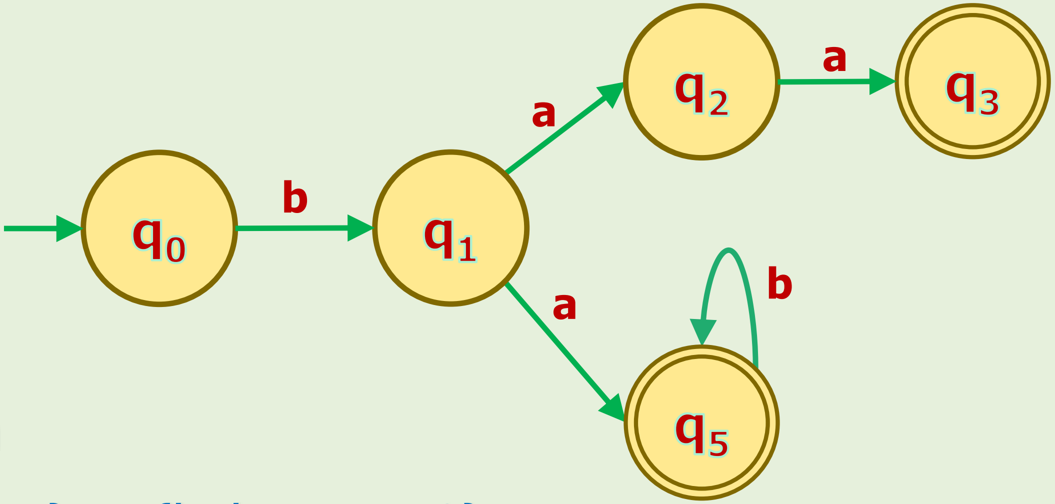


NFA

Associated Language to NFAs

Example 10

- What is the associated language to the following automaton over $\Sigma = \{a, b\}$?



Solution

$L = \{baa\} \cup \{bab^n : n \geq 0\}$

- Design a DFA to accept L.

A Special Transition

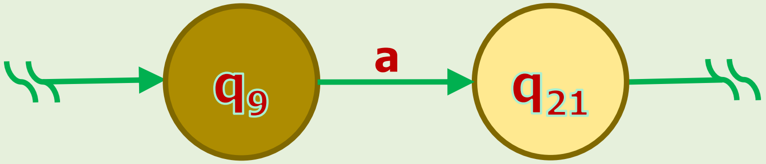
Introduction

- We are going to introduce a special kind of transitions.
- They are strictly prohibited in DFAs ...
- But allowed in NFAs.

Let's Shine our Knowledge

Question

- In the following transition, if the machine is in q_9, then what is the "condition" for transition to q_{21}?



Answer

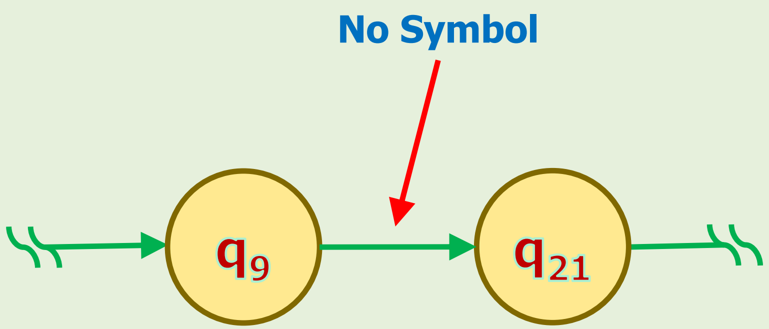
- If the machine is in q_9 AND the next input symbol is 'a', then the machine transits to q_{21}.

Conclusion

- The transition from q_9 to q_{21} is "conditional".

Let's Remove the Condition

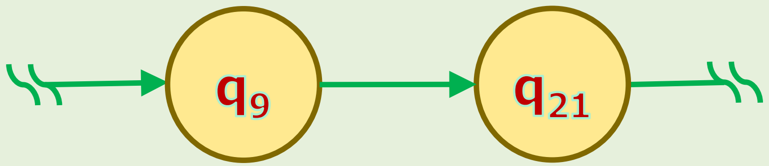
- What if we remove the condition?



- Then we've created a "short-circuit".
- A "short-circuit" is an edge with no symbol.

What is the Meaning of Short-Circuit?

- If there is no symbol, then there is NO condition for transition!

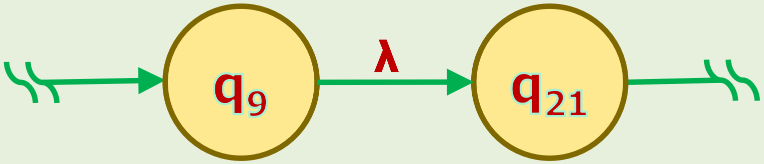


Consequently, the machine can transit unconditionally!

- In other words, if the machine is in q_9,
  it can unconditionally transit to q_{21}.

The Symbol of Short-Circuit

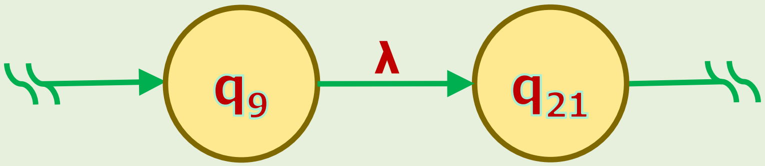
- The symbol "λ" was chosen to represent "short-circuit".



- Because of this symbol, this type of transitions are called "lambda transition" or "λ-transition".

- It has an important role in automata theory.

Meaning of λ From Different Angle



- We've already used λ to represent "empty string".
- As we said before, λ means "NO symbol" (or zero symbol).
- A short-circuit has "no symbol" too.
- That's why the short-circuit is represented by λ.

- Be careful:
  - Using λ as "empty string" and the symbol of "short-circuit" can be confusing but you'll get used to it!
- But what is the meaning and consequence of this?

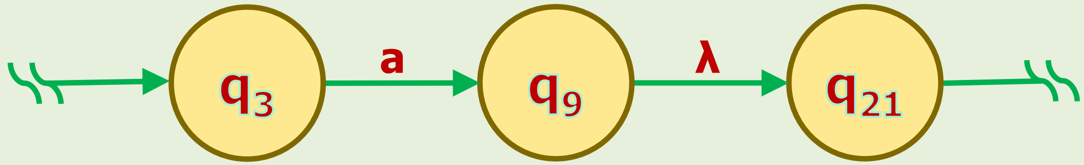
λ-Transition Definition

Definition

- λ-transition in automata theory is a transition that the machine can unconditionally transit.
- This is a general definition for all types of automata.
- The concept of λ-transition changes our view about sub-rules of transition function.
- Let's take an example.

How To Represent the Sub-Rule

- What is the value of $\delta(q_3 , a) = ?$



- Since the machine can transit unconditionally, it means that ...

  it can stay as well.

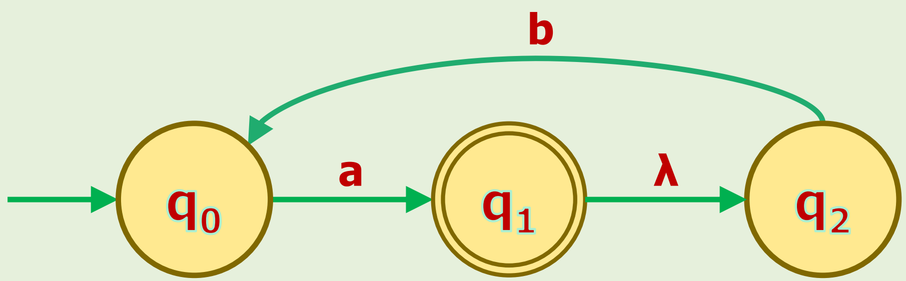
- Therefore, the sub-rule for this example is:

$$\delta(q_3 , a) = \{q_9 , q_{21}\}$$

How To Represent the Sub-Rule : Example

Example 11

- Write the transition function δ of the following transition graph over $\Sigma = \{a, b\}$ by using algebraic notation.



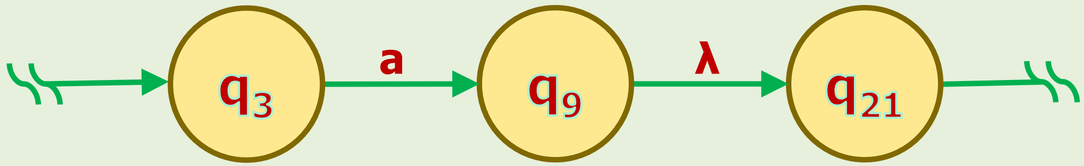
Solution

$$\begin{cases} \delta(q_0, a) = \{q_1, q_2\} \\ \delta(q_0, b) = \{\ \} \\ \delta(q_1, a) = \{\ \} \\ \delta(q_1, b) = \{q_0\} \\ \delta(q_2, a) = \{\ \} \\ \delta(q_2, b) = \{q_0\} \end{cases} \Rightarrow \begin{cases} \delta(q_0, a) = \{q_1, q_2\} \\ \delta(q_1, b) = \{q_0\} \\ \delta(q_2, b) = \{q_0\} \end{cases}$$

- We can eliminate those having empty ranges.

NFAs Behavior

- We learned that:



$$\delta(q_3 , a) = \{q_9 , q_{21}\}$$

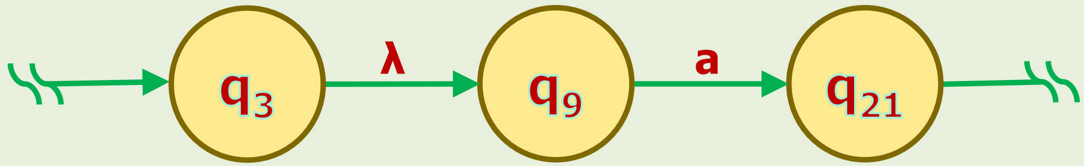
- The NFA has multiple choices.

Stay in q_9, OR transit to q_{21}.

- How should it behave when it has multiple choices?
- It would check all possibilities by "parallel processing".
- The procedure of creating new processes is the same way that we learned before.

NFAs Behavior

- Another situation:



$\delta(q_3\ ,\ a) = \{q_{21}\}$

- Note that in this situation, the NFA has only one choice.
- Therefore, it does NOT need to initiate new process.
- As a general rule:

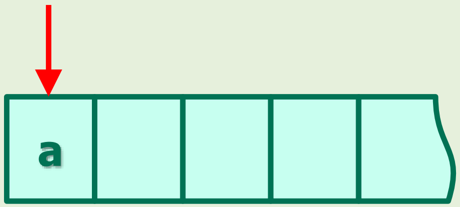
All machines initiate new processes when encounter multiple choices.

