BLM2502 Theory of Computation

BLM2502 Theory of Computation

Course Outline

Week	Content
1	Introduction to Course
2	Computability Theory, Complexity Theory, Automata Theory, Set Theory, Relations, Proofs, Pigeonhole Principle
3	Regular Expressions
4	Finite Automata
5	**Deterministic and Nondeterministic Finite Automata**
6	**Epsilon Transition, Equivalence of Automata**
7	Pumping Theorem
8	April 10 - 14 week is the first midterm week
9	Context Free Grammars
10	Parse Tree, Ambiguity,
11	Pumping Theorem
12	Turing Machines, Recognition and Computation, Church-Turing Hypothesis
13	Turing Machines, Recognition and Computation, Church-Turing Hypothesis
14	May 22 – 27 week is the second midterm week
15	Review
16	Final Exam date will be announced

Week I - Introduction

Non-regular languages

(Pumping Lemma)

Non-regular languages

$$\{a^n b^n : \quad n \geq 0\}$$

$$\{vv^R : \quad v \in \{a,b\}*\}$$

Regular languages

$$a*b$$

$$b*c+a$$

$$b+c(a+b)*$$

$$etc...$$

How can we prove that a language L is not regular?

Prove that there is no DFA or NFA or RE that accepts L

Difficulty: this is not easy to prove

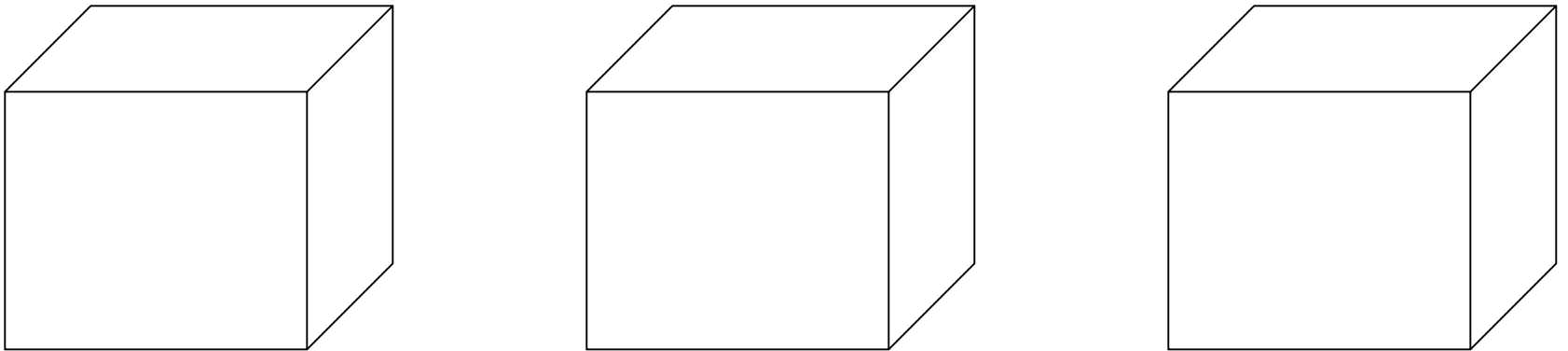(since there is an infinite number of them)

