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

String 2018

Agenda of Day 10

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

About Midterm 1

2nd Announcement

- Midterm #1 (aka Quiz+)
  - Date: Thursday 03/01
  - Value: 10%
  - Topics: Everything covered from the beginning of the semester
  - Type: Closed y ∈ Material

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

- The cutoff for midterm #1 is the end of lecture 09.

Solution and Feedback of Quiz 3 (Out of 27)

Metrics	Section 1	Section 2	Section 3
Average	23	24	22
High Score	27	27	27
Low Score	17	19	15

Summary of Lecture 09: We learned ...

NFAs

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

NFAs Behavior

1. When an NFA has no choice ...
   - ... it halts.
2. When there are more than one choice, ...
   - ... it starts parallel processing.

When NFAs halt

- All input symbols are consumed. ≡ **c**

OR

- It has zero transition. ≡ **z**

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

Any question?

Summary of Lecture 09: We learned ...

NFAs

- A string is accepted if ...
  - ... at least one process accepts it.
  - For NFAs, **(h ∧ c ∧ f)** ↔ **a** is valid for accepting strings by one process.
  - Because **h** and **c** might have different values.
- A string is rejected if ...
  - ... all processes reject it.

Any question?

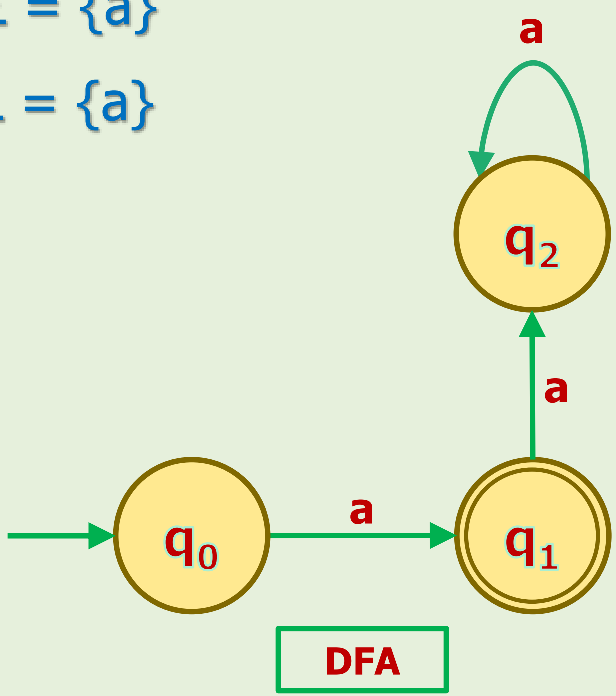
Why We Need a New Class

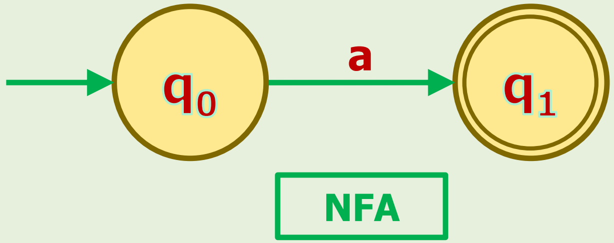
Revisited Question

- NFAs are interesting because their transition graphs are simpler.

Example 7

- $\Sigma = \{a\}$
- $L = \{a\}$





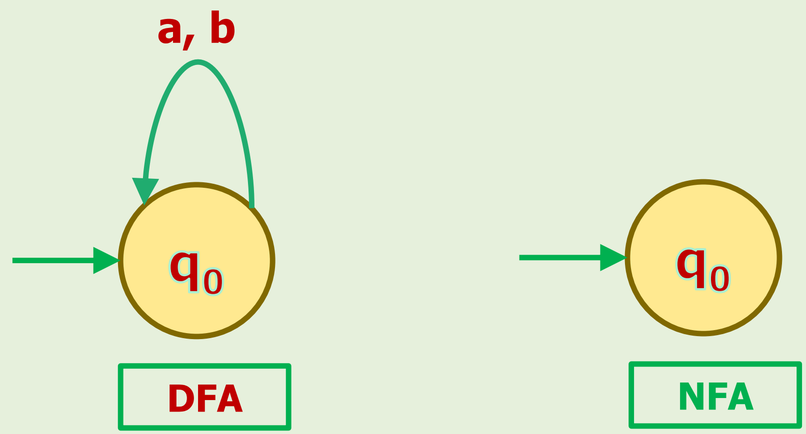
No trap needed!

Why We Need a New Class

Revisited Question

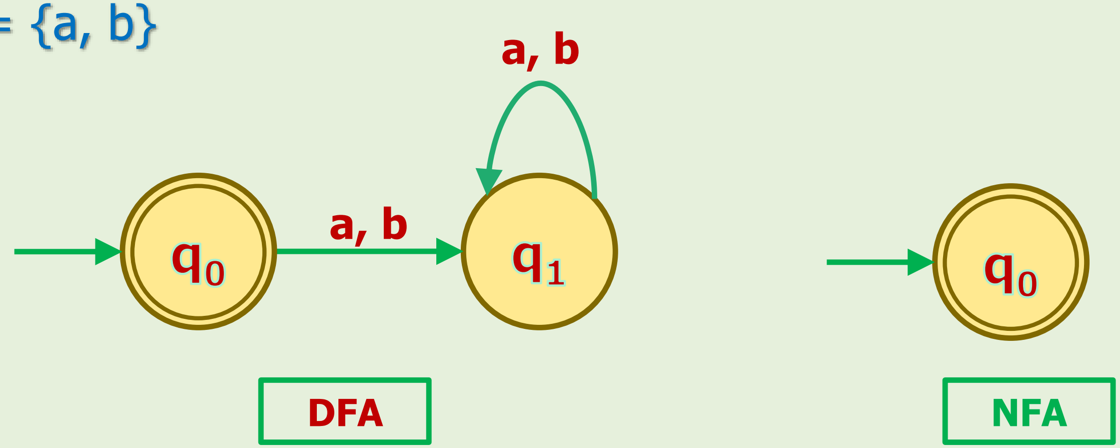
Example 8: Empty Language

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



Example 9: Empty String Language

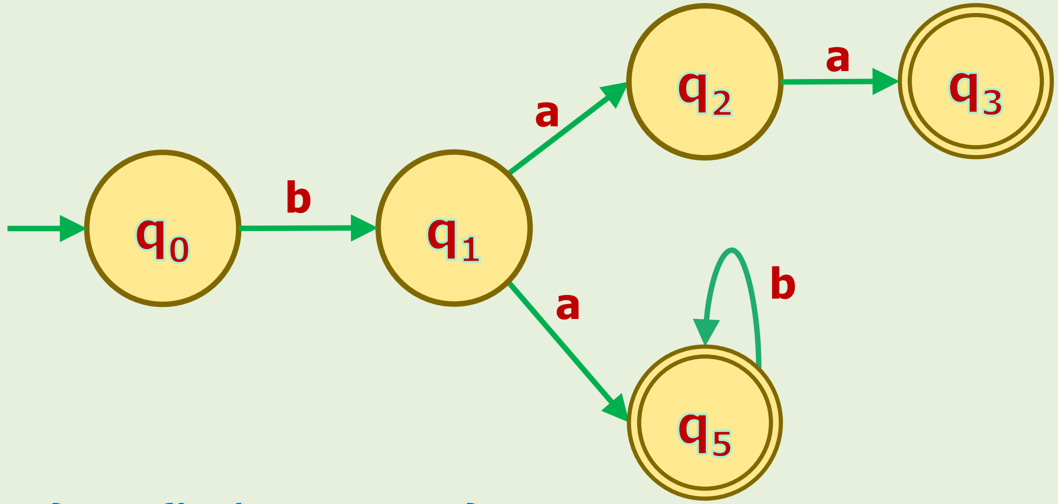
$L = \{\lambda\}$ over $\Sigma = \{a, b\}$



Associated Language to NFAs

Example 10

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



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

- Design this machine by DFA.

Lambda Transition

Introduction

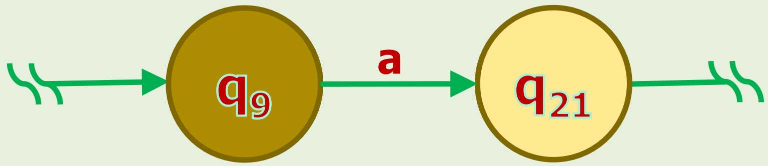
- So far, we learned two violations on DFAs.
- They encouraged us to introduce a new class of machines.
- We named it NFA.

- In this section, we'll talk about a special kind of transition.
- Another possible transitions that are strictly prohibited in DFAs ...
- But is allowed in NFAs.

