San José State University
Department of Computer Science

Ahmad Yazdankhah

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

Non-Regular Languages

(Part 2)

Lecture 25

Day 29/31

CS 154

Formal Languages and Computability

Spring 2018

Agenda of Day 29

- Summary of Lecture 24
- Quiz 10
- Lecture 25: Teaching ...
  - Non-Regular Languages (Part 2)

Summary of Lecture 24: We learned ...

Non-Regular Languages

- The main question is:

  How to prove a language is non-regular?

- We introduced an important theorem called "pumping lemma".

- Pumping lemma is NOT applicable to "finite languages".

- Pumping lemma is an important property of
  infinite regular languages.

Pumping Lemma

If L is an infinite regular language,

Then

there exists an $m \geq 1$ such that

If $w \in L$ and $|w| \geq m$

Then //P. L. guarantees that ...

We must be able to divide w into xyz in such a way that all of the following conditions are satisfied:

$|xy| \leq m$, and

$|y| \geq 1$, and

$w_i = x\, y^i\, z \in L$

for $i = 0, 1, 2, \ldots$.

NAME	Alan M. Turing		
SUBJECT	CS 154	TEST NO.	10
DATE	05/03/2018	PERIOD	1 / 2 / 3

TEST RECORD	
PART 1	123
PART 2	
TOTAL	

Your list # goes here!

Quiz 10

No Scantron

Application of Pumping Lemma

How to Prove a Language is Non-Regular?

- Use "proof by contradiction"

1. Assume L is regular. So, the pumping lemma should hold for L.
2. Apply pumping lemma
3. Find a contradiction.
4. Then, blame your assumption and conclude that L must be non-regular.

- Let's take some examples!

Applications of Pumping Lemma

Example 7

- Prove $L = \{a^n b^n : n \geq 0\}$ is non-regular language.

Proof

Applications of Pumping Lemma

Example 8

- Prove $L = \{ww : w \in \{a, b\}^*\}$ is non-regular language.

Proof

Homework

- Prove that the following languages are non-regular:
  1. $L = \{ww^R : w \in \{a, b\}^*\}$
  2. $L = \{a^n b^n c^n : n \geq 0\}$
  3. $L = \{www : w \in \{a, b\}^*\}$
  4. $L = \{a^n b^k c^{n+k} : n \geq 0, k \geq 0\}$

Pigeonhole Principle

Pigeonhole Principle

Recap

Example 9

- If we have 10 pigeons and 9 pigeonholes (boxes), then one pigeonhole must contain more than one pigeon.

Pigeonhole Principle

If we put n objects (pigeon) into m boxes (pigeonholes) &&

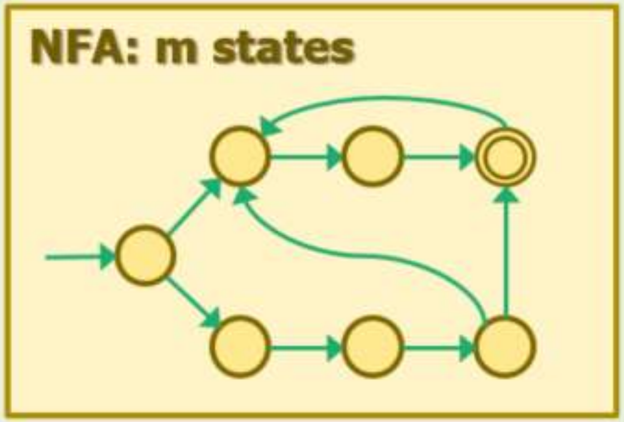
n > m

∴ At least one box must have more than one object in it.

- Reference: https://en.wikipedia.org/wiki/Pigeonhole_principle

Pigeonhole Principle and DFA's

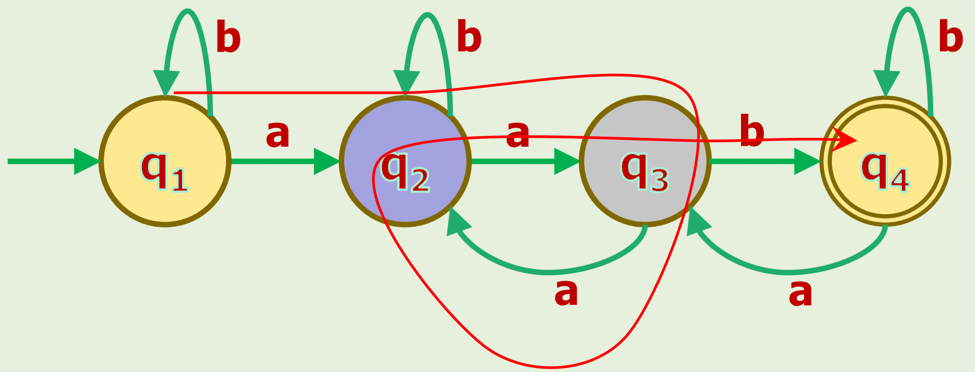
???