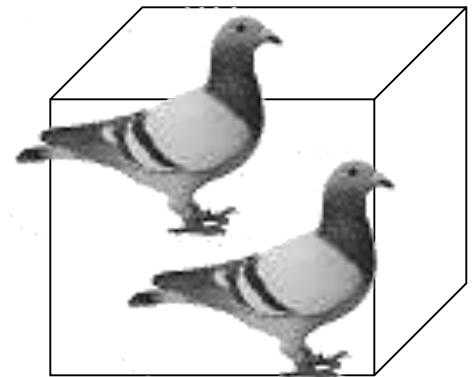**Solution**: use the Pumping Lemma !!!
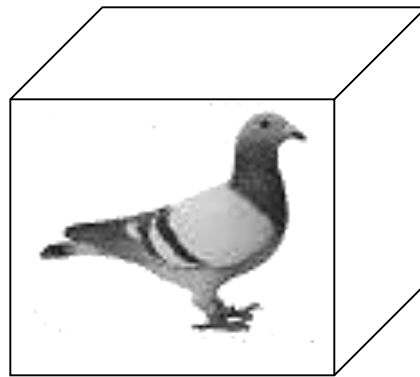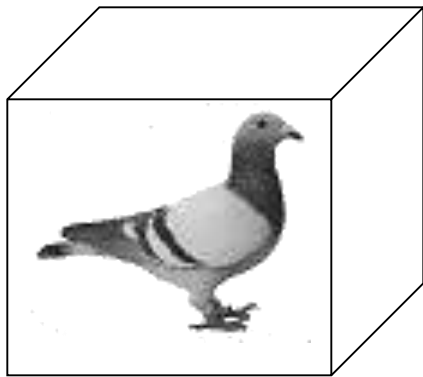
The Pigeonhole Principle

4 pigeons

3 pigeonholes

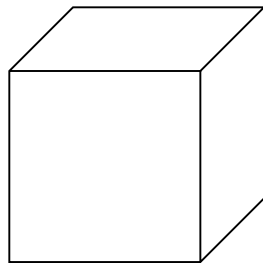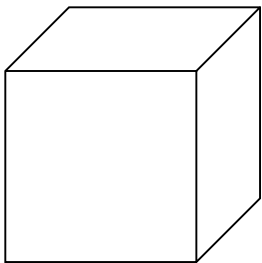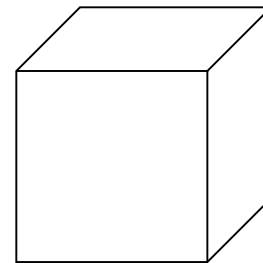# A pigeonhole must contain at least two pigeons
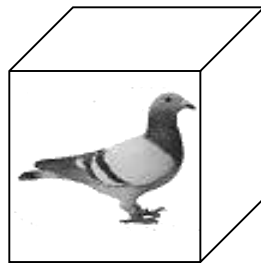
n pigeons

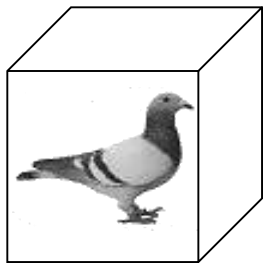# m pigeonholes $n > m$

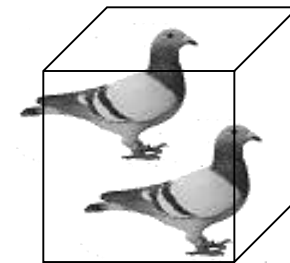# The Pigeonhole Principle

n pigeons

m pigeonholes

$n > m$

There is a pigeonhole
with at least 2 pigeons

....