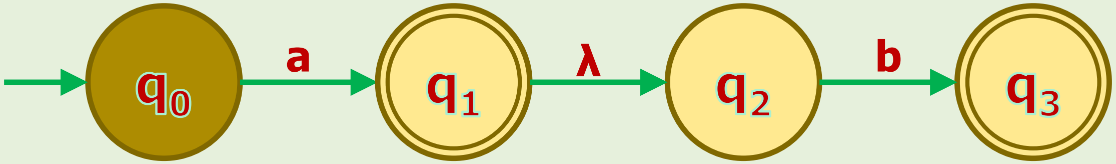
- Now, let's see λ-transitions in Action!

λ-Transitions in Action

λ-Transitions in Action

Example 12: Starting Configuration

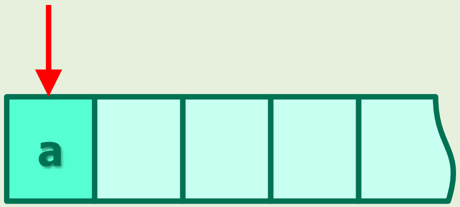


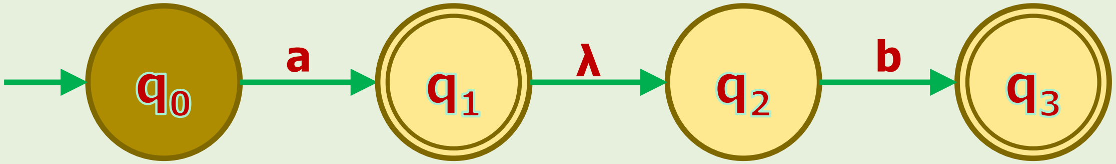


- Process #1 (main) starts normally.

λ-Transitions in Action

Example 12 (cont'd)



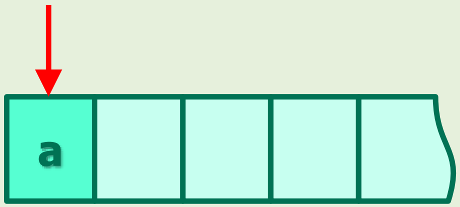


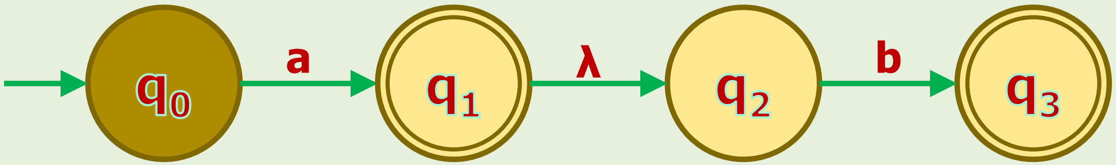
- Input tape reads 'a' and sends it to the control unit.
- The control unit makes a decision based on $\delta(q_0, a) = \{q_1, q_2\}$

λ-Transitions in Action

Example 12 (cont'd)

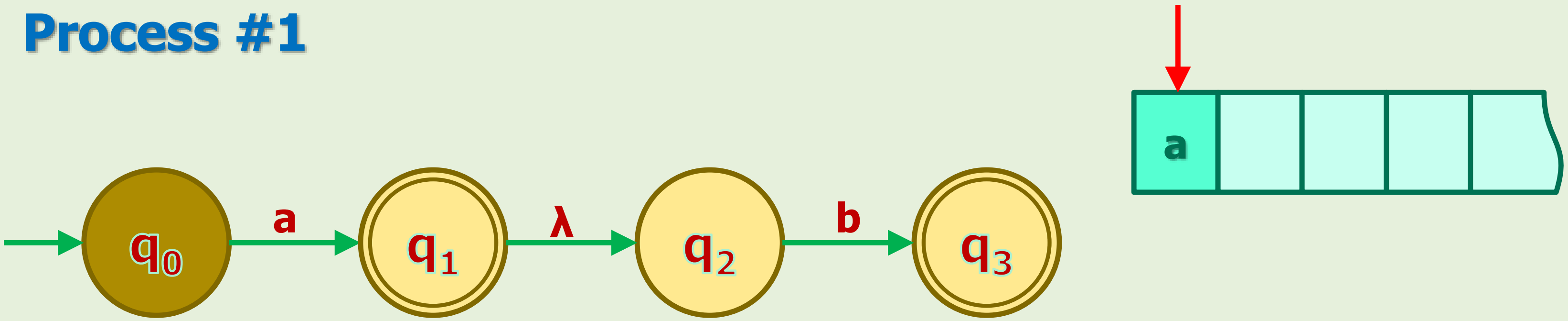
$\delta(q_0, a) = \{q_1, q_2\}$





- It encounters two possibilities: transition to q_1 or q_2.
- So, parallel processing starts!

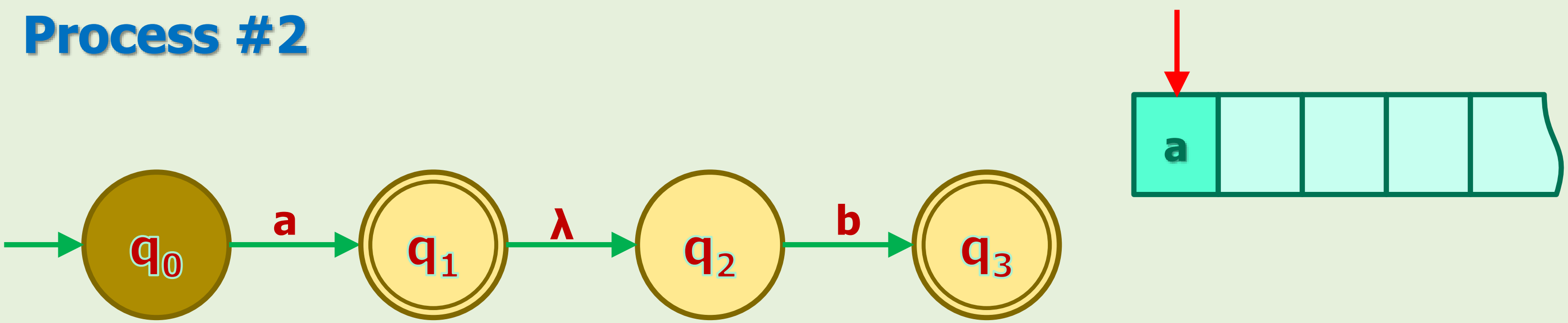
Process #1



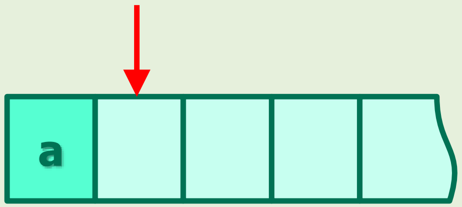
$\delta(q_0\ , a) = \{q_1\ , q_2\}$

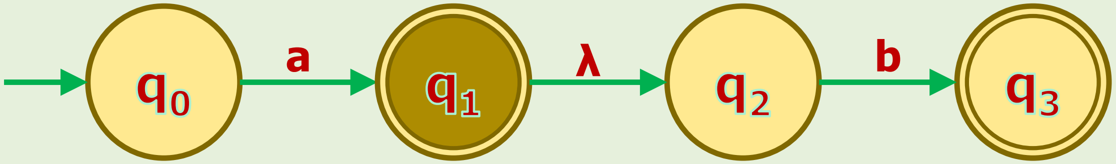
It replicates itself.

Process #2



Process #1

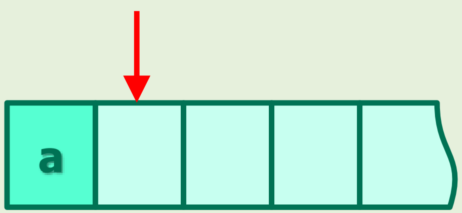


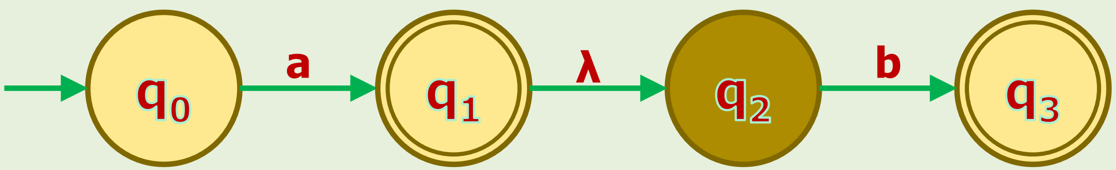


$\delta(q_0\ ,\ a) = \{q_1\ ,\ q_2\}$

This process consumes 'a' and transits to q_1.

Process #2

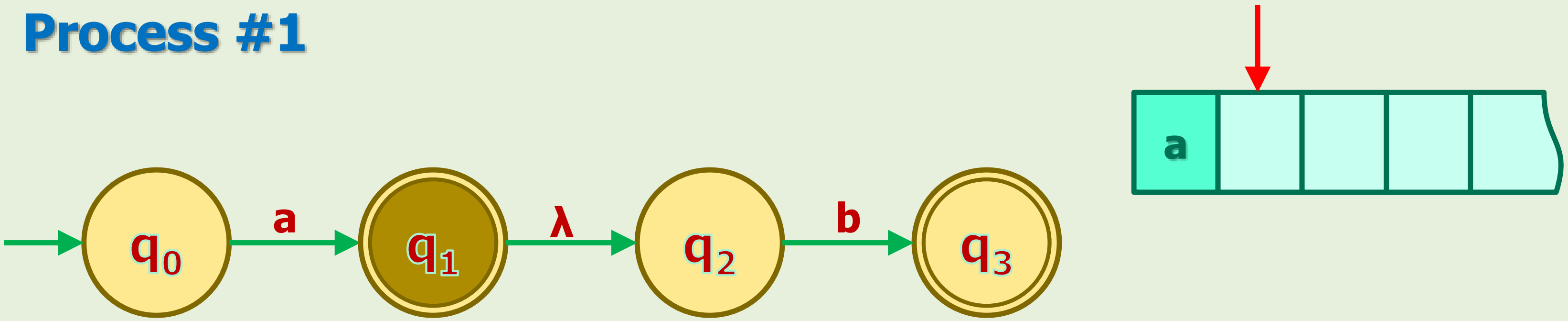




This process consumes 'a' and transits to q_2.