Pigeonhole Principle and DFA's

Example 10

- Given following DFA with 4 states.



- Consider the walk of w = aaaab. (|w| = 5)

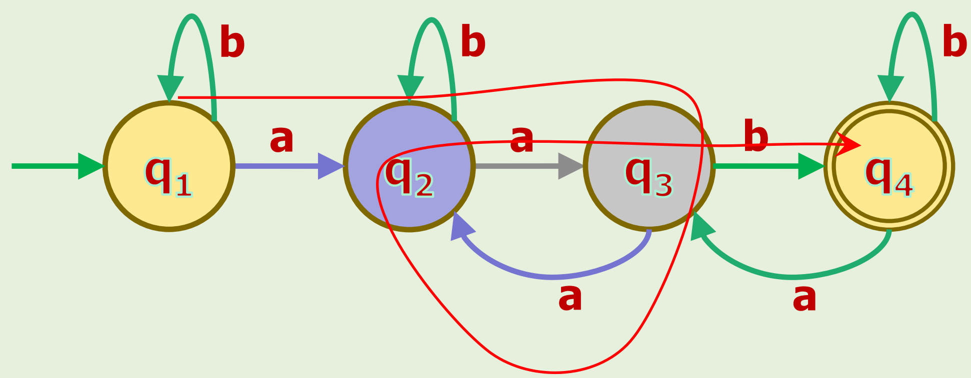
- Can we conclude that:

  At least one state must be visited more than once.

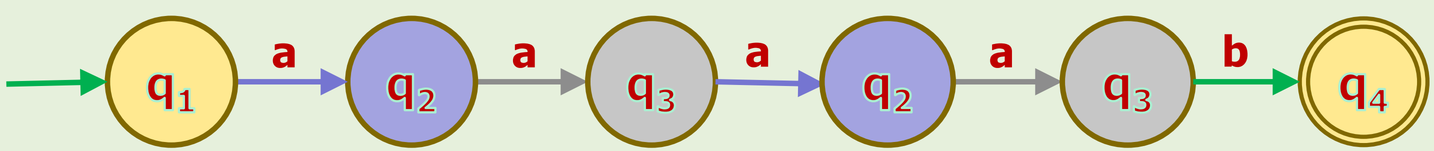
- Yes, because the size of the string is bigger than the number of states.