The Pigeonhole Principle

and

DFAs

Consider a DFA with 4 states
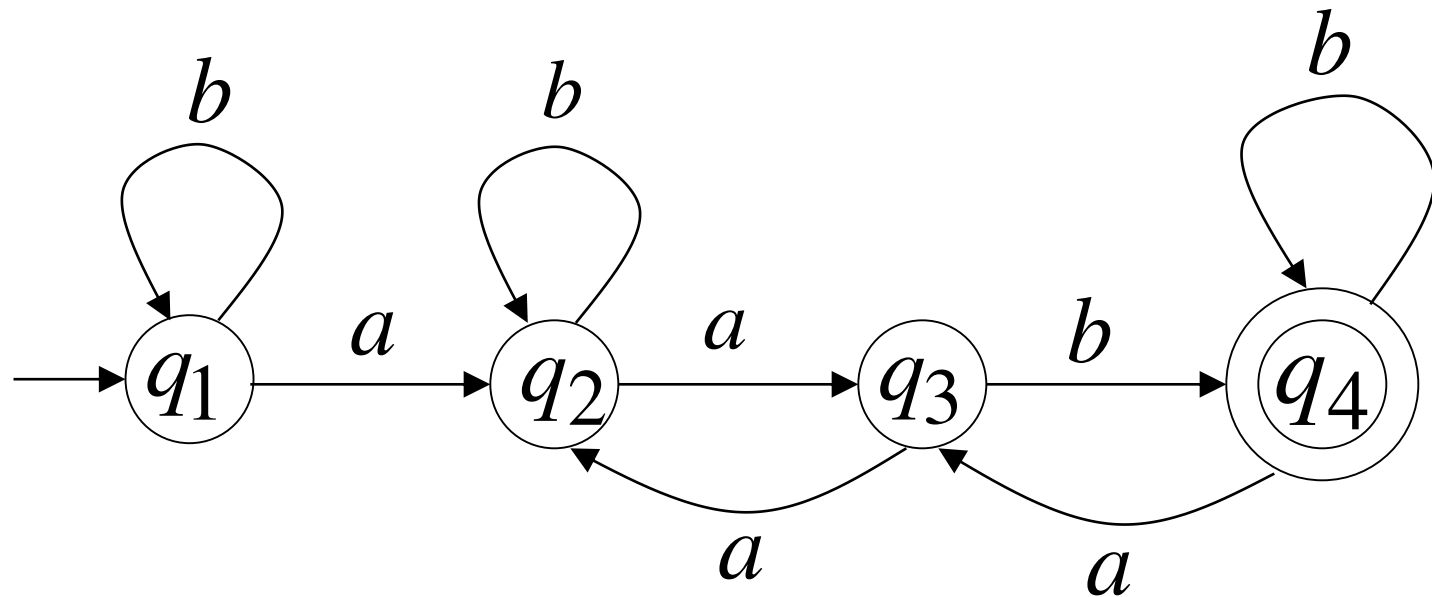
Consider the walk of a "long" string: $aaaab$ (length at least 4)