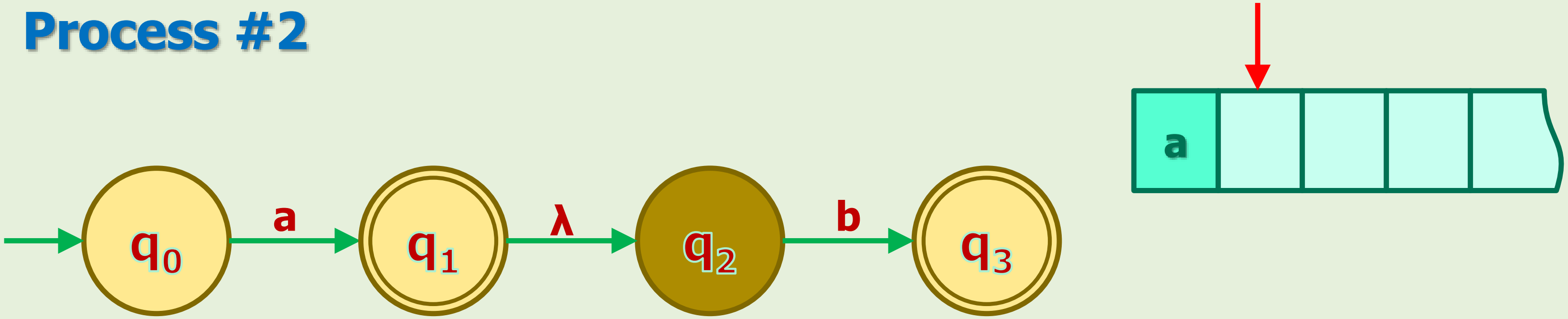
This is the end of timeframe 1.

Process #1



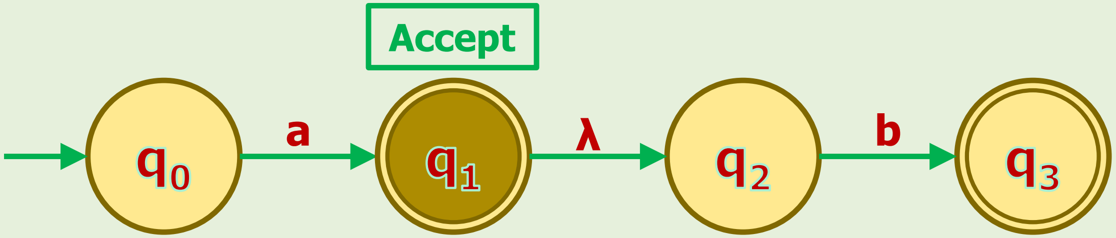
Process #1 is out of symbol and has to halt.

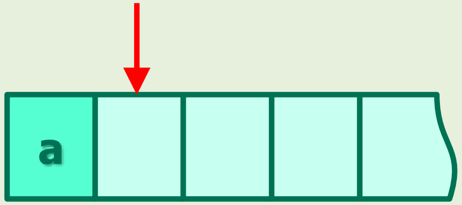
Process #2



Process #2 is out of symbol and has to halt.

Process #1

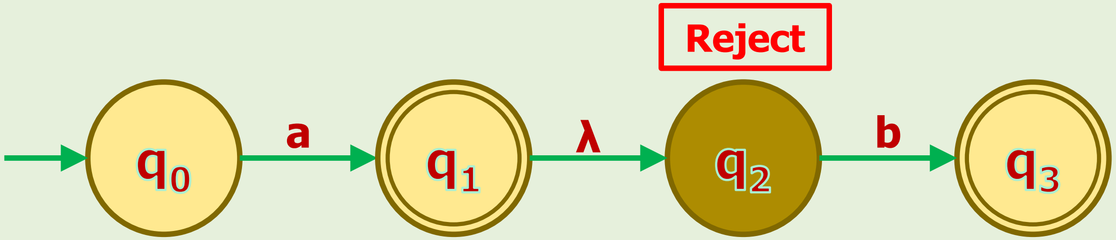


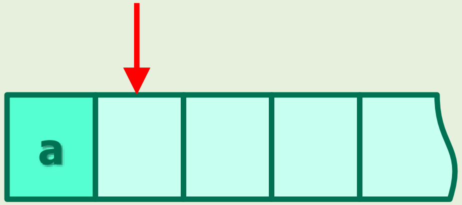


Process #1 halts in an accepting state AND all symbols are consumed.

So, process #1 accepts w.

Process #2





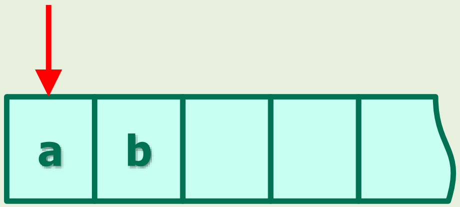
Process #2 halts in a non-accepting state.

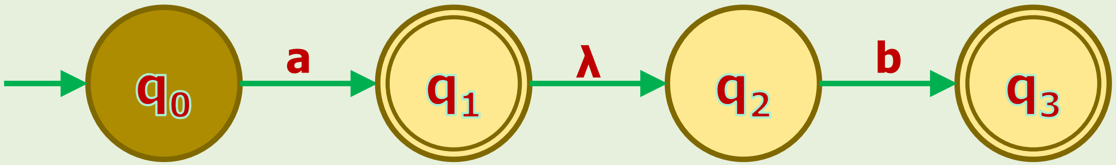
So, process #2 rejects w.

Overall Accepted

λ-Transitions in Action

Example 13: Starting Configuration

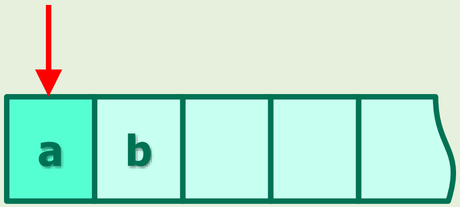


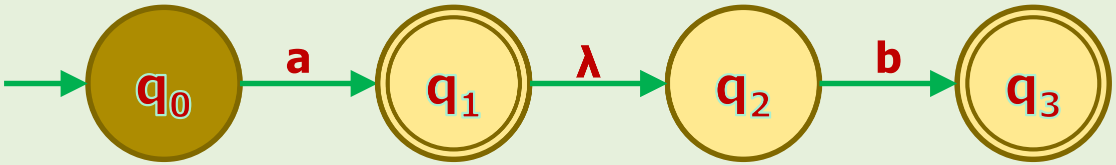


- Process #1 (main) starts normally.

λ-Transitions in Action

Example 13 (cont'd)

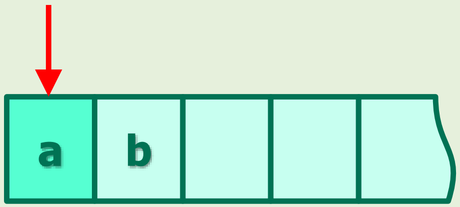




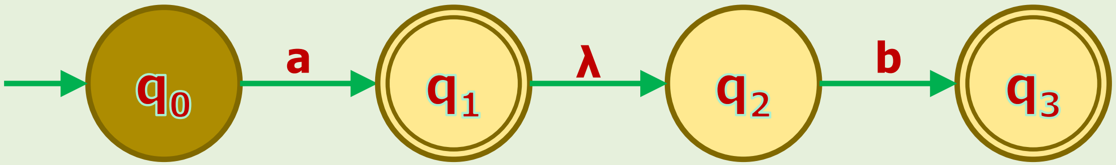
- Input tape reads 'a' and sends it to the control unit.
- The control unit makes a decision based on $\delta(q_0, a) = \{q_1, q_2\}$

λ-Transitions in Action

Example 13 (cont'd)

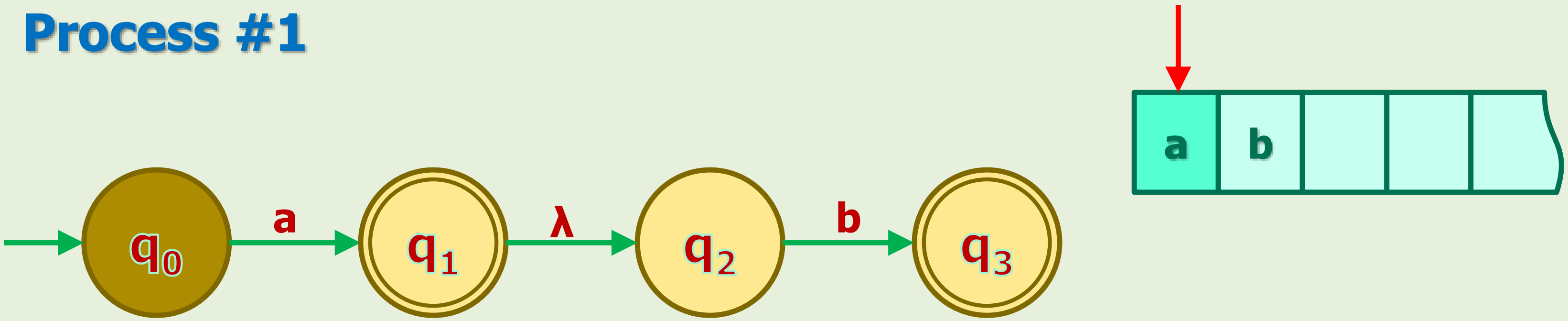


$\delta(q_0, a) = \{q_1, q_2\}$



- It encounters two possibilities: transition to q_1 or q_2.
- So, parallel processing starts!