Pigeonhole Principle and DFA's

Example 10 (cont'd) w = aaaab

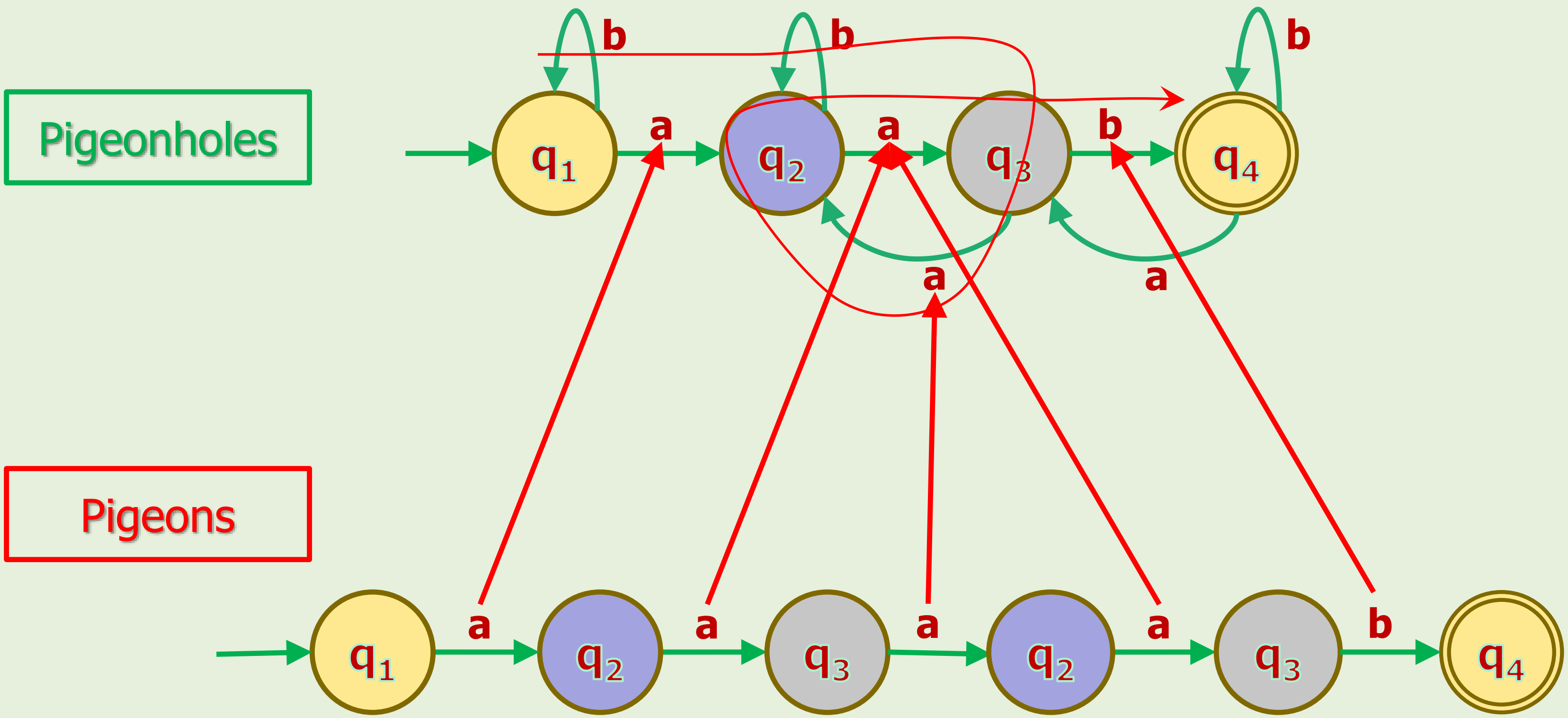


- Now, let's show the walk by one-dimensional projection method to investigate our guess.



- q_2 and q_3 are visited twice.

What is Pigeon and what is Pigeonhole?



- Pigeons = the symbols of the string
- Pigeonholes = the transition plus next states
- To simplify the pigeonholes, it is easier to consider only the states.

Pigeonhole Principle and DFA's

In general

If a DFA has m states, and

we process a string w with $|w| \geq m$,

then by the pigeonhole principle,

at least one state will be visited more than once.

Visualizing Pumping Lemma

Visualizing Pumping Lemma

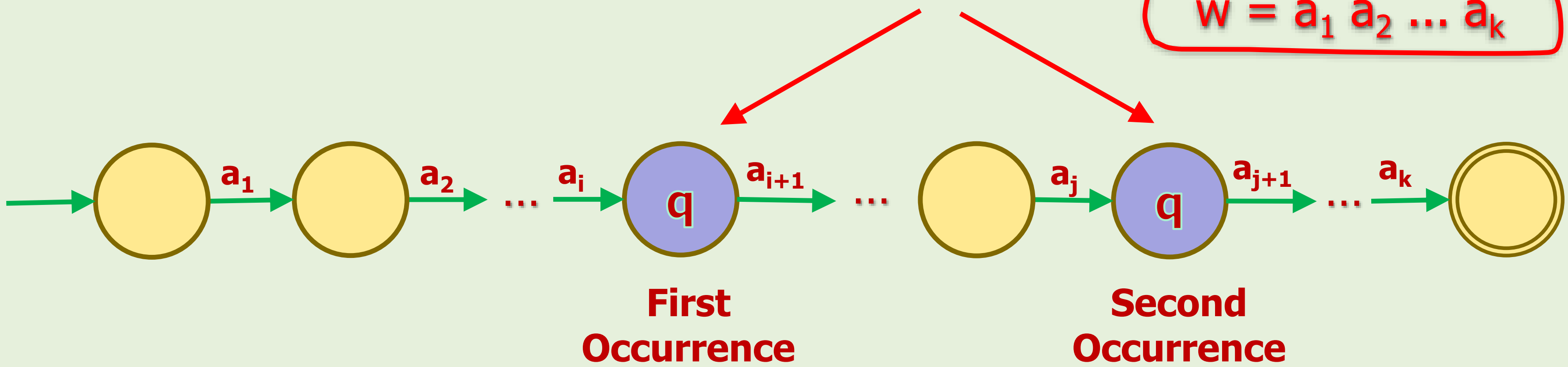
- Consider L as an infinite regular language.
- Since L is regular, so, there exists a DFA M that accepts it.
- Let's assume this DFA has m states (that should be finite) .
- Take a string $w = a_1 a_2 \dots a_k \in L$ whose size is $|w| \geq m$.
- Since $|w| \geq m$, therefore, in the walk of w,

  at least one state is visited more than once.