A state is repeated in the walk of $aaaab$
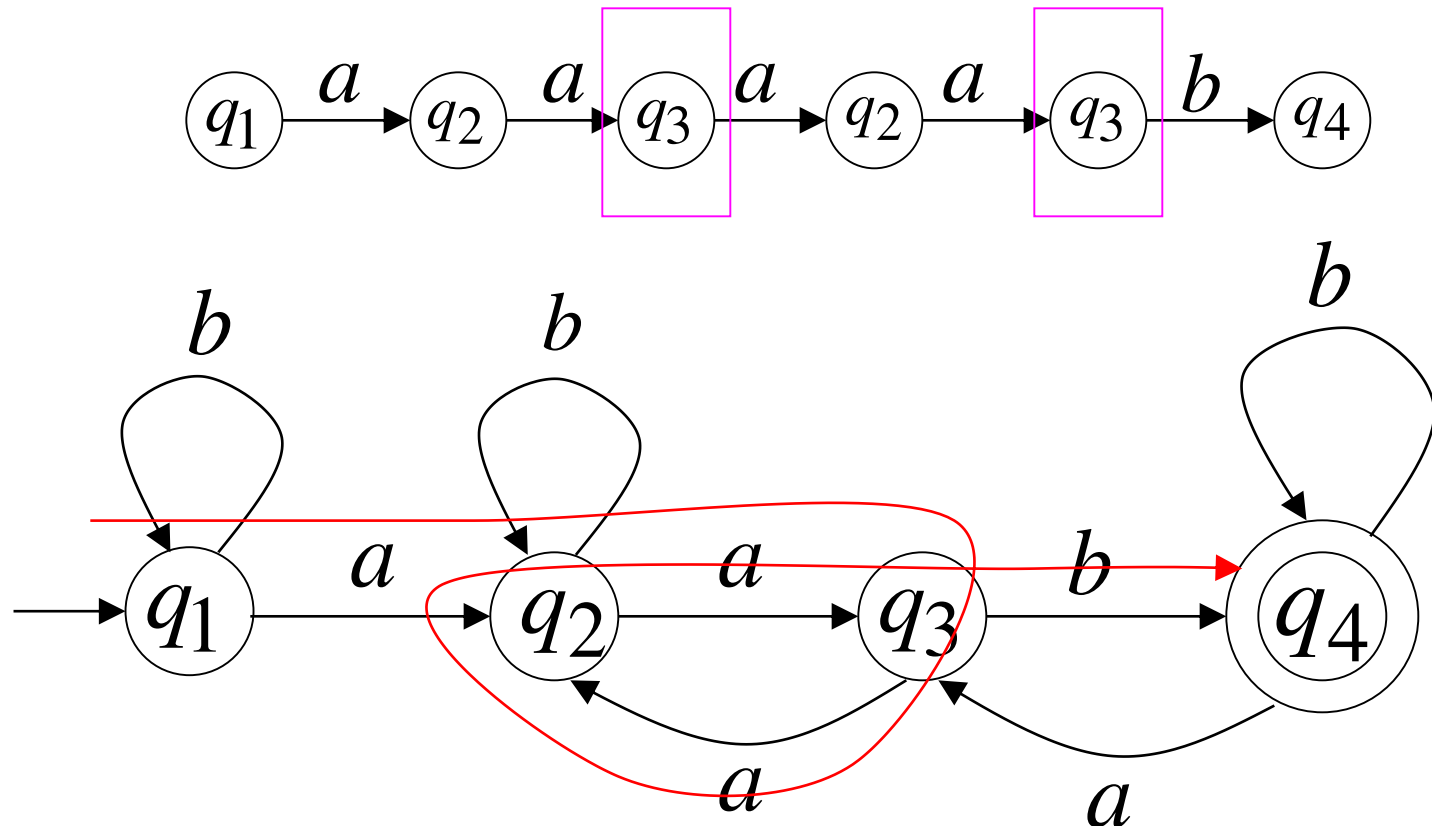
The state is repeated as a result of the pigeonhole principle

N state

en n≥ z dovo geç'lmd. N+1 inpvt z dove 1 state dan

Walk of $adaab$

Pigeons:
(walk states)

Are more than

$$q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \xrightarrow{b} q_4$$

Nests:
(Automaton states)

q_1 q_2 q_3 q_4

Repeated state

Consider the walk of a "long" string: $aabb$
(length at least 4)

Due to the pigeonhole principle:

A state is repeated in the walk of $aabb$

The state is repeated as a result of the pigeonhole principle

Walk of $aabb$

Pigeons:
(walk states)

$$q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \xrightarrow{b} q_4 \xrightarrow{b} q_4$$

Are more than

Nests:
(Automaton states)