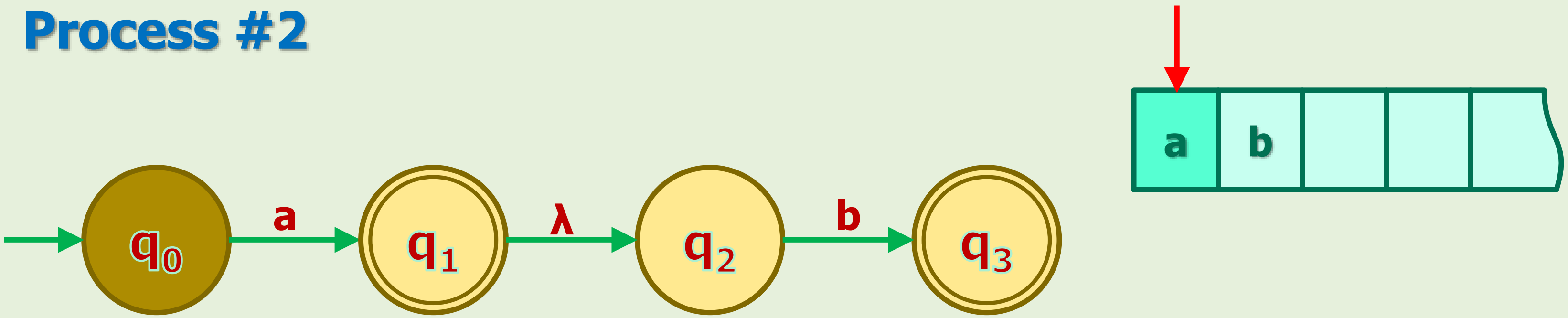
Process #1



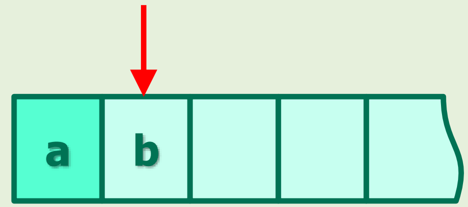
$\delta(q_0 , a) = \{q_1 , q_2\}$

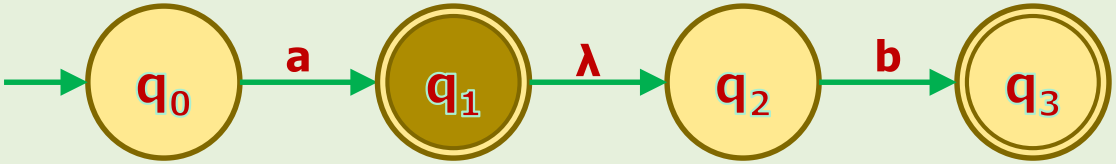
It replicates itself.

Process #2



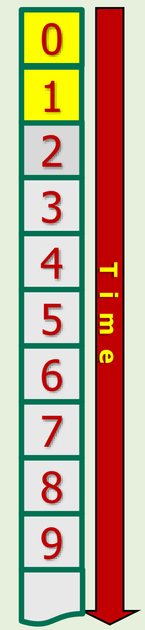
Process #1



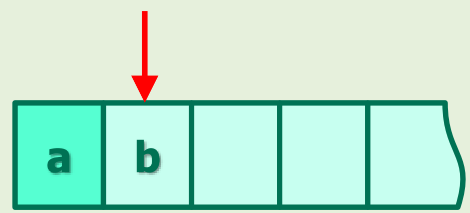


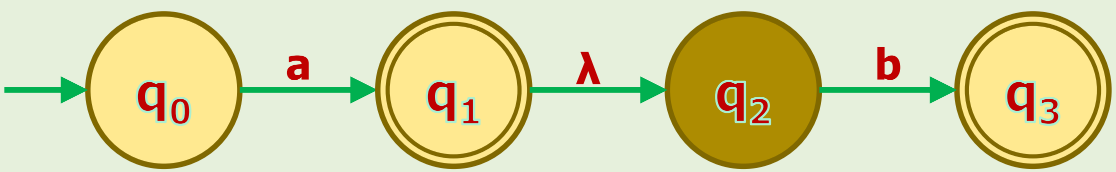
$\delta(q_0, a) = \{q_1, q_2\}$

Process #1 consumes 'a' and transits to q_1.



Process #2

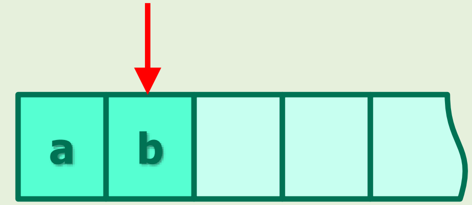


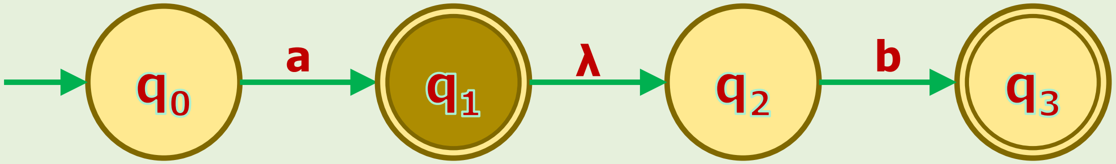


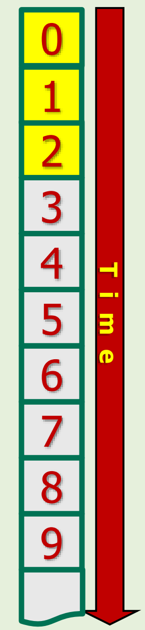
Process #2 consumes 'a' and transits to q_2.

This is the end of timeframe 1.

Process #1



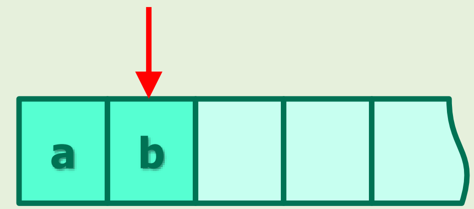


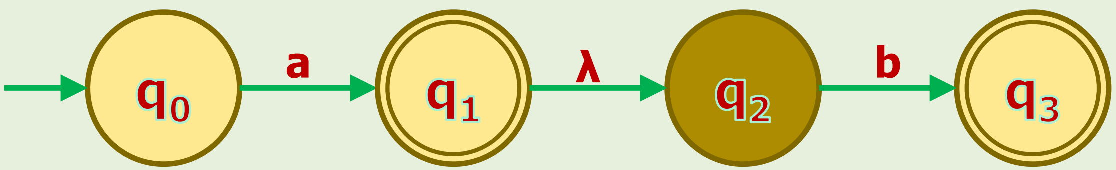


The symbol 'b' is read and sent to the control unit.

Process #1 calculates $\delta(q_1, b) = \{q_3\}$.

Process #2

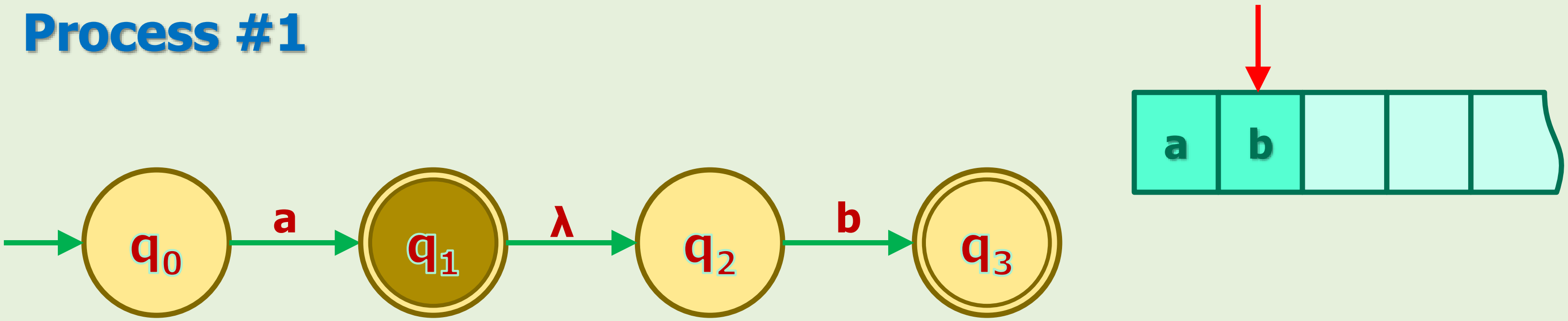




The symbol 'b' is read and sent to the control unit.

Process #2 calculates $\delta(q_2, b) = \{q_3\}$.

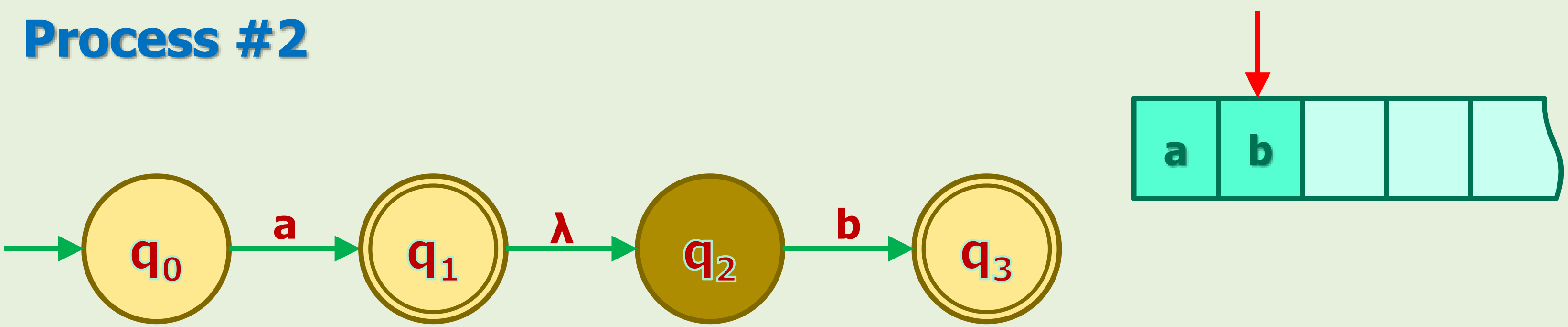
Process #1



$\delta(q_1 , b) = \{q_3\}$

Process #1 consumes 'b' and transits to q_3.

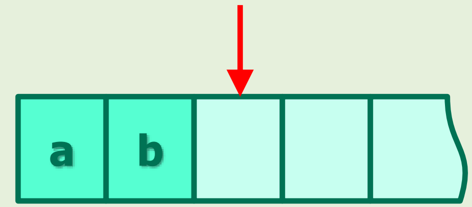
Process #2

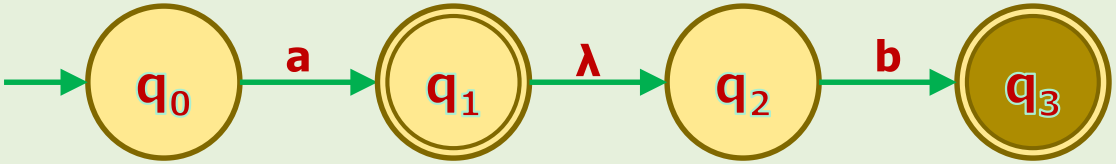


$\delta(q_2 , b) = \{q_3\}$

Process #2 consumes 'b' and transits to q_3.