Visualizing Pumping Lemma

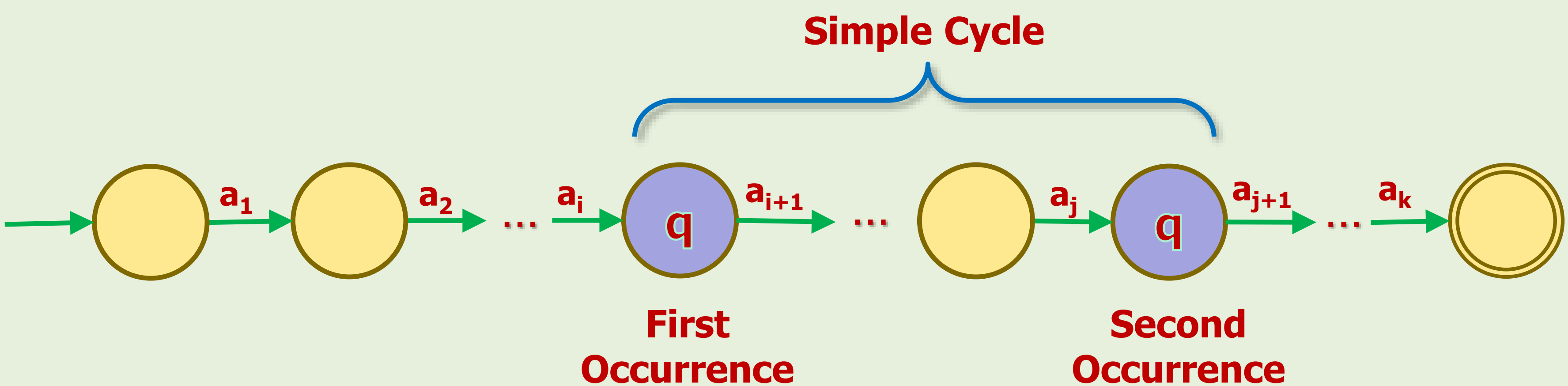
- The following graph is the one-dimensional projection of w.
- Why is the last state "accepting state"?
  - Since $w \in L$, so, the last state should be an "accepting state".
- Can there be accepting states in the middle too?
  - Yes! It does not bother what we want to show.
- We assume the first repeated state is "q".

$w = a_1\ a_2\ \ldots\ a_k$



Visualizing Pumping Lemma

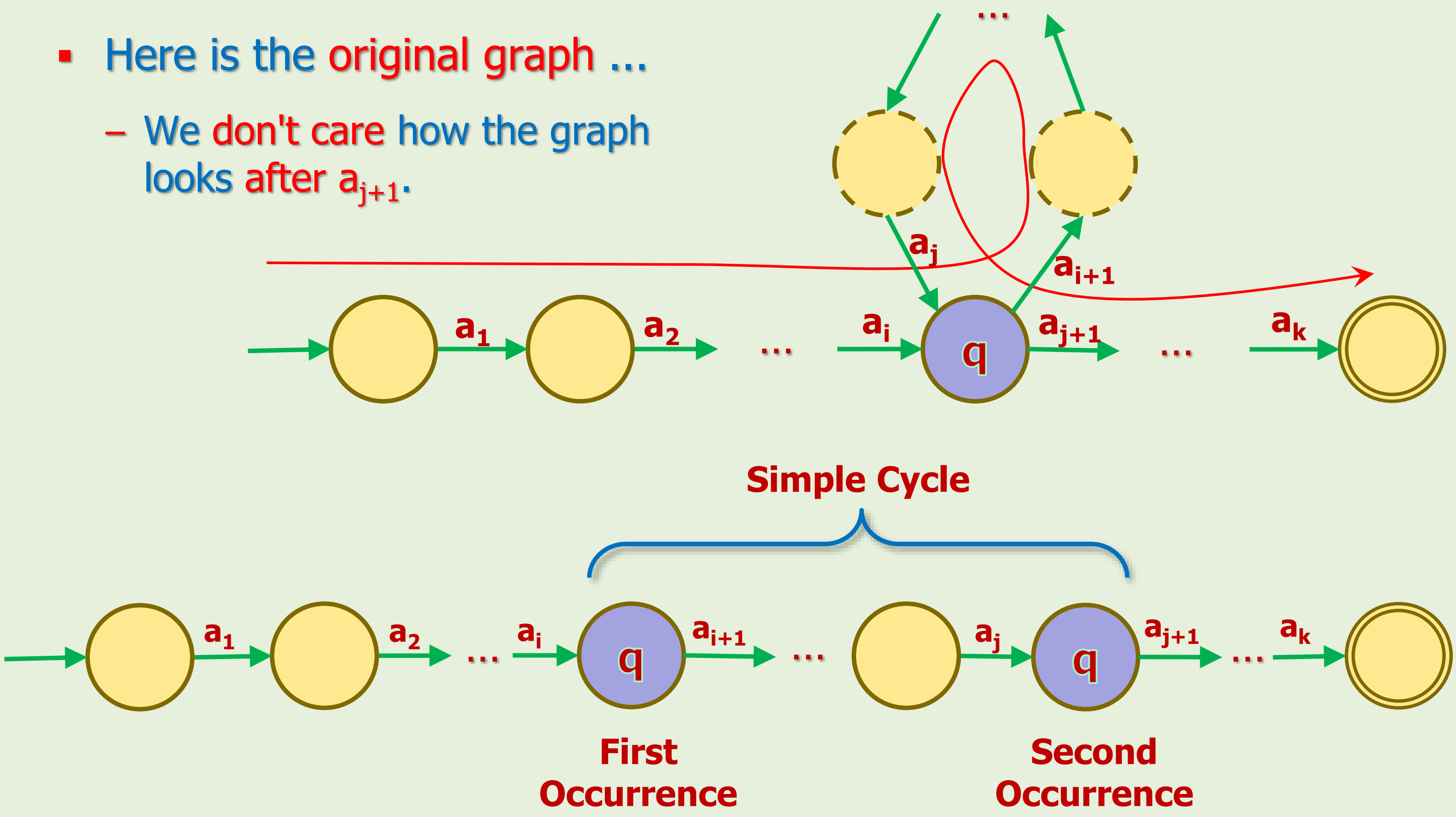
- Note that between two q's, there is no nested repeated states.
  - We can always pick the first repeated state in which there is no nested repeated states.
- Therefore, if we show the original graph, this portion must be a "simple cycle".