Let's Shine our Knowledge

Question

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



Answer

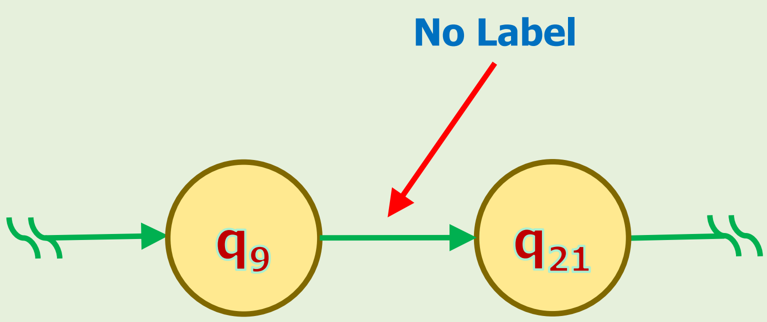
- If the machine is in q_9 AND the next input symbol is 'a', then the machine transits to q_{21}.

Conclusion

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

Let's Remove the Condition

- What would happen if we remove the condition?

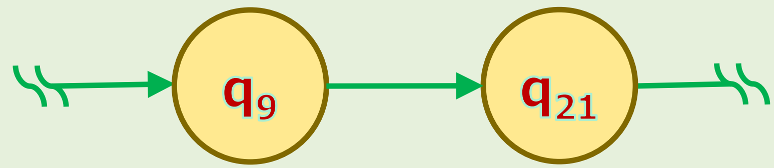


- This possible situation is called "short-circuit".
- A "short-circuit" is an edge with no label (symbol).

- What is the meaning of short-circuit?
- What is the behavior of the machine when it encounters it?

What is the Meaning of Short Circuit?

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

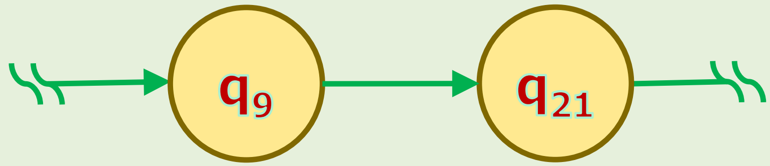


The machine can transit unconditionally!

- What would happen to the input tape?
- Does it need to consume any symbol?
- No, it doesn't!
- In fact, the control unit does not need to wait to receive the input symbol for deciding where to go.

How To Represent the Transition Function

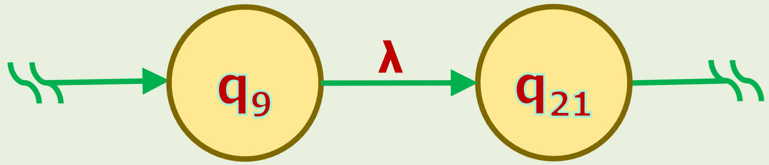
- What is the sub-rule of the following transition?



- The format of the sub-rules is: $\delta\ (q_9\ ,\ ?) = \{??\}$
- We should find out what to substitute for ? and ??.
- One suggestion for '?' is blank: $\delta\ (q_9\ ,\ ) = \{??\}$
- But this is not desirable format in math!
- We need to put something there.

How To Represent the Transition Function

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



- So, the sub-rule would be:

$$\delta\ (q_9\ ,\ \lambda) = \{??\}$$

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

- We still need to find out what is the value of ??.

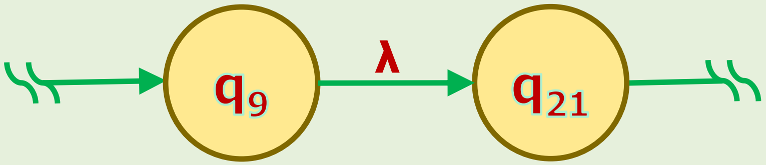
Another Look to Meaning of λ

- Before going further, note that:
- We've already used λ to represent "empty string".
- In fact λ means "NO symbol". (Empty String = NO Symbol)
- And short-circuit has "no symbol".
- 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!

How To Represent the Transition Function

$$\delta(q_9, \lambda) = \{??\}$$



- What is the value of ??.
- Since the machine may transit unconditionally, it means that ...

  it may stay as well.

- So, when a machine encounters a λ-transition,

  it may stay or it may transit.

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

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

λ-Transition Definition

- It's time to define λ-transition officially.

Definition

- λ-transition in automata theory means:

  The machine may "unconditionally" transit.

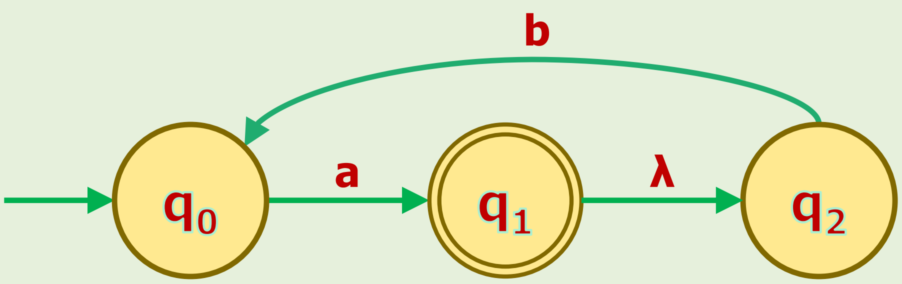
- Note that this is a general definition for all types of automata.
- Also note that λ-transition concept changes our view about sub-rules.

- Let's take an example.

How To Represent the Transition Function

Example 11

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



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

NFAs Transition Function

- We should change the transition function definition to accommodate λ-transitions.
- Recall that we concluded NFAs' transition function to be:

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

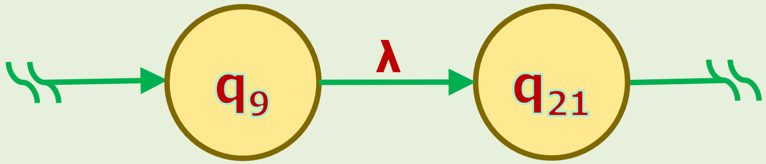
- But now, we have a new symbol for λ-transition.
- So, we just need to add it to Σ.

$$\delta: Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$$

- We'll mention it again when we are giving the formal definition of NFAs.

How NFAs Behave If They Encounter λ-transitions

- Let's review the example one more time:



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

- 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".
  - In other words, for every possible choice, it initiates a new independent process and every process independently continues processing the string.

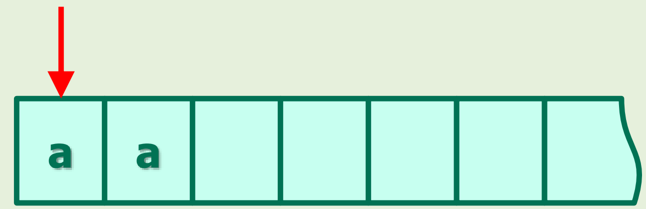
- Let's see some practical examples of λ-Transitions!