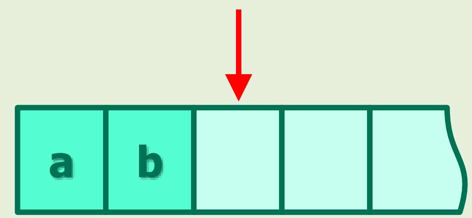
Process #1

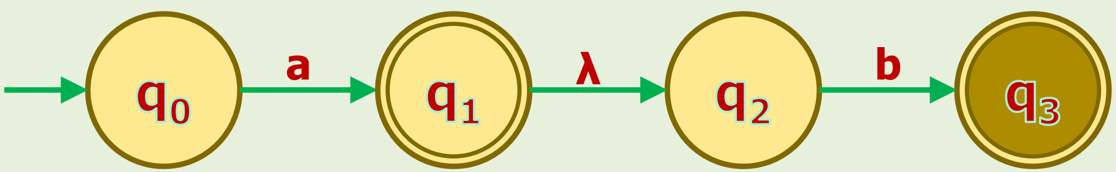




Process #1 is out of symbol and has to halt.

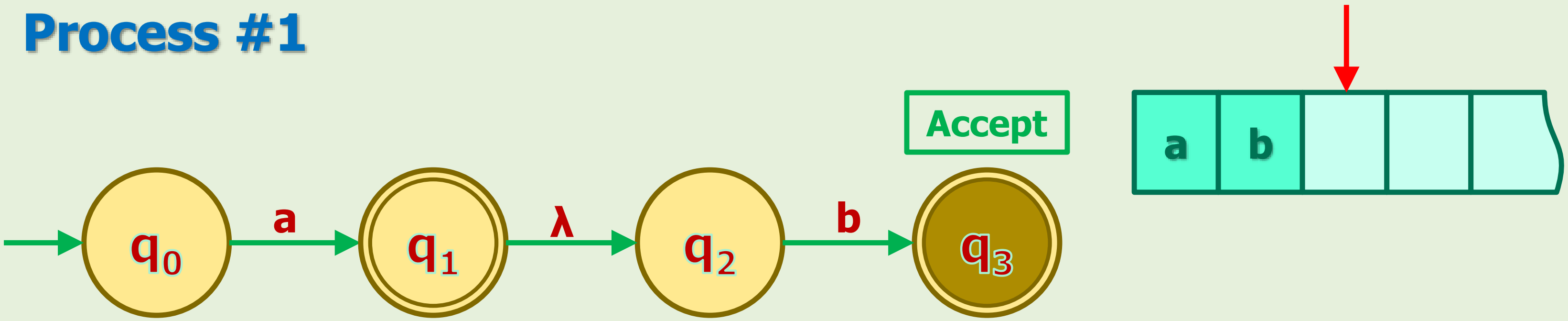
Process #2





Process #2 is out of symbol and hast to halt.

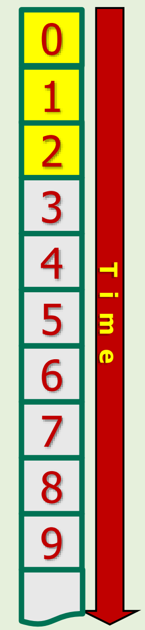
Process #1



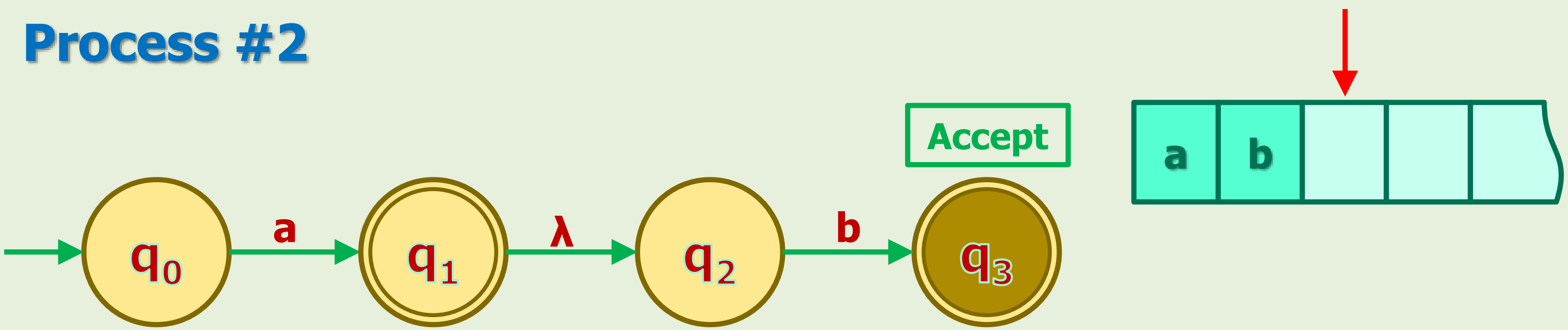
Process #1 halts in an accepting state AND all symbols are consumed.

So, process #1 accepts w.

Overall Accepted



Process #2

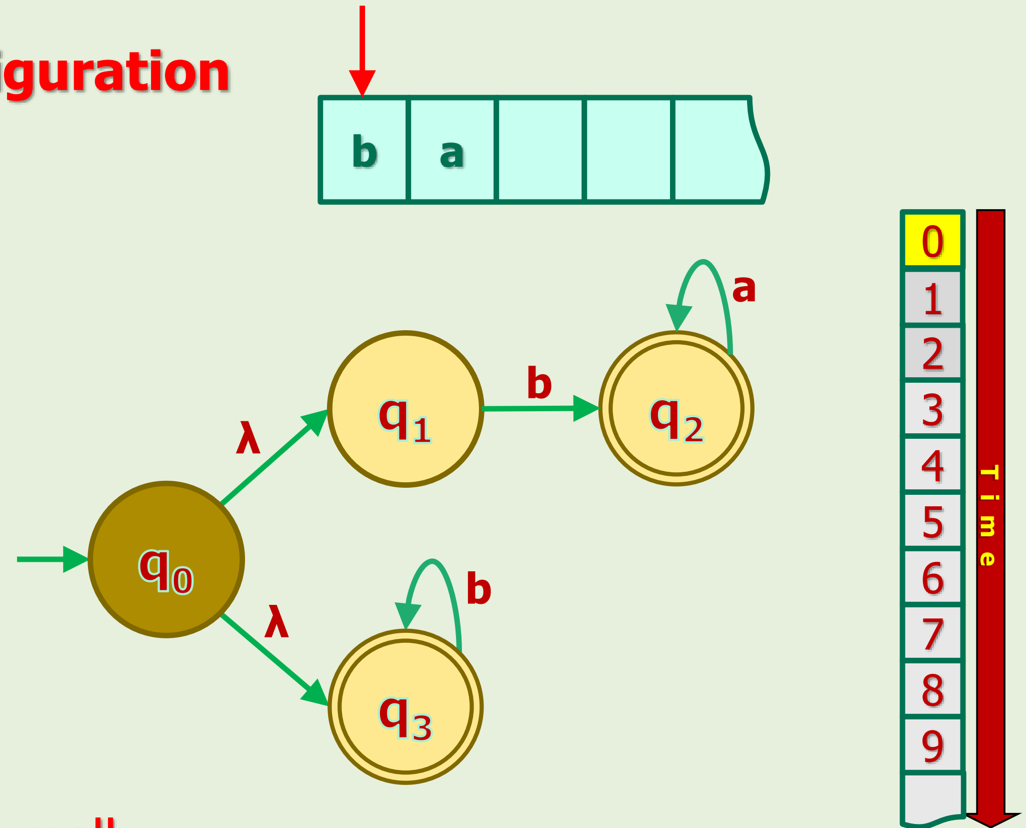


Process #2 halts in an accepting state AND all symbols are consumed.

So, process #1 accepts w.

λ-Transitions in Action

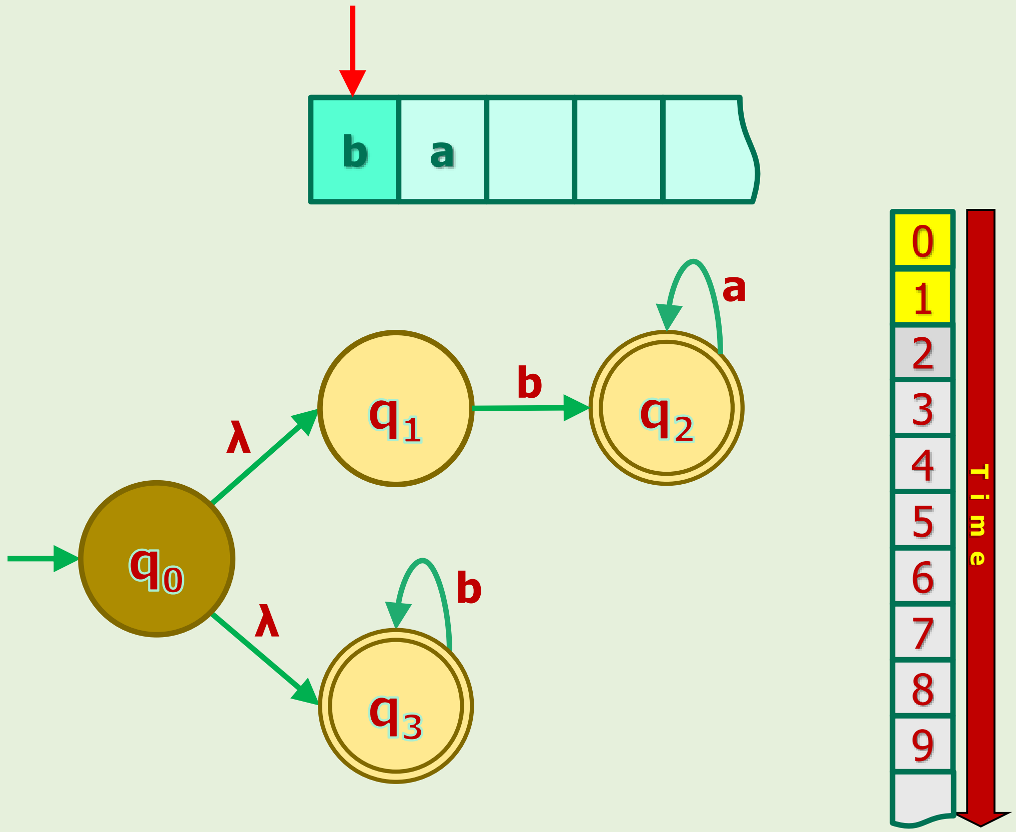
Example 14: Starting Configuration



- Process #1 (main) starts normally.