- So, let's see the original graph ...

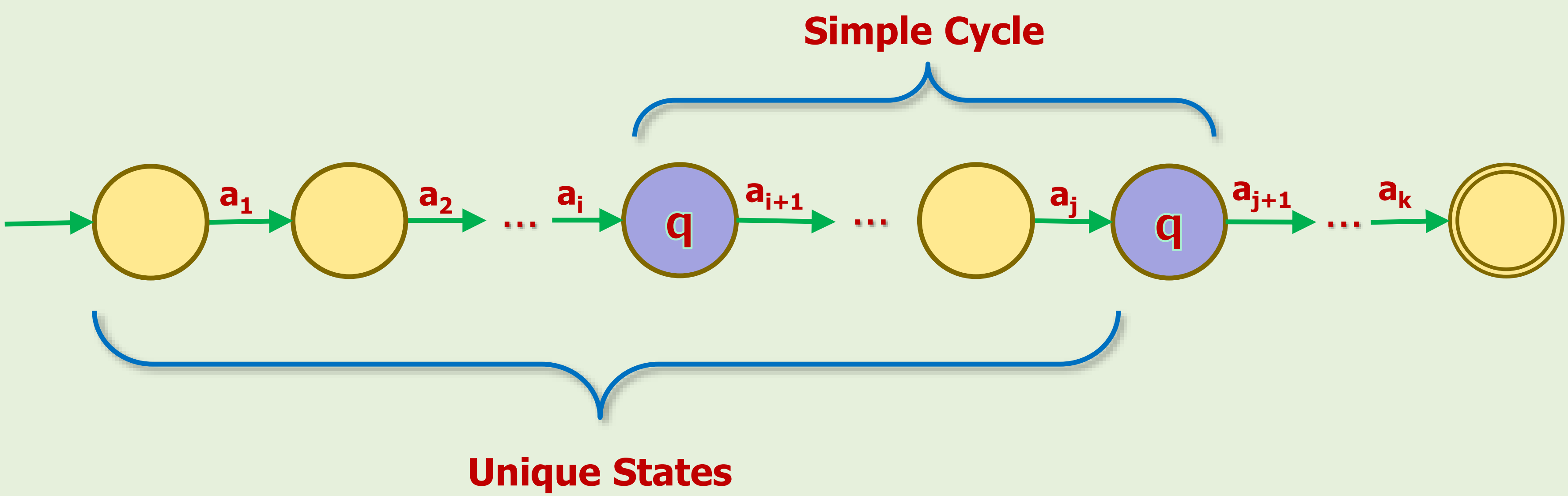
Visualizing Pumping Lemma

- Here is the original graph ...
  - We don't care how the graph looks after a_{j+1}.