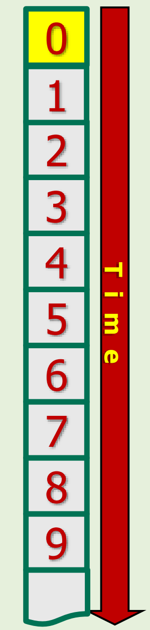
λ-Transitions in Action

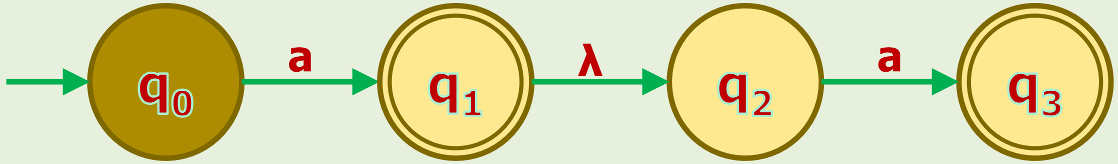
λ-Transitions in Action

Example 12: Process #1

- $\Sigma = \{a\}$
- $w = aa$
- $\delta(q_0, a) = \{q_1\}$



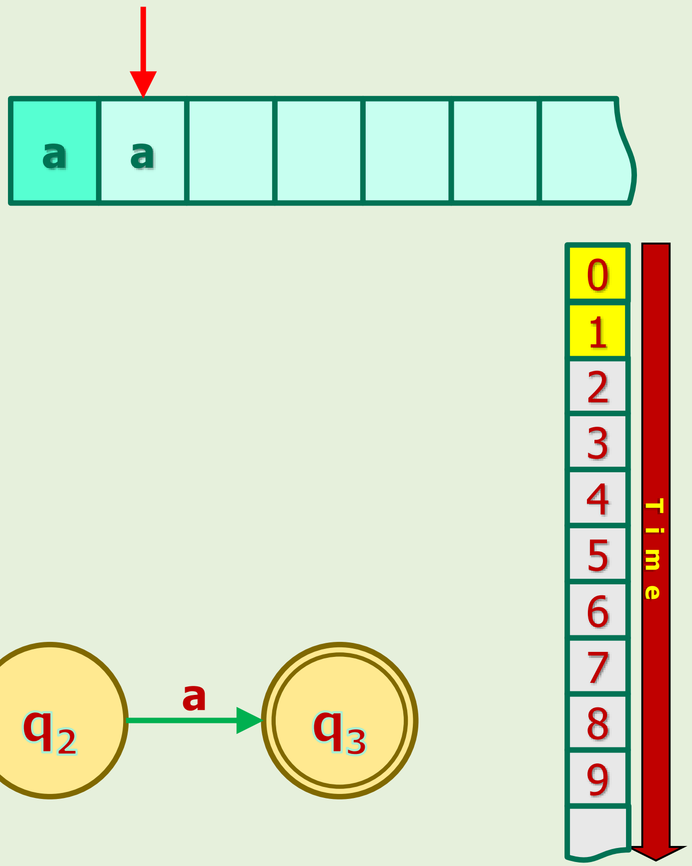


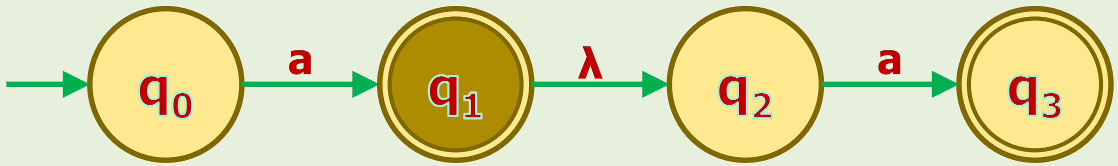


λ-Transitions in Action

Example 12: Process #1

- $\Sigma = \{a\}$
- $w = aa$
- $\delta(q_1, \lambda) = \{q_1, q_2\}$



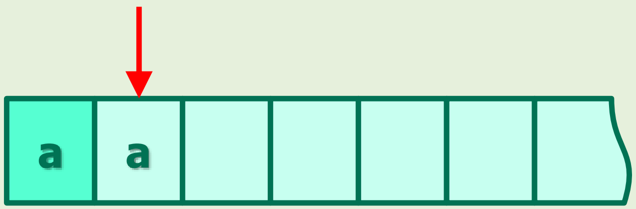


- The NFA encounters a λ-transition.
- Process #2 is initiated.
- Initial configuration: state=q_2, rest of input=a, clock=1

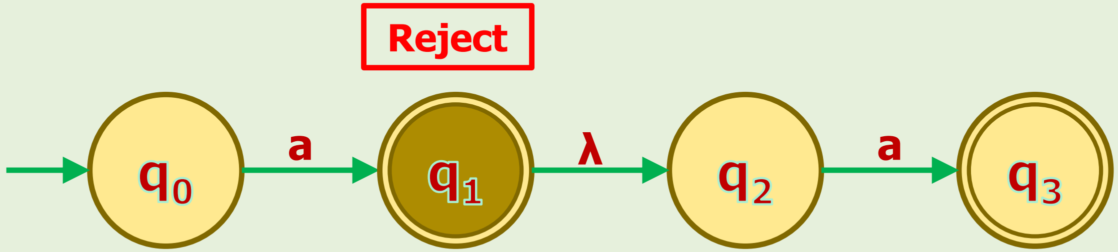
λ-Transitions in Action

Example 12: Process #1

- $\Sigma = \{a\}$
- $w = aa$



- Process #1 cannot continue because it has no choice for symbol 'a' in q_1.
- So, it halts in q_1.

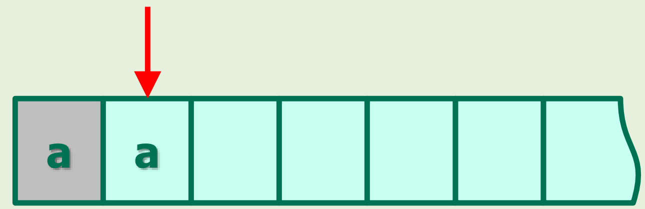


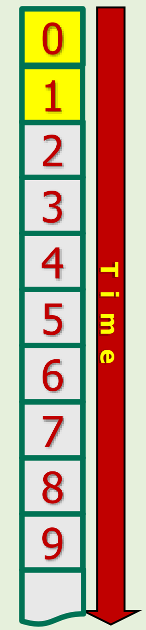
- Some symbols are NOT consumed, so, process #1 rejects the string.

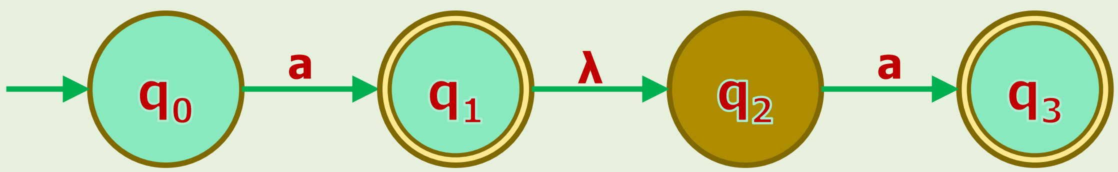
λ-Transitions in Action

Example 12: Process #2

- $\Sigma = \{a\}$
- $w = aa$
- Initial config: state=q_2, rest of input=a, clock=1



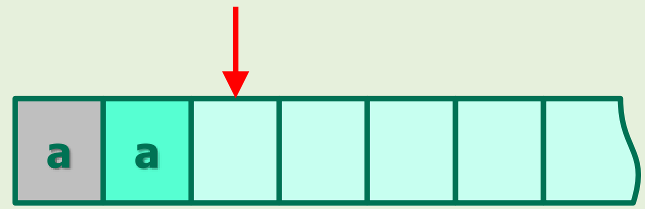


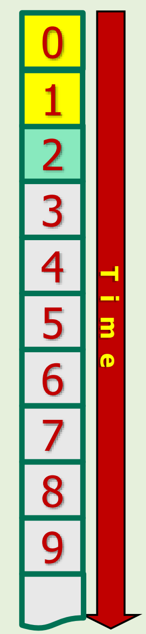


λ-Transitions in Action

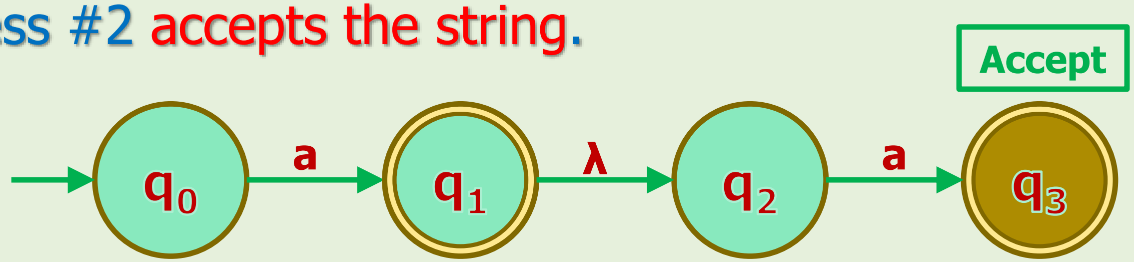
Example 12: Process #2

- $\Sigma = \{a\}$
- w = aa





- All symbols are consumed.
- The machine halts in an accepting state.
- So, process #2 accepts the string.

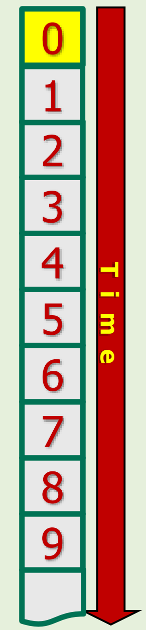


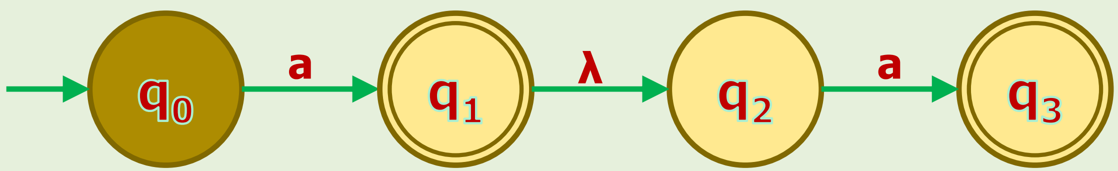
- Recap: if one process accepts a string, then the string is accepted.
- So, overall, the string aa is accepted.