λ-Transitions in Action

Example 14 (cont'd)

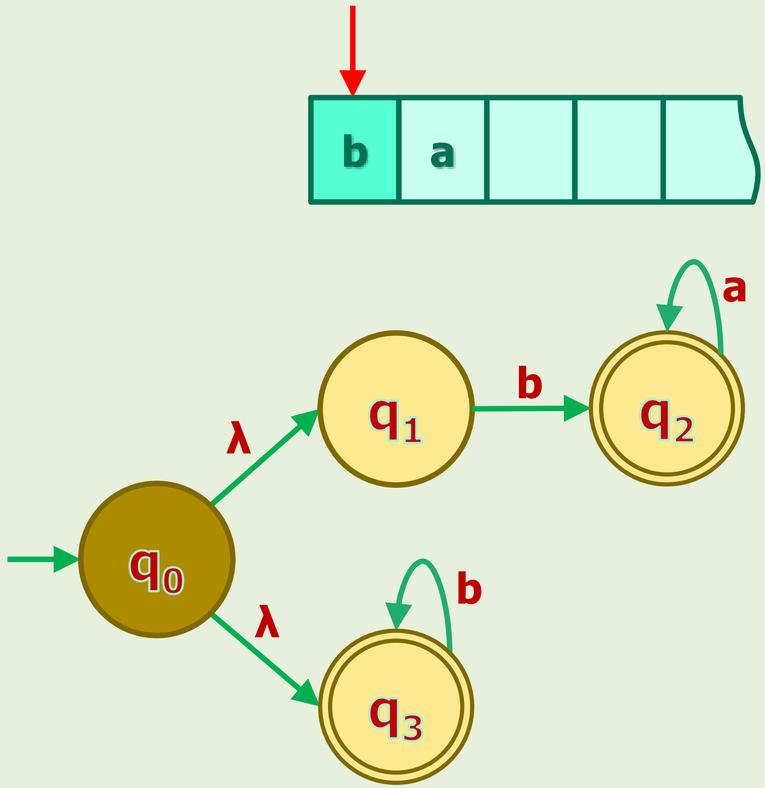


- Input tape reads 'b' and sends it to the control unit.
- The control unit makes a decision based on $\delta(q_0 , b) = \{q_2 , q_3\}$

λ-Transitions in Action

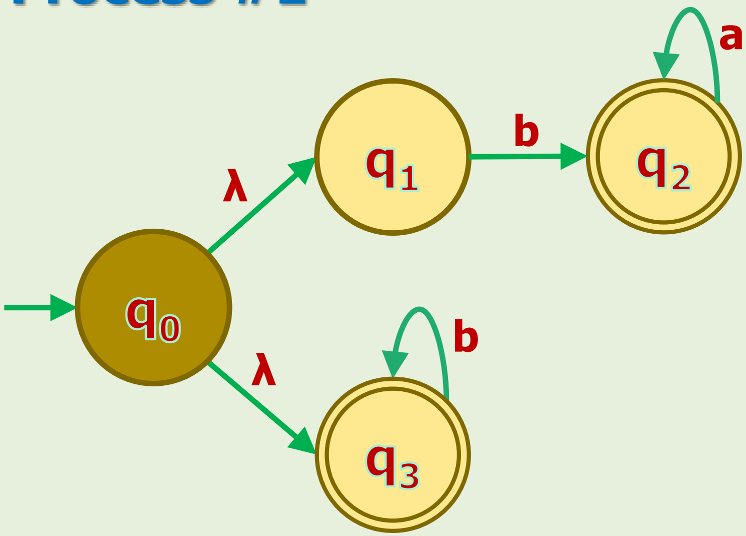
Example 14 (cont'd)

$\delta(q_0 , b) = \{q_2 , q_3\}$



- It encounters two possibilities: transition to q_2 or q_3.
- So, parallel processing starts!

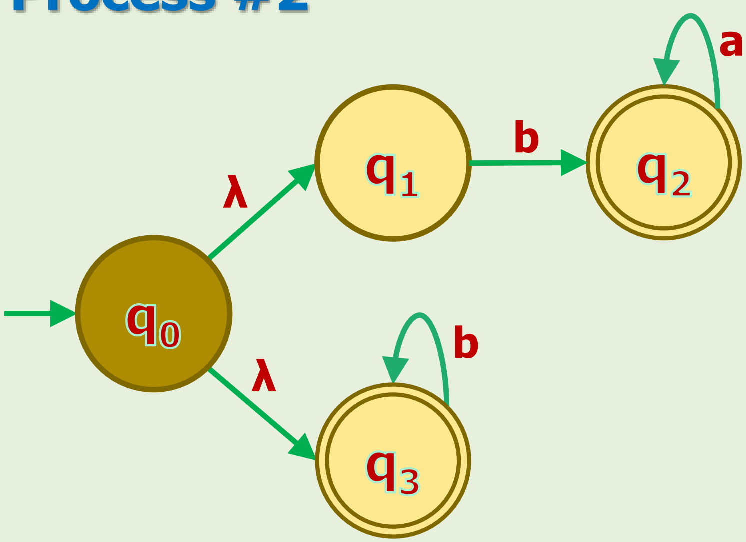
Process #1



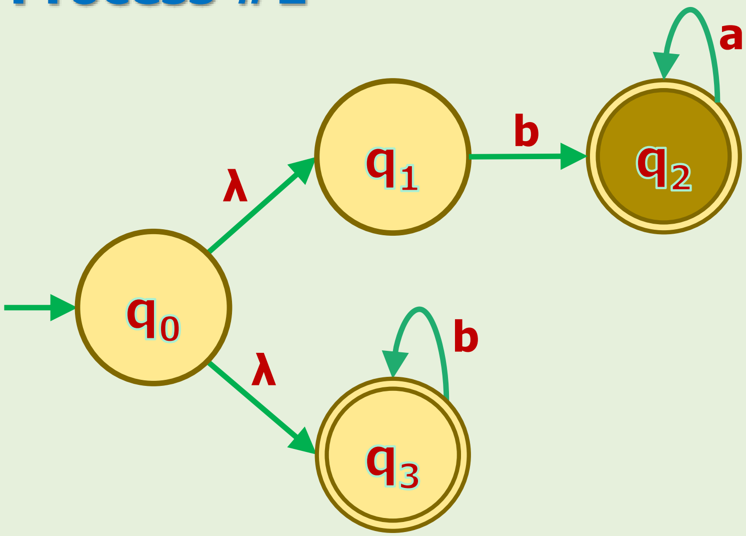
$\delta(q_0, b) = \{q_2, q_3\}$

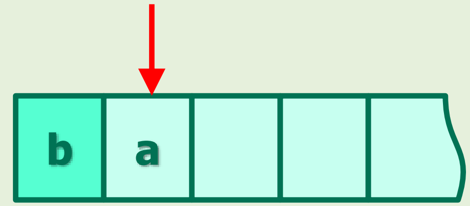
It replicates itself!

Process #2



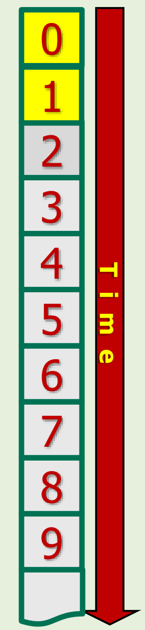
Process #1



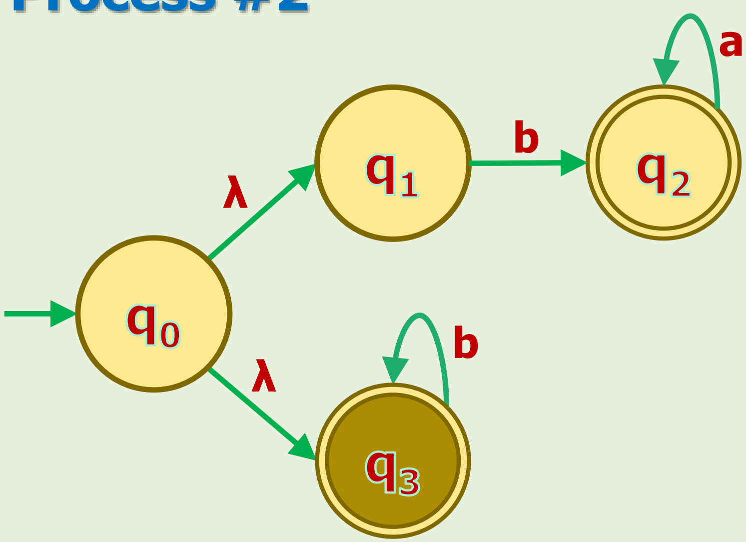


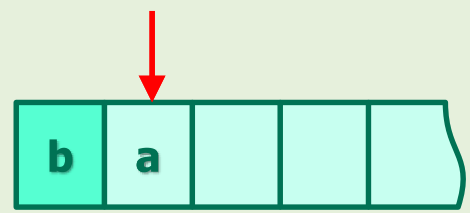
$\delta(q_0 , b) = \{q_2 , q_3\}$

Process #1 consumes 'b' and transits to q_2.



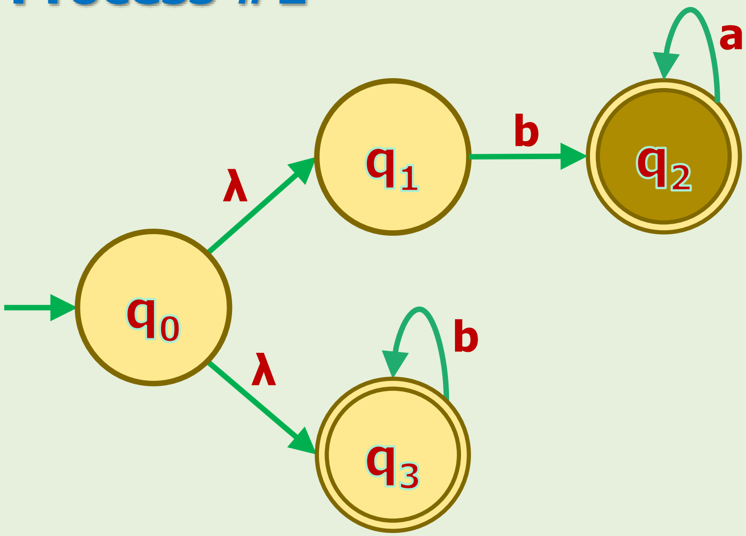
Process #2

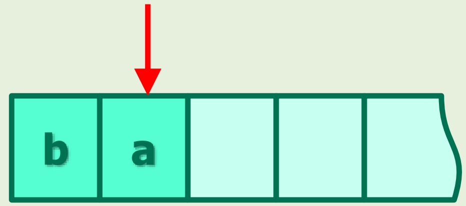




Process #2 consumes 'b' and transits to q_3.