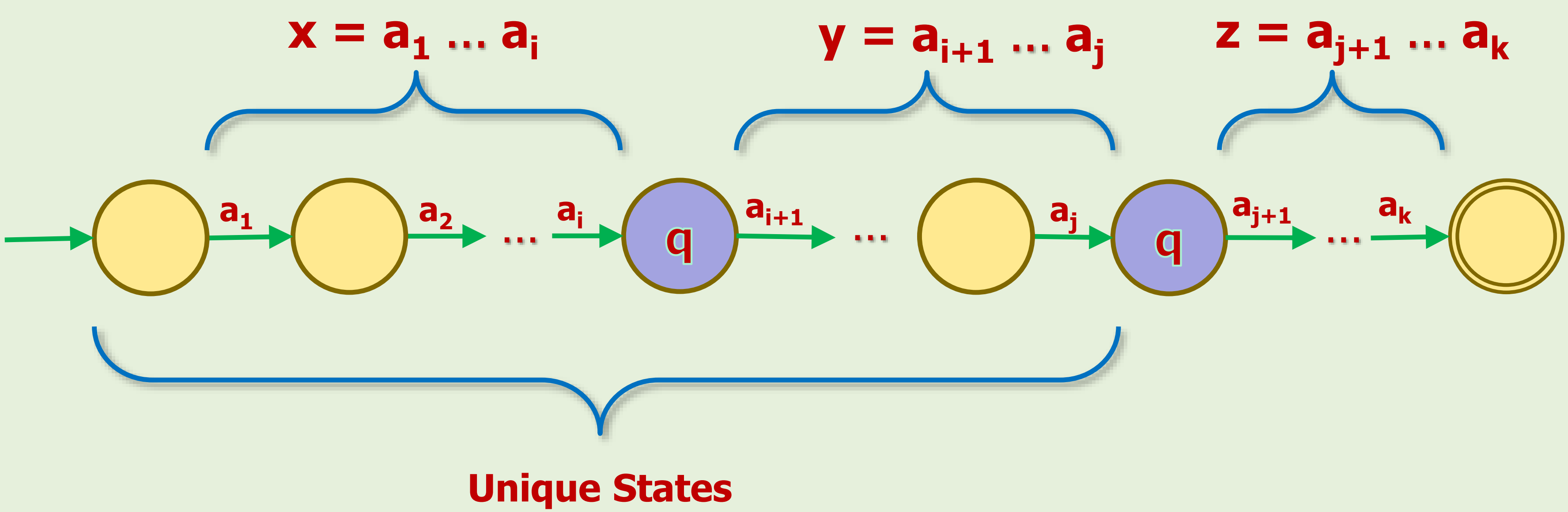
Visualizing Pumping Lemma

- Let's review the facts we have so far:
  1. From a_1 to a_i, we have unique states (visited once). because we assumed q is the first repeating state.
  2. From a_{i+1} to a_j, we have unique states because it is a simple cycle.
- Therefore, we have unique states from a_1 to a_j.



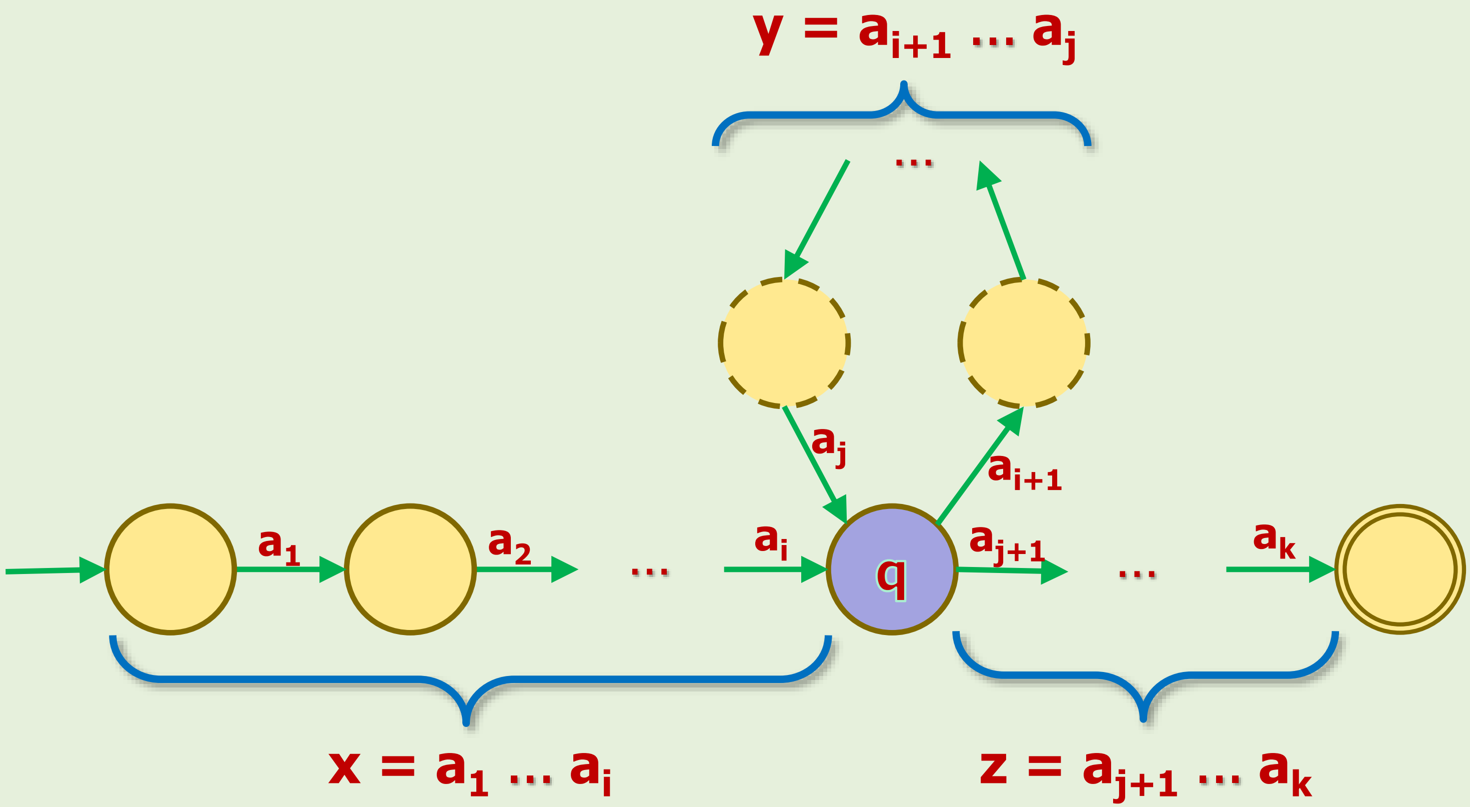
Visualizing Pumping Lemma

- Now, let's name different portions of the string:
- We split w as $w = xyz$.
- Note that y corresponds to substring between two q's.