q_1 q_2 q_3 q_4
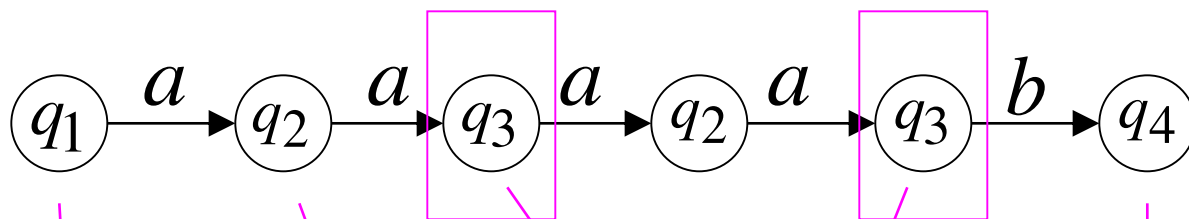
Automaton States
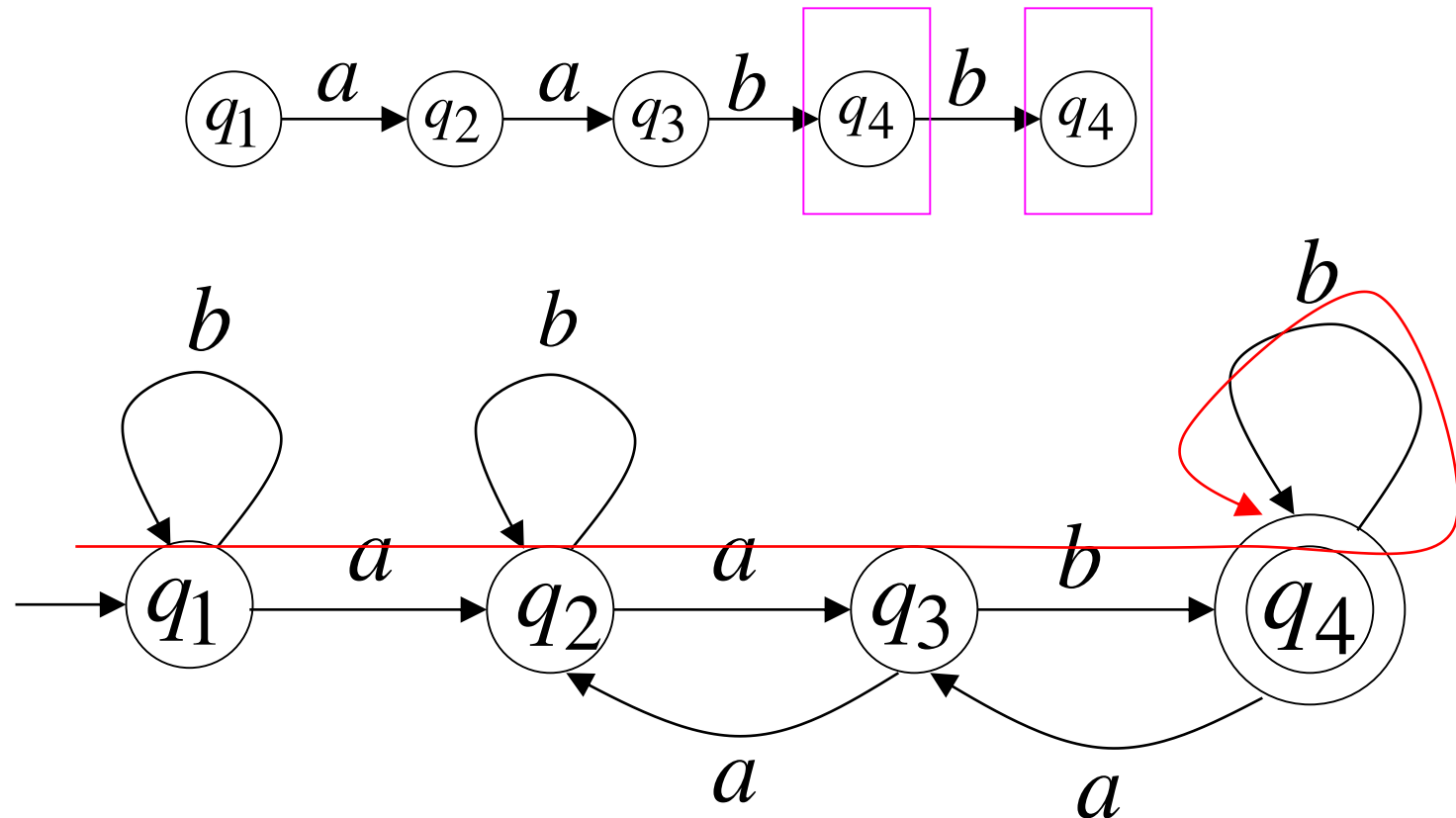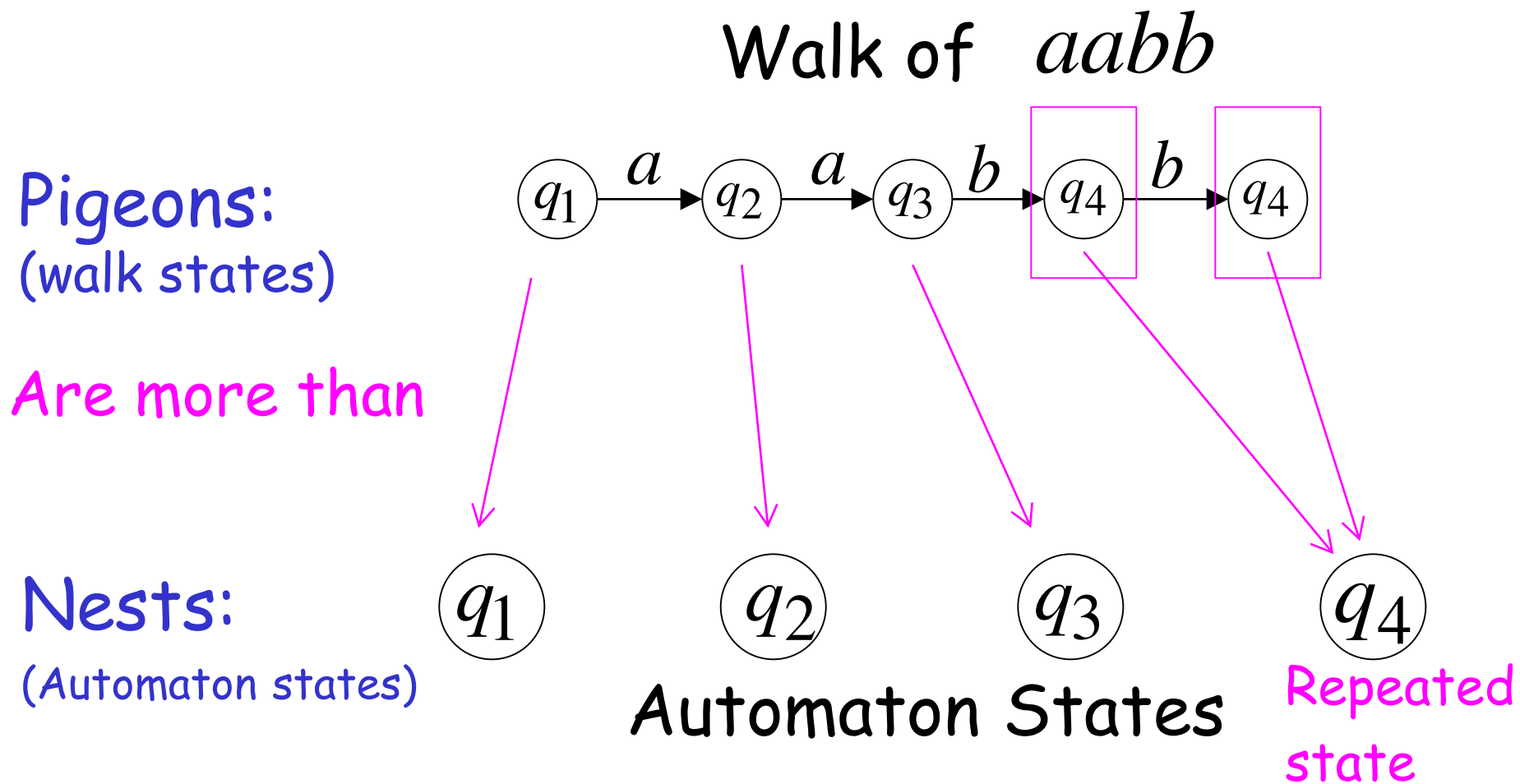
Repeated state