λ-Transitions in Action

Example 13: Process #1

- $\Sigma = \{a\}$
- $w = aaa$

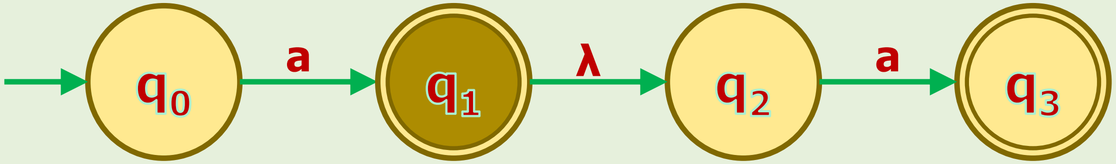




λ-Transitions in Action

Example 13: Process #1

- $\Sigma = \{a\}$
- w = aaa
- Again, the NFA initiates another process because of the λ-transition.
- Initial config: state=q_2, rest of input=aa, clock=1
- Process #1 continues.

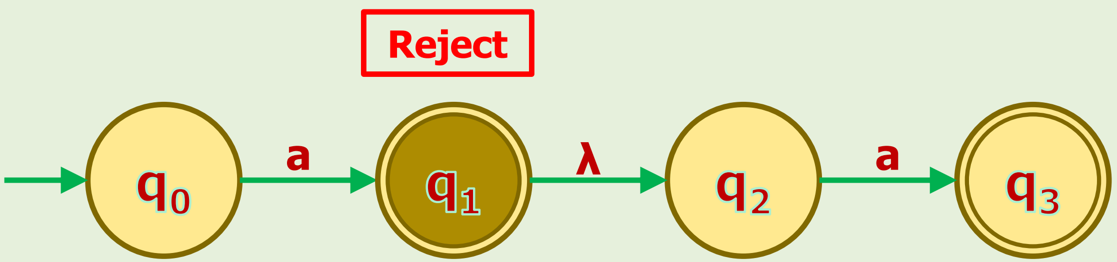


λ-Transitions in Action

Example 13: Process #1

- $\Sigma = \{a\}$
- $w = aaa$

- Process #1 cannot continue any longer because it has no choice for symbol 'a' in q_1.

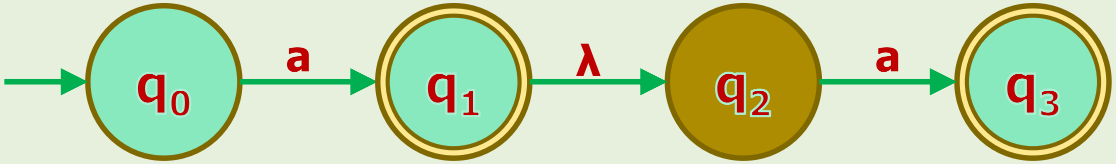


- So, process #1 rejects the string because some symbols are not consumed.

λ-Transitions in Action

Example 13: Process #2

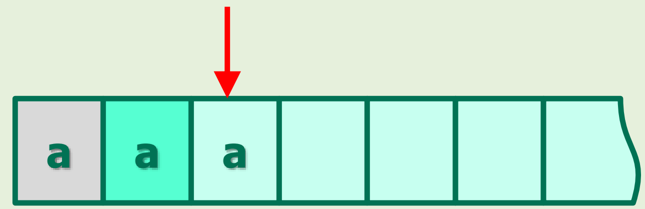
- $\Sigma = \{a\}$
- $w = aaa$
- Initial config: state=q_2, rest of input=aa, clock=1

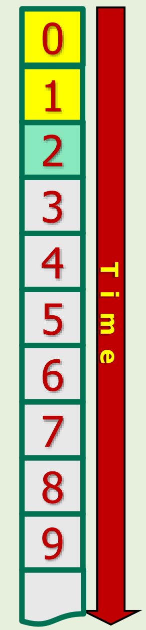


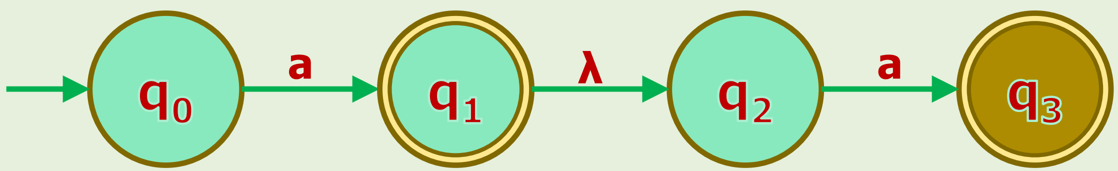
λ-Transitions in Action

Example 13: Process #2

- $\Sigma = \{a\}$
- $w = aaa$



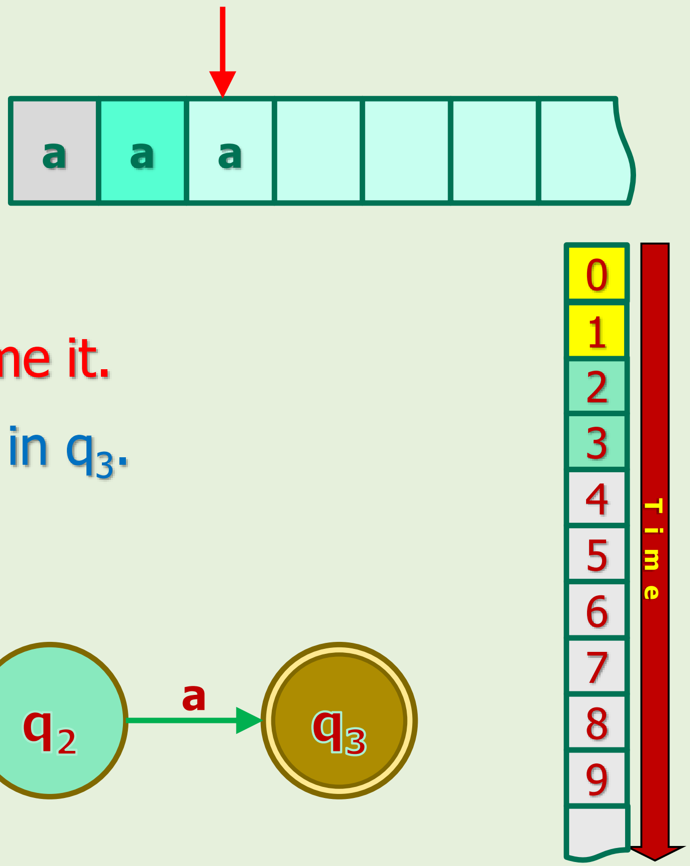




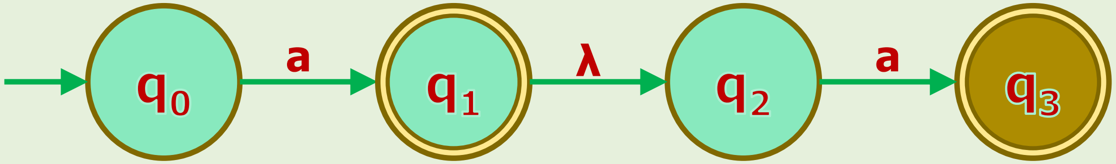
λ-Transitions in Action

Example 13: Process #2

- $\Sigma = \{a\}$
- w = aaa



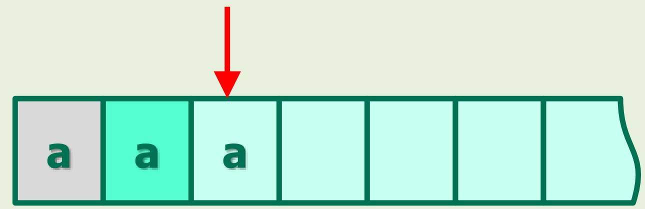
- Process #2 reads 'a' but cannot consume it.
- Because it has no choice for symbol 'a' in q_3.