Visualizing Pumping Lemma

- Now let's see how x, y, and z looks in the original transition graph.



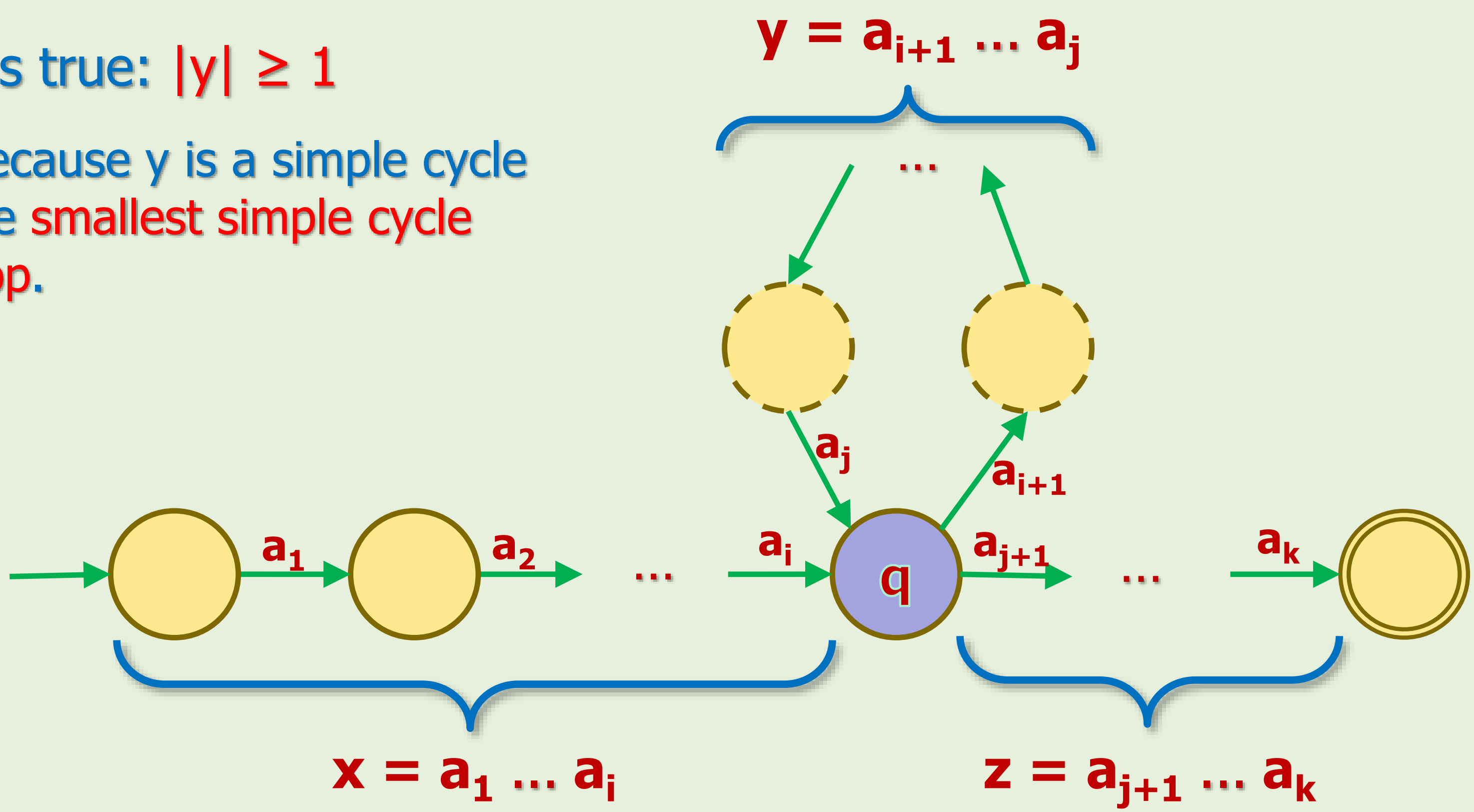
Important Questions

1. Is this true: $|xy| \leq m$

   Yes, because we learned a_1 to a_j (= xy) are unique states and there is no repeated states between them.

2. Is this true: $|y| \geq 1$

   Yes, because y is a simple cycle and the smallest simple cycle is a loop.



More Questions!