Process #1

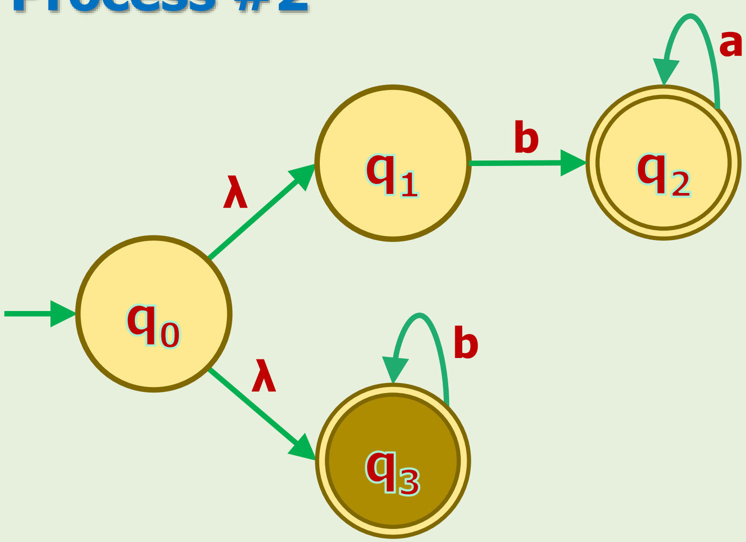


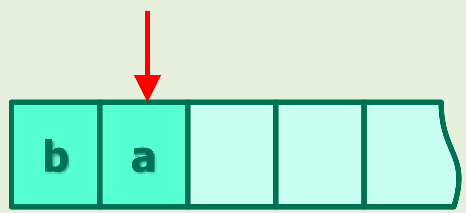


The symbol 'a' is read and sent to the control unit.

It calculates $\delta(q_2, a) = \{q_2\}$

Process #2

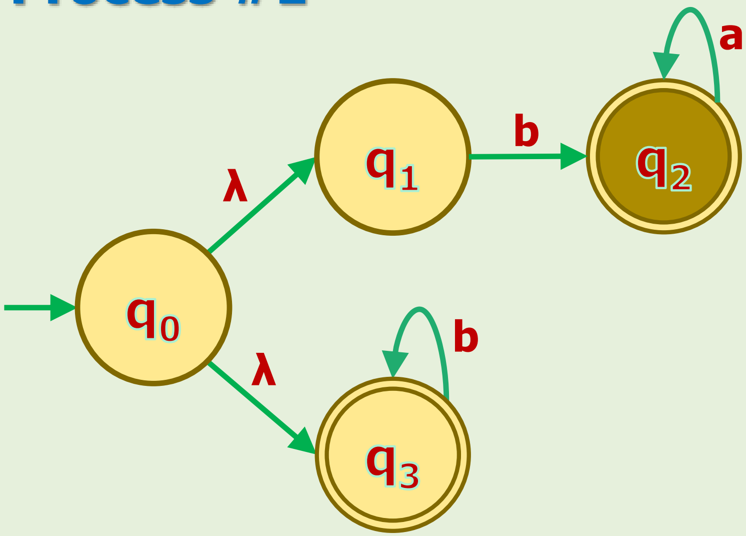


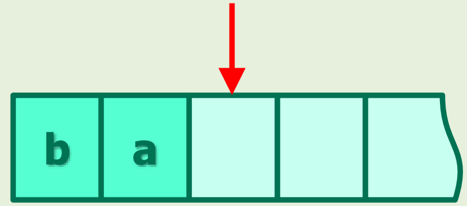


The symbol 'a' is read and sent to the control unit.

It calculates $\delta(q_3, a) = \{\ \}$

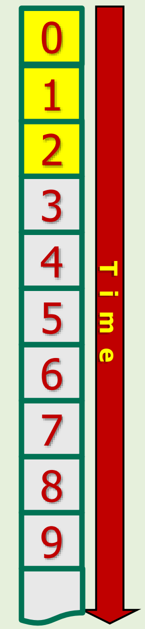
Process #1



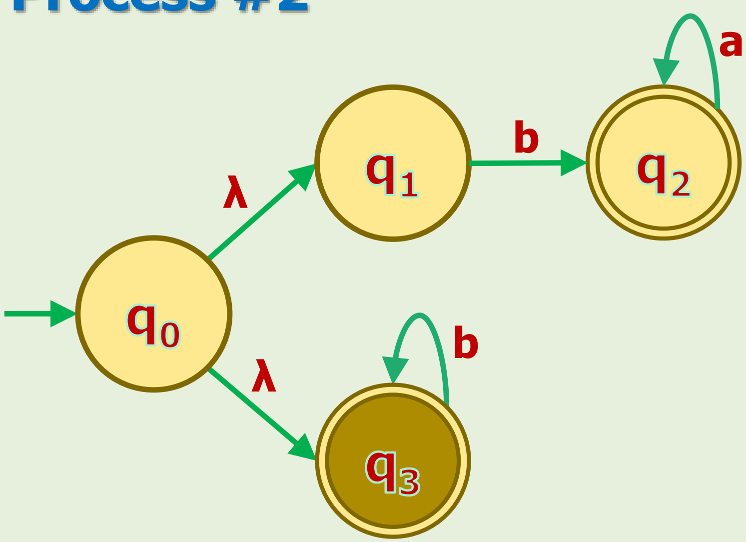


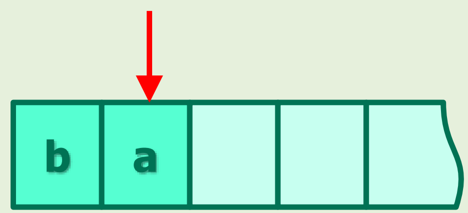
Process #1 consumes 'a' and transits to q_2.

It is out of symbol.



Process #2

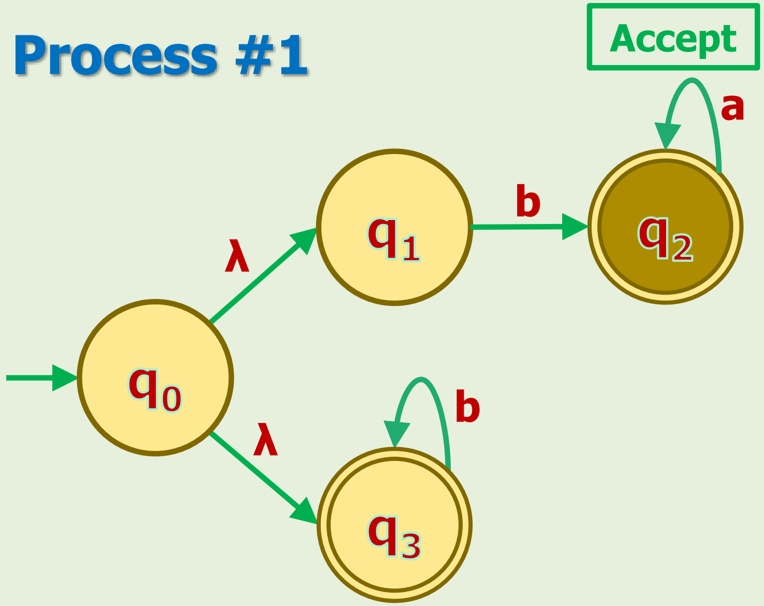


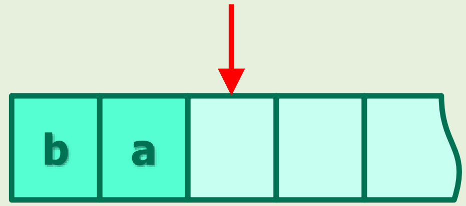


Process #2 has no choice for 'a'.

It has to halt.

Process #1

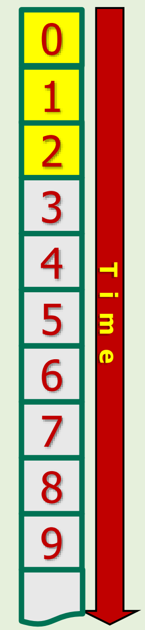




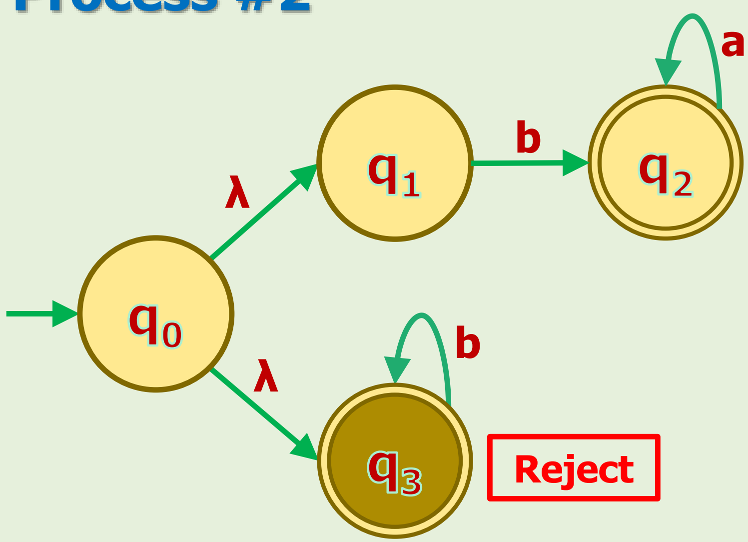
Overall Accepted

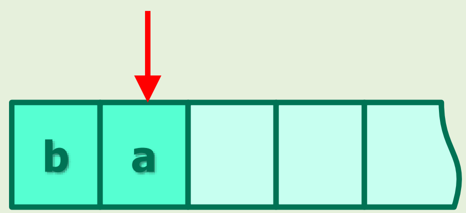
Process #1 halts in an accepting state AND
all symbols are consumed.