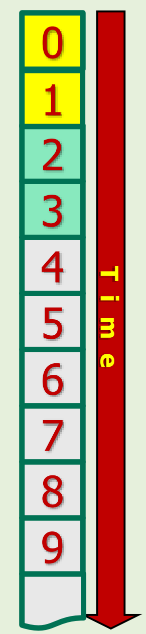


λ-Transitions in Action

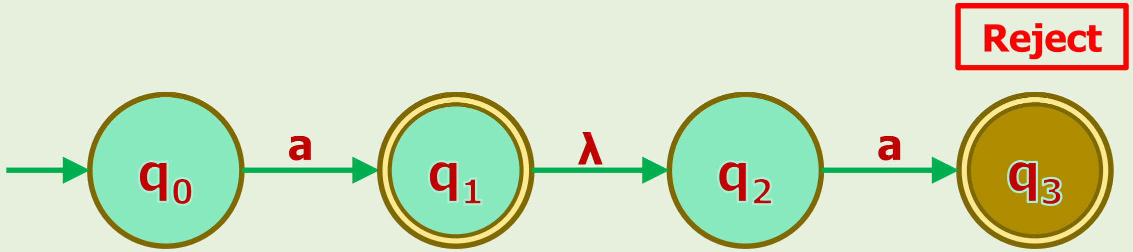
Example 13: Process #2

- $\Sigma = \{a\}$
- $w = aaa$





- Some input symbols are NOT consumed.
- No matter where the machine halted, the string is rejected.

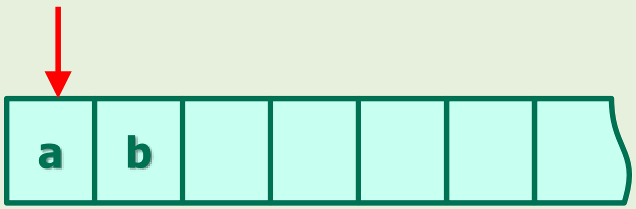


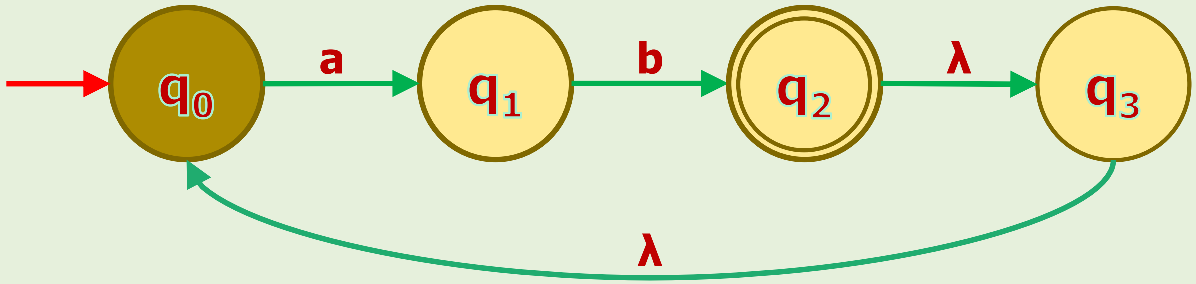
- Both processes rejected the string. So, aaa is rejected.