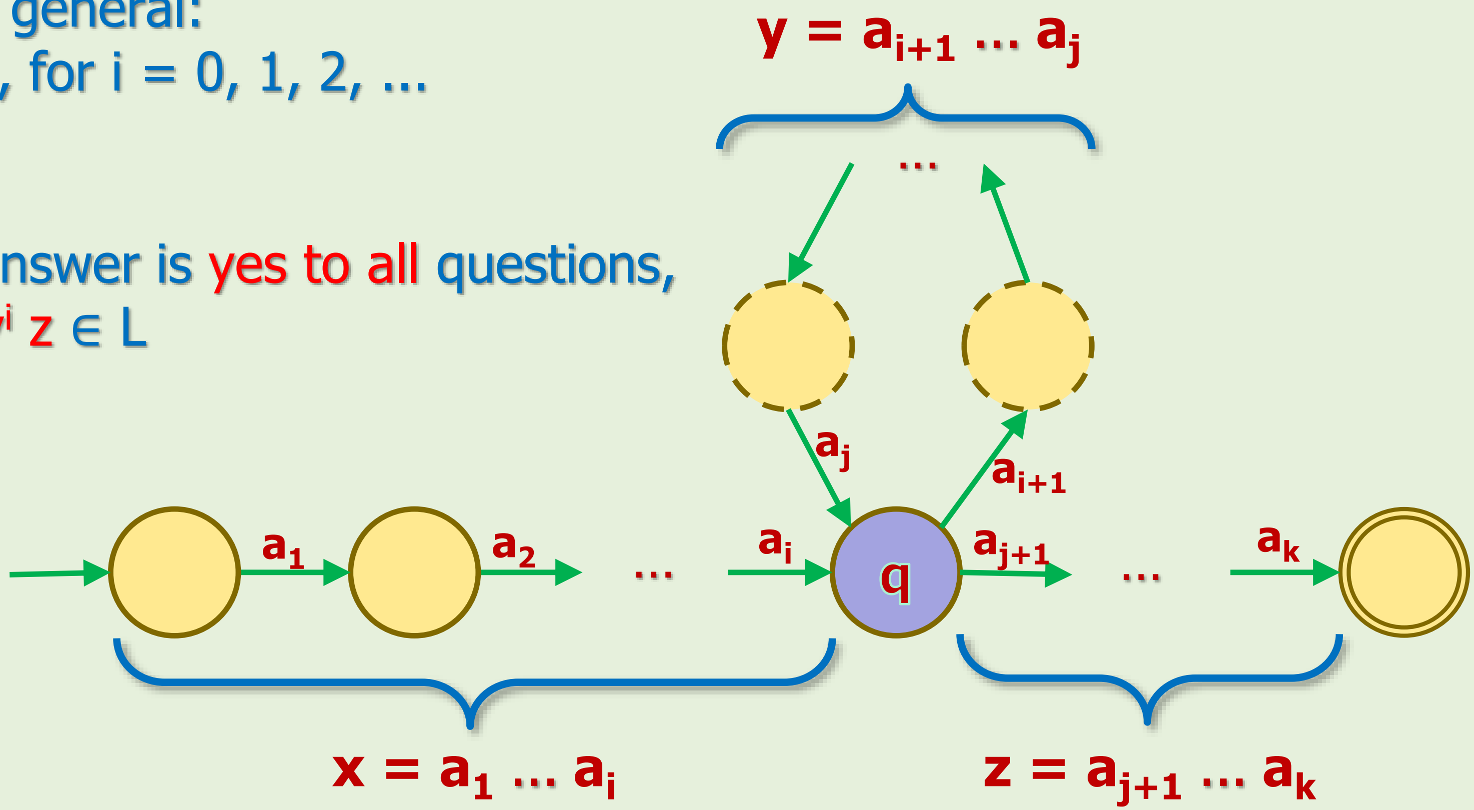
3. Is string $xz = a_1\ a_2 \ldots a_i\ a_{j+1} \ldots a_k$ accepted by this DFA?

   Yes, so $xz \in L$

4. How about $xyyz$? Or, $xyyyz$?

5. Or in general:
   $x\ y^i\ z$, for $i = 0, 1, 2, \ldots$

- The answer is yes to all questions, so $x\ y^i\ z \in L$



Visualizing Pumping Lemma

Conclusion

- We could pump any number of y and the resulting strings were accepted by the DFA.
- So, if $w = xyz \in L$, then $w_i = xy^iz \in L$ for $i = 0, 1, 2, \ldots$
- And this was the **mysterious concept** of "Pumping Lemma".

Notes About Pumping Lemma

1. Pumping lemma is difficult to understand! [Text book, P#121]

   NOT anymore! 

2. Pumping lemma is not applicable to finite languages.

   Because we need to pump infinite y's!

3. Pumping lemma cannot prove that a languages is regular.

   Because you'd need to verify infinite cases!

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. Costas Busch's website: http://csc.lsu.edu/~busch/
4. Michael Sipser, "Introduction to the Theory of Computation, 3rd ed.," CENGAGE Learning, United States, 2013
ISBN-13: 978-1133187790