So, process #1 accepts w.



Process #2





Process #2 halts in an accepting state BUT all symbols are not consumed.

So, process #2 rejects w.

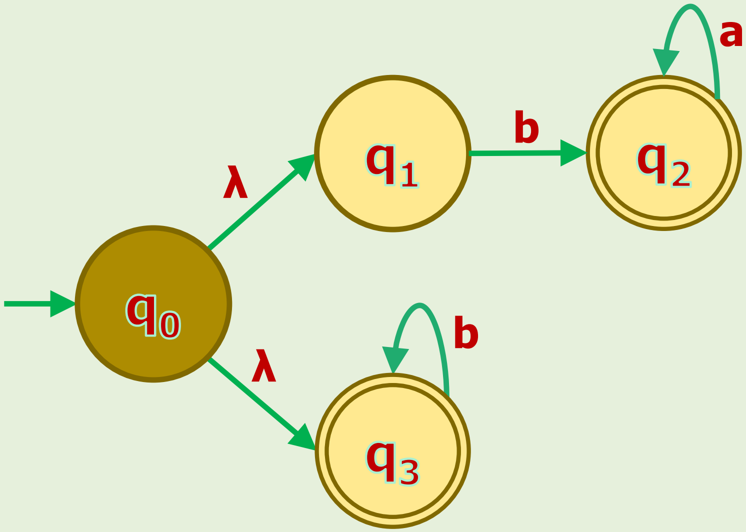
Homework

- Which of the following strings are accepted by this NFA over $\Sigma = \{a , b\}$?
- Draw all processes.

  w = b

  w = bb

  w = baa



6. Definitions

NFAs Transition Function

- We've learned that the range of NFAs can be zero, one, or more states ...
  - e.g.: $\delta(q_0, a) = \{\ \}$, $\delta(q_1, b) = \{q_5\}$, $\delta(q_2, a) = \{q_2, q_5\}$

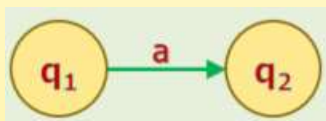
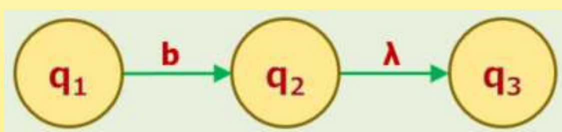
- Now the question is:
- How to change DFAs' transition function δ to be suitable for NFAs?
  - Recall that DFAs' transition function is defined as:

$$\delta: Q \times \Sigma \rightarrow Q$$

- We already know that 2^Q is the power set of Q and it contains all subsets of Q.
- Therefore, we change the range from Q to 2^Q.

$$\delta: Q \times \Sigma \rightarrow 2^Q$$

Transition Function: DFAs vs NFAs

Class	Transition	Sub-Rule Example Transition Function
DFAs		$\delta(q_1, a) = q_2$ $\delta : Q \times \Sigma \rightarrow Q$
NFAs		$\delta(q_1, b) = \{q_2, q_3\}$ $\delta(q_2, a) = \{\ \}$ $\delta : Q \times \Sigma \rightarrow 2^Q$

6. Formal Definition of NFAs

- An NFA M is defined by the quintuple (5-tuple):

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

- Where:
  - Q is a finite and nonempty set of states of the transition graph.
  - Σ is a finite and nonempty set of symbols called input alphabet.
  - δ is called transition function and is defined as:

$$\delta : Q \times \Sigma \rightarrow 2^Q$$

δ is total function.

  - $q_0 \in Q$ is the initial state of the transition graph.
  - $F \subseteq Q$ is the set of accepting states of the transition graph.

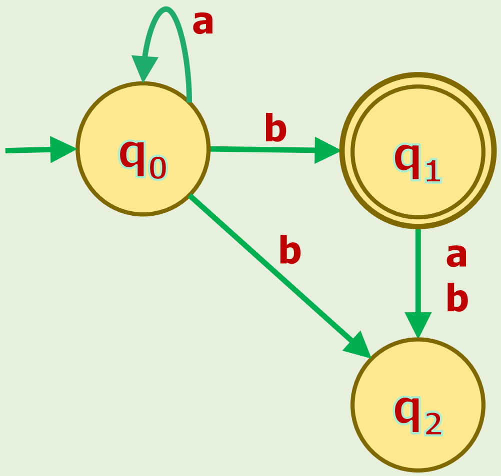
- Except δ, the rest items are the same as DFAs'.

Why δ is Total Function?

- The following example demonstrates why!

Example 15

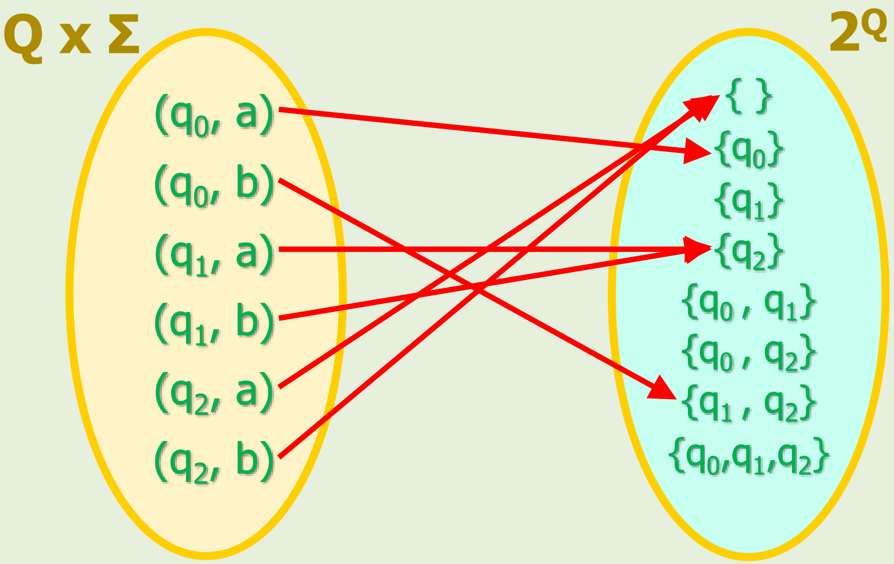
- Write the algebraic notation of the NFA's δ.



Solution

$$\begin{cases} \delta(q_0, a) = \{q_0\} \\ \delta(q_0, b) = \{q_1, q_2\} \\ \delta(q_1, a) = \{q_2\} \\ \delta(q_1, b) = \{q_2\} \\ \delta(q_2, a) = \{\,\} \\ \delta(q_2, b) = \{\,\} \end{cases}$$

- Draw the Venn diagram of δ.
- Isn't it total function?



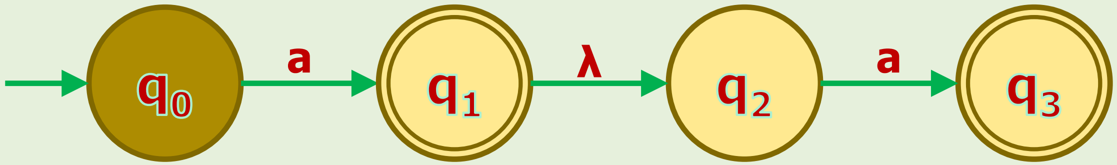
6. Formal Definitions: NFAs vs DFAs

	NFAs	DFAs
Transition function	$\delta : Q \times \Sigma \rightarrow 2^Q$	$\delta : Q \times \Sigma \rightarrow Q$
Examples	$\delta(q_1, a) = \{q_2, q_5, q_3\}$ $\delta(q_1, b) = \{q_1, q_3\}$ $\delta(q_2, a) = \{\ \}$	$\delta(q_1, a) = q_2$
Type of function	Total	Total
Type of processing	Parallel processing	Single processing

Associated Language to NFAs Examples

Example 16

- What is the associated language to the following NFA?



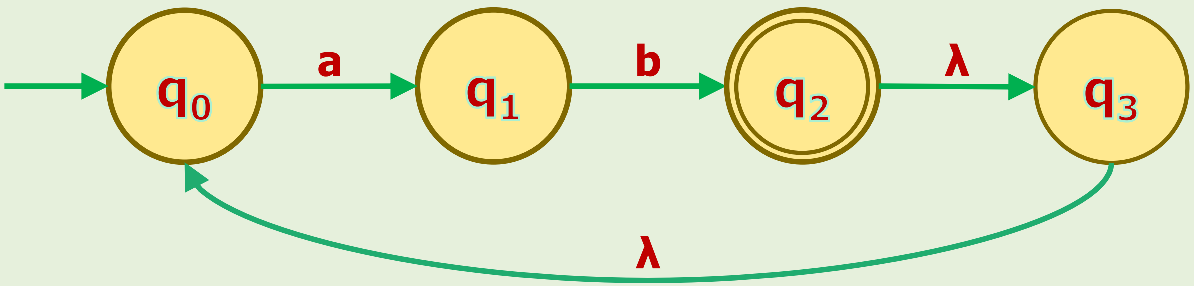
Solution

- L(M) = {a, aa}

Associated Language to NFAs Examples

Example 17

- What is the associated language to the following NFA?



Solution

- $L = \{ab, abab, ababab, \ldots\}$

  $= \{(ab)^n : n \geq 1\}$

NFA Design Example

Example 18

- Design a DFA and an NFA with 3 states for the following language over $\Sigma = \{a , b\}$.

"The set of all strings that ends with aa."

Homework

1. Let $L = \{a^n b : n \geq 0\}$, and $L' = L\,(L \cup \{\lambda\})$ over $\Sigma = \{a , b\}$.

   Design an NFA with 3 states for accepting L'.

2. Design an NFA for each of the following languages.

   a. $L = \{a^n b^m a^k : n, m \geq 0, k \geq 1\}$ with 3 states over $\Sigma = \{a , b\}$

   b. $L = \{(ab)^n (abc)^m : n \geq 0, m \geq 0\}$ over $\Sigma = \{a , b , c\}$

References

1. Linz, Peter, "An Introduction to Formal Languages and Automata, 5th ed.," Jones & Bartlett Learning, LLC, Canada, 2012
2. Michael Sipser, "Introduction to the Theory of Computation, 3rd ed.," CENGAGE Learning, United States, 2013
   ISBN-13: 978-1133187790