λ-Transitions in Action

Example 14: Process #1

- $\Sigma = \{a, b\}$
- $w = ab$
- $\delta(q_0, a) = \{q_1\}$

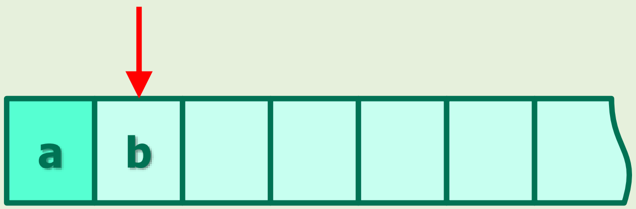


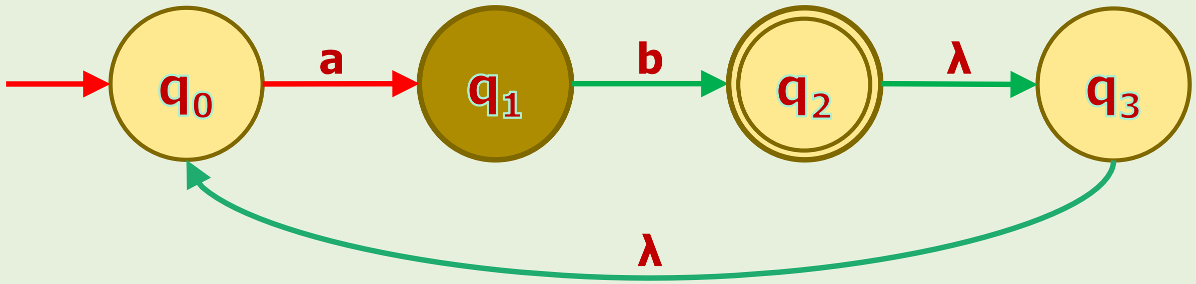


λ-Transitions in Action

Example 14: Process #1

- $\Sigma = \{a, b\}$
- $w = ab$
- $\delta(q_1, b) = \{q_2\}$

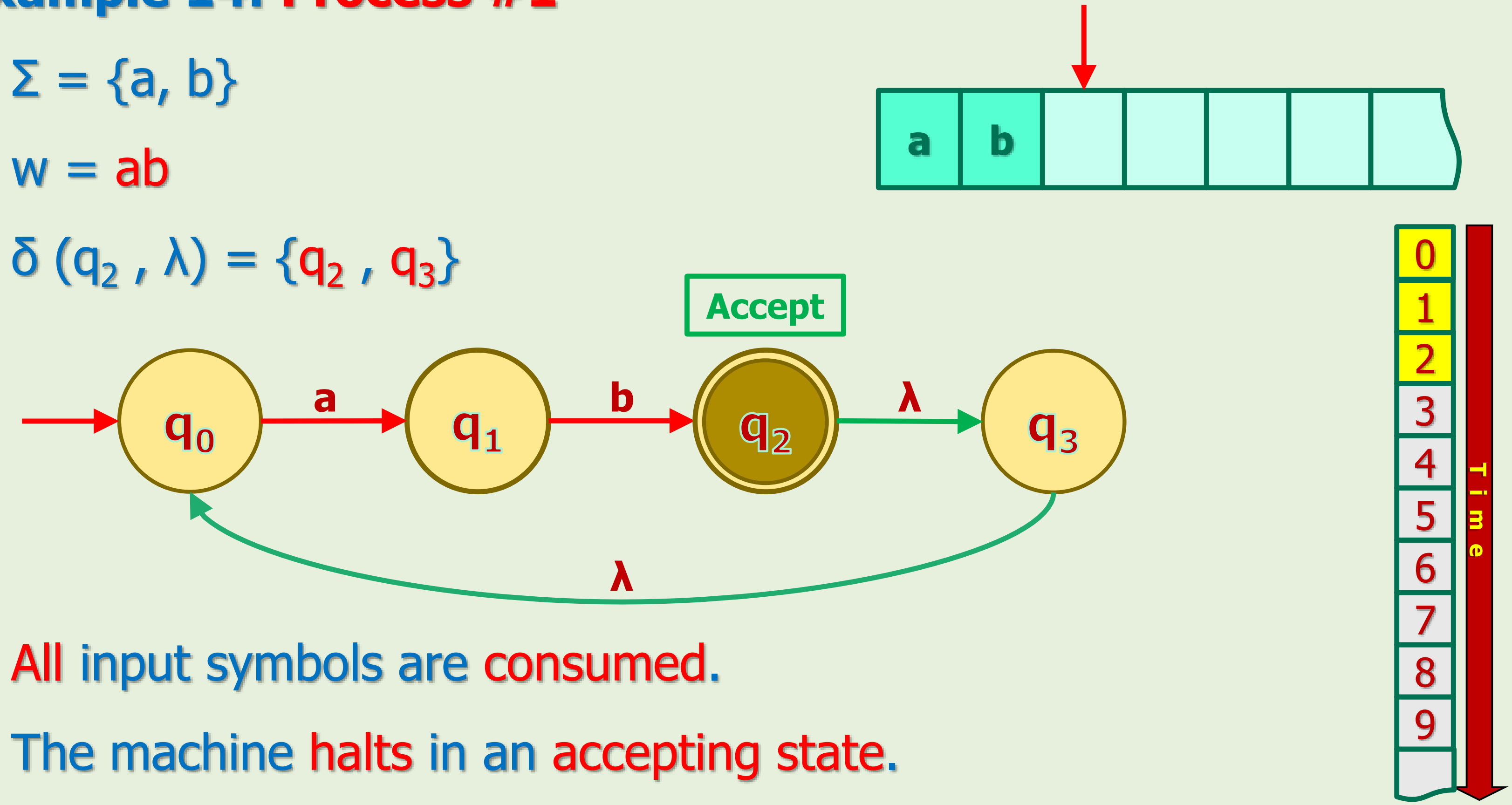




λ-Transitions in Action

Example 14: Process #1

- $\Sigma = \{a, b\}$
- $w = ab$
- $\delta(q_2, \lambda) = \{q_2, q_3\}$

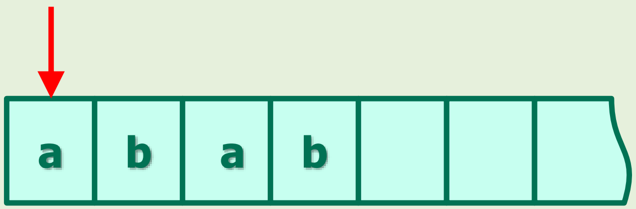


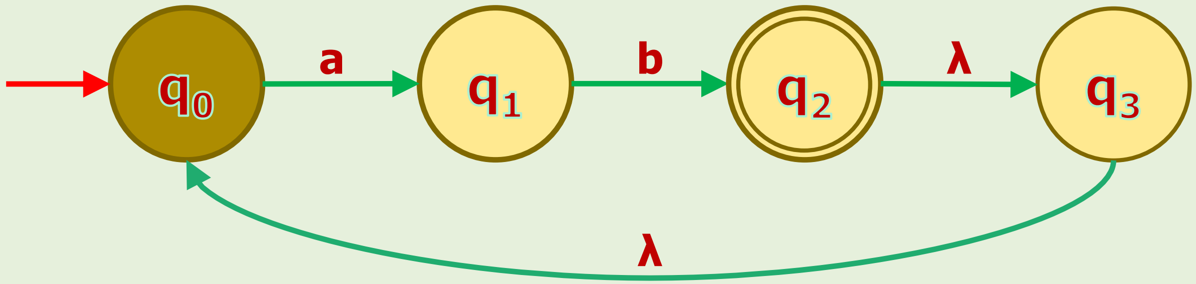
- All input symbols are consumed.
- The machine halts in an accepting state.
- So, the string is accepted.
- It does not need to initiate another process!

λ-Transitions in Action

Example 15: Process #1

- $\Sigma = \{a, b\}$
- $w = abab$
- $\delta(q_0, a) = \{q_1\}$

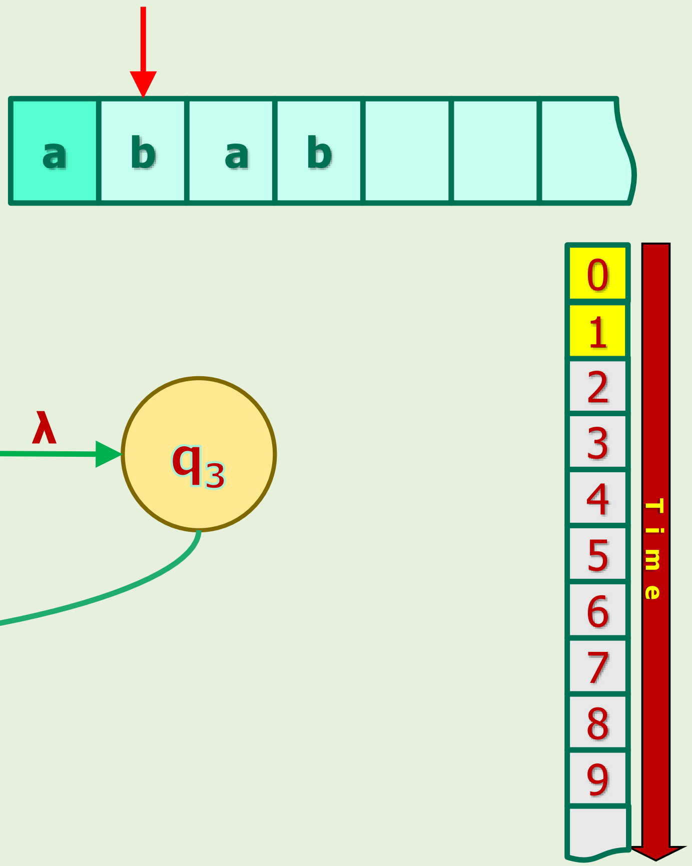


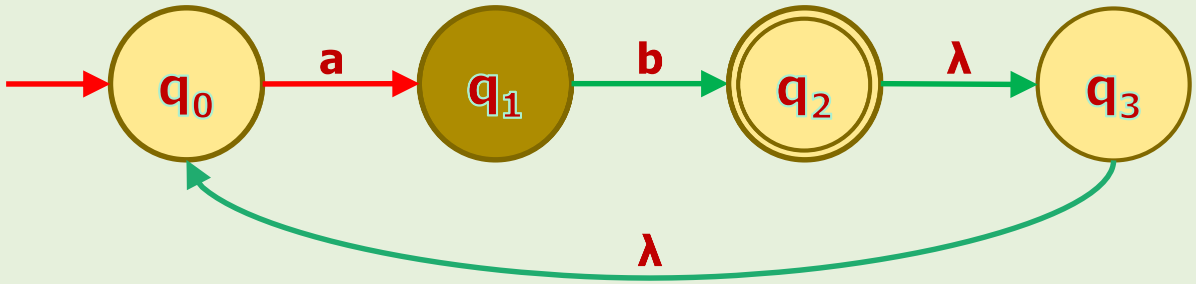


λ-Transitions in Action

Example 15: Process #1

- $\Sigma = \{a, b\}$
- $w = abab$
- $\delta(q_1, b) = \{q_2\}$

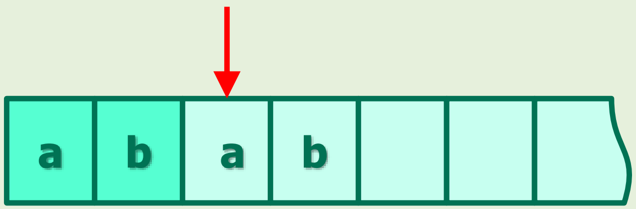


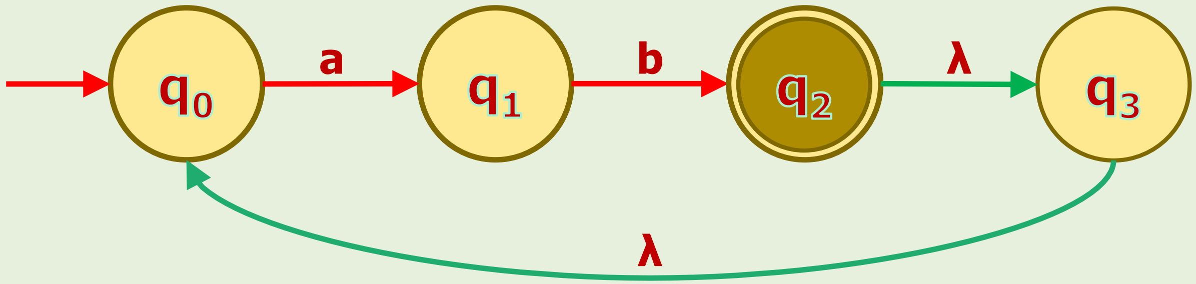


λ-Transitions in Action

Example 15: Process #1

- $\Sigma = \{a, b\}$
- $w = abab$
- $\delta(q_2, \lambda) = \{q_2, q_3\}$



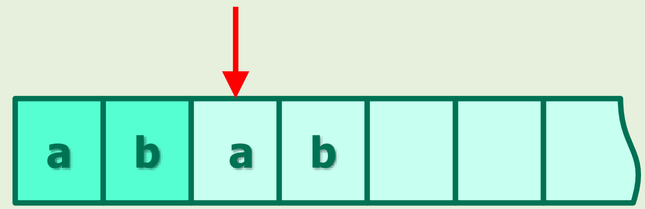


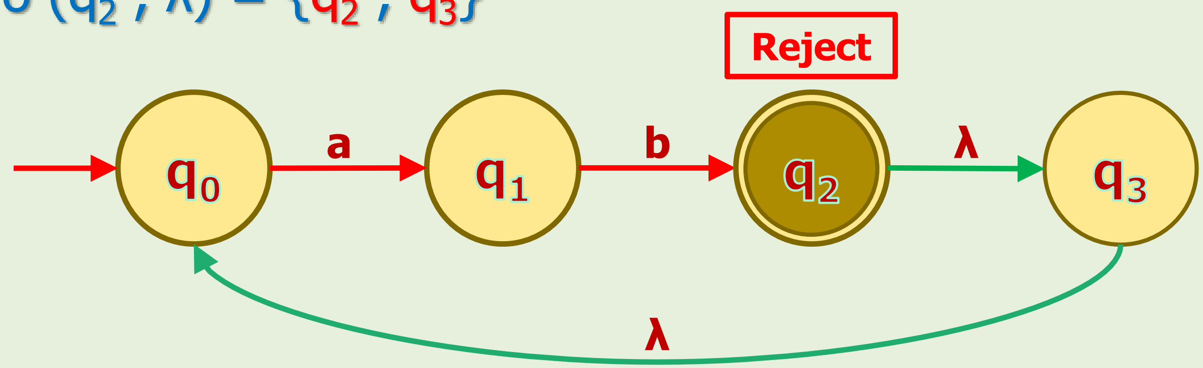
- It initiates process #2 because of λ-transition.
- Initial config: state=q_3, rest of input=ab, clock=2

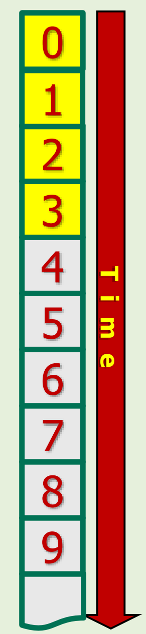
λ-Transitions in Action

Example 15: Process #1

- $\Sigma = \{a, b\}$
- $w = abab$
- $\delta(q_2, \lambda) = \{q_2, q_3\}$



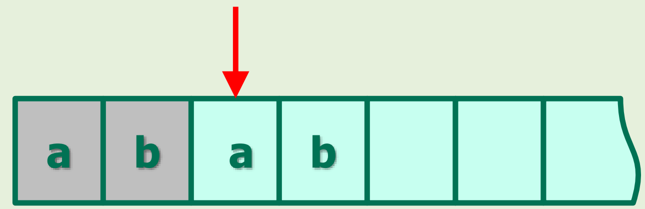




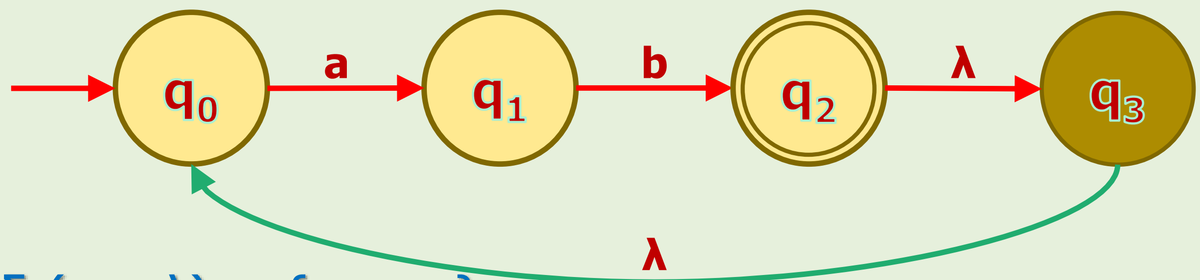
- Process 1 reads 'a' but cannot consume it.
- Some input symbols are NOT consumed.
- So, process #1 rejects the string.

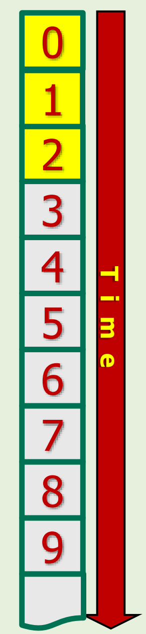
λ-Transitions in Action

Example 15: Process #2



- $\Sigma = \{a, b\}$
- w = abab
- Initial config: state=q_3, rest of input=ab, clock=2



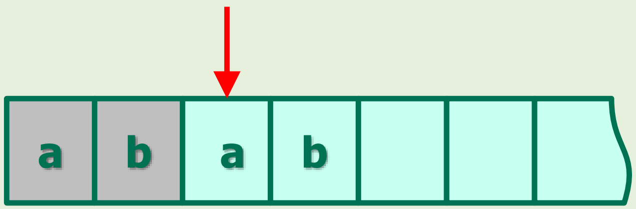


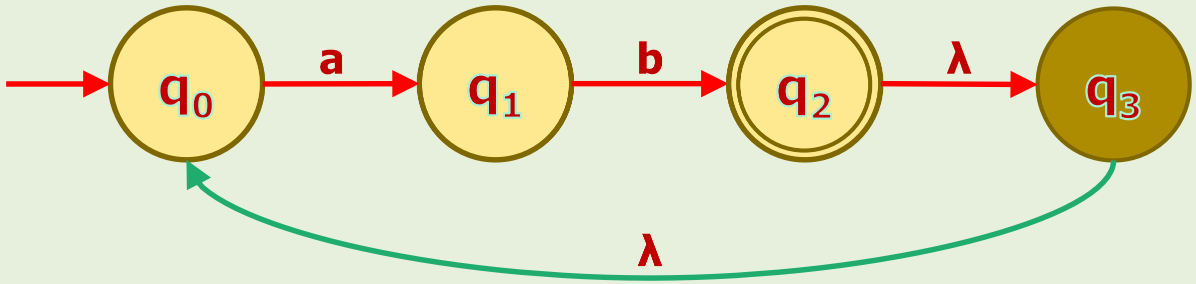
- $\delta(q_3, \lambda) = \{q_3, q_0\}$
- It encounters another λ-transition.
- It initiates process #3 because of λ-transition.
- Initial config: state=q_0, rest of input=ab, clock=2

λ-Transitions in Action

Example 15: Process #2

- $\Sigma = \{a, b\}$
- $w = abab$



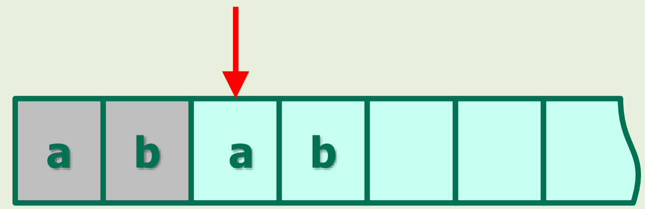


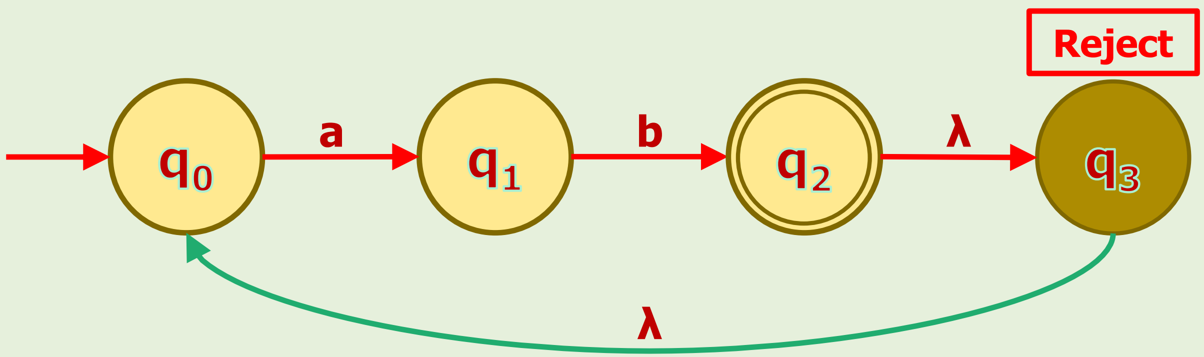
- Process #2 reads 'a' but cannot consume it.
- So, it has to halt.

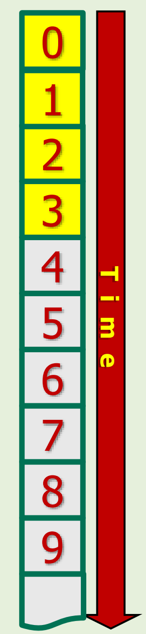
λ-Transitions in Action

Example 15: Process #2

- $\Sigma = \{a, b\}$
- w = abab





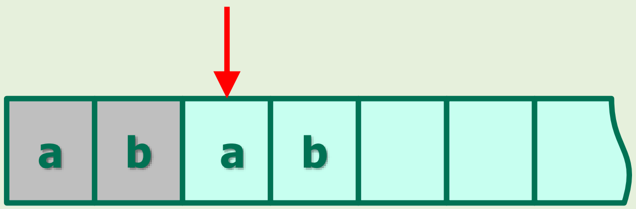


- Process #2 rejects the string because some input symbols are NOT consumed.

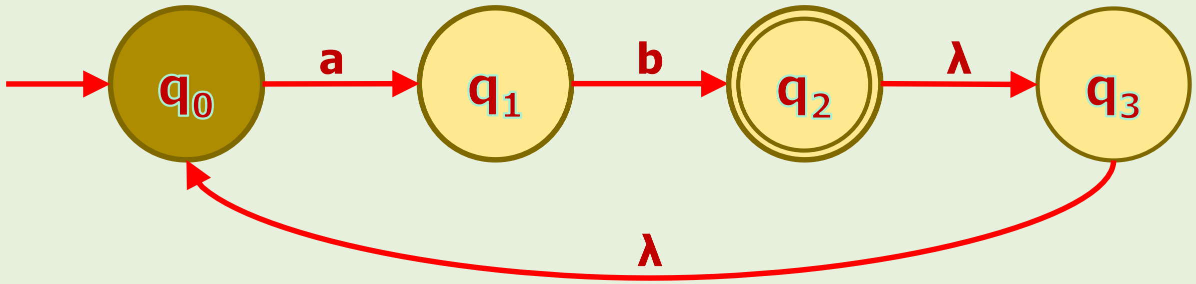
λ-Transitions in Action

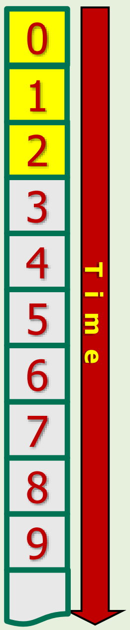
Example 15: Process #3

- $\Sigma = \{a, b\}$
- w = abab



- Initial config: state=q_0, rest of input=ab, clock=2



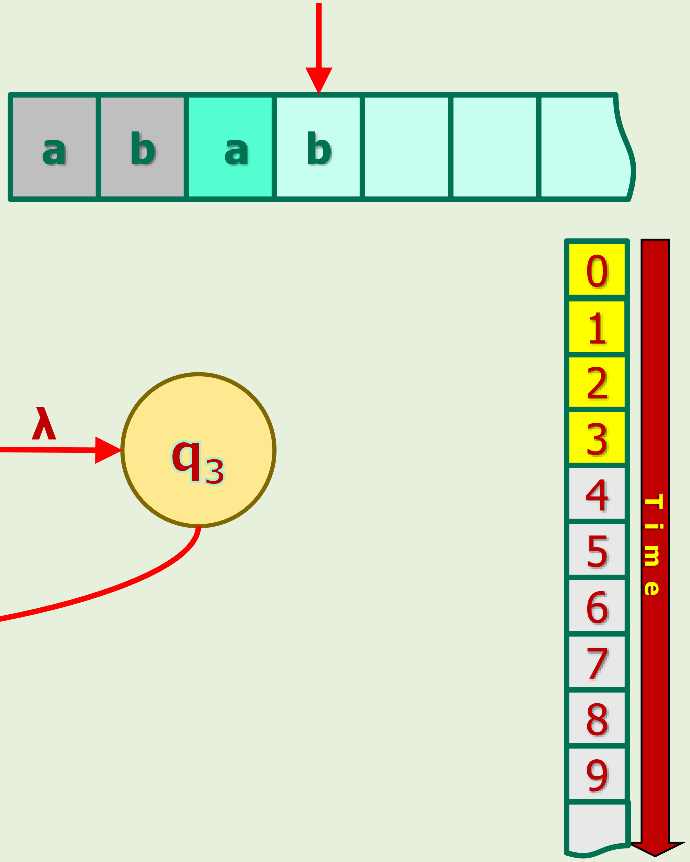


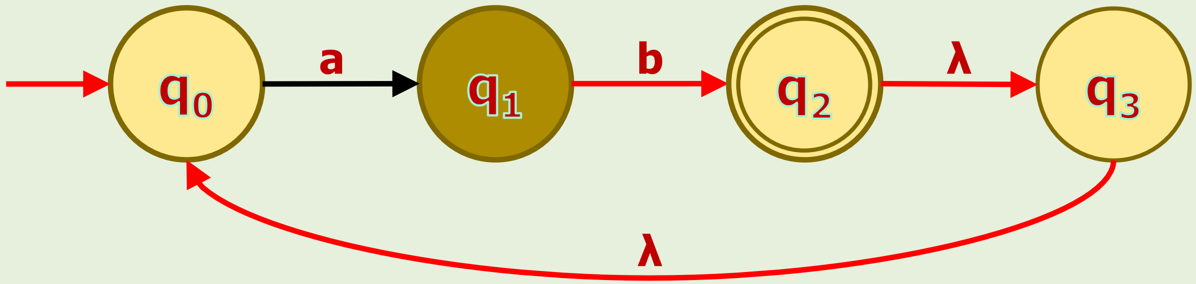
- $\delta(q_0, a) = \{q_1\}$

λ-Transitions in Action

Example 15: Process #3

- $\Sigma = \{a, b\}$
- $w = abab$
- $\delta(q_1, b) = \{q_2\}$

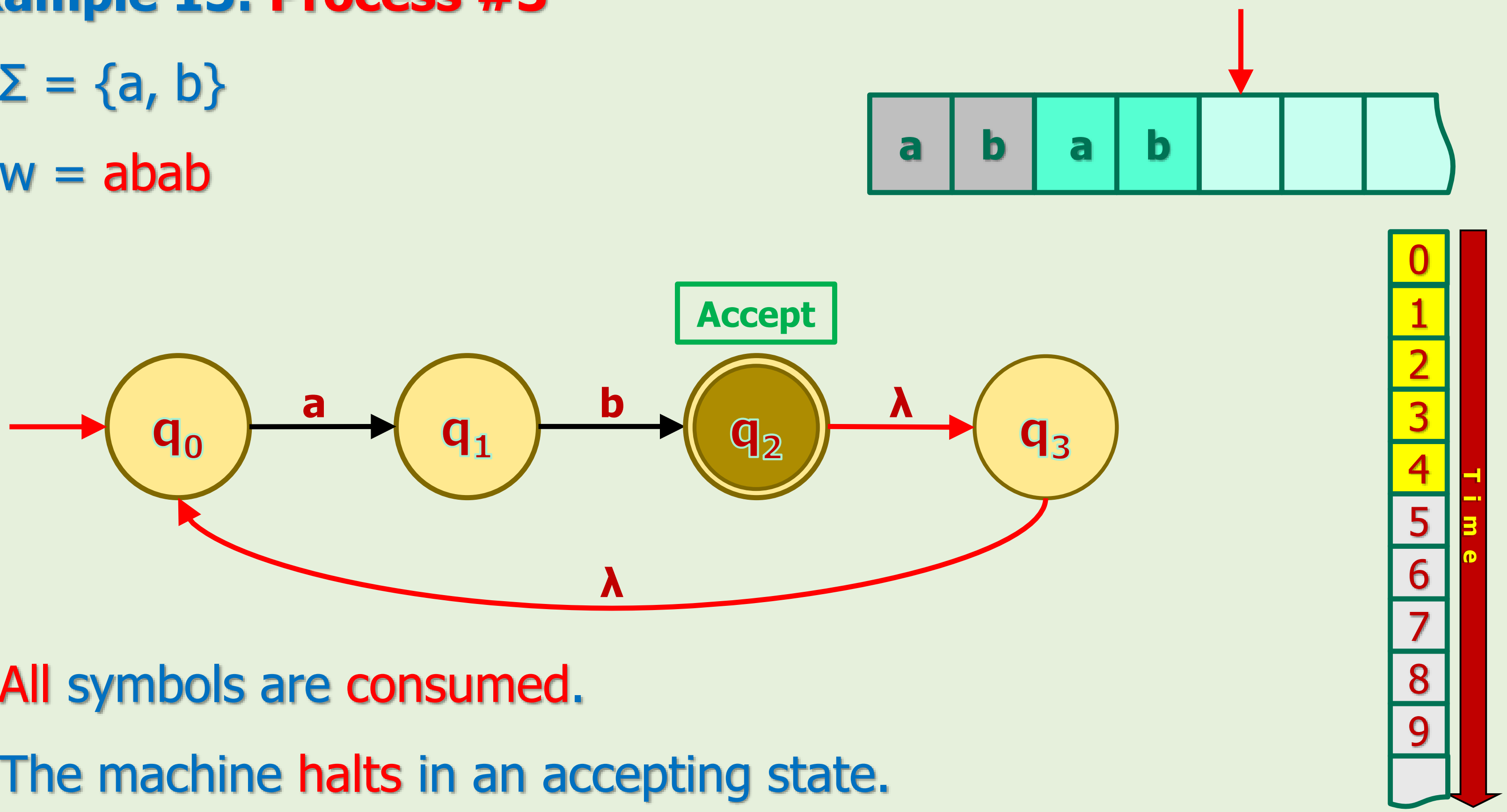




λ-Transitions in Action

Example 15: Process #3

- $\Sigma = \{a, b\}$
- $w = abab$



- All symbols are consumed.
- The machine halts in an accepting state.
- So, the string is accepted.
- It does not need to initiate another process!

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