In General: If $|w| \geq \#$states of DFA,
by the pigeonhole principle,
a state is repeated in the walk w

Walk of $w = \sigma_1 \sigma_2 \cdots \sigma_k$

Arbitrary DFA

X Repeated state

$$|w| \geq \#\text{states of DFA} = m$$

Walk of w

Pigeons: (walk states)

Are more than

Nests: (Automaton states)
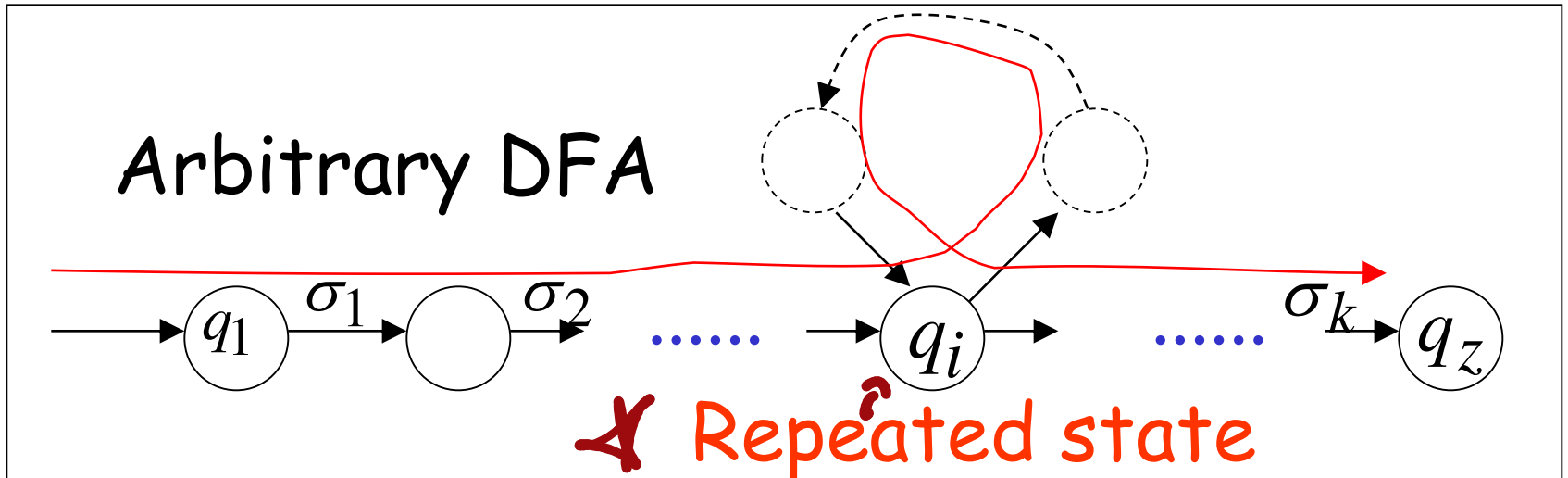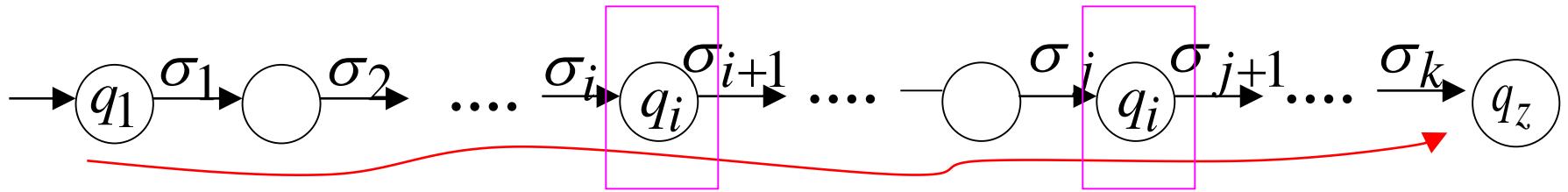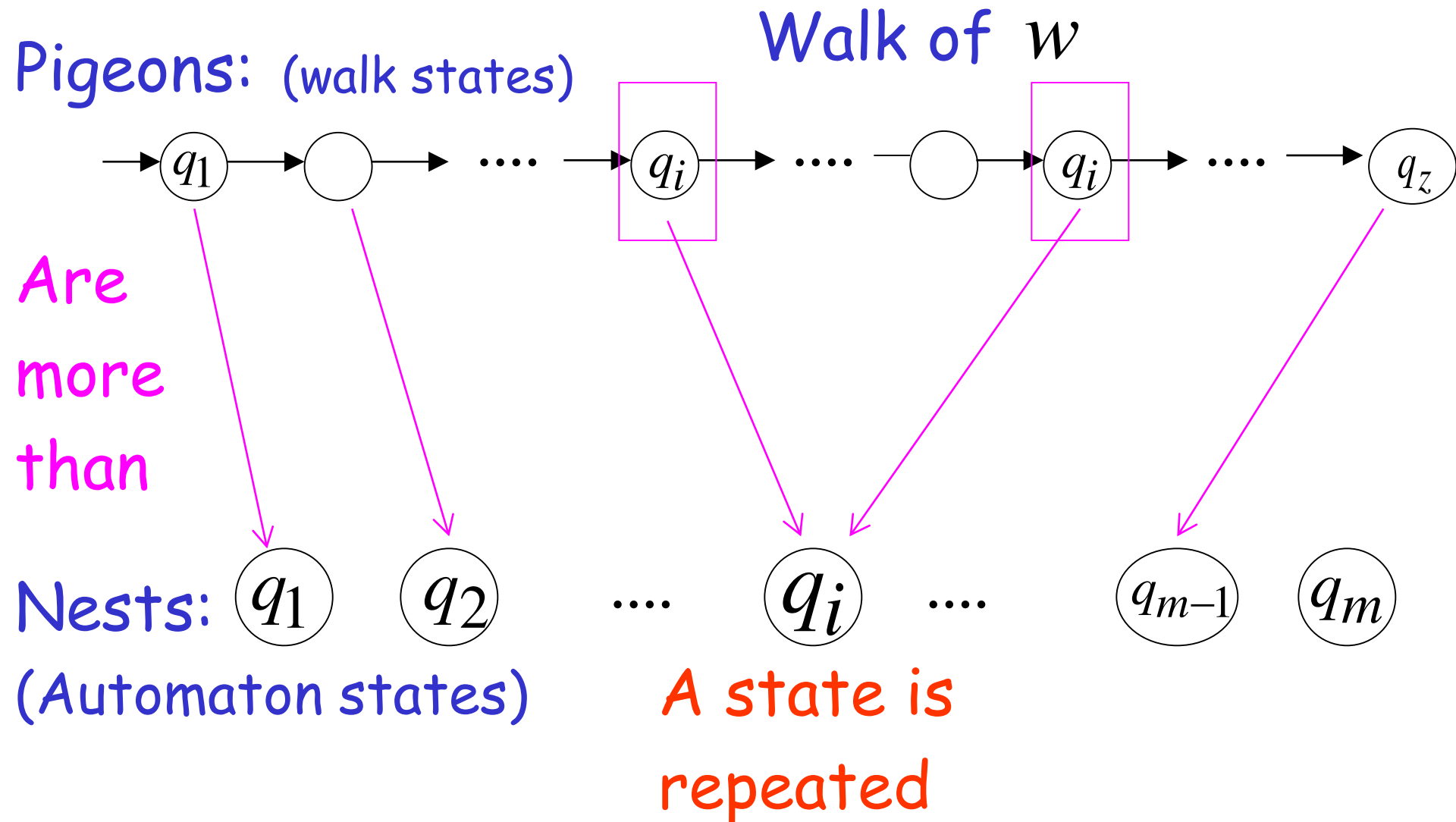
A state is repeated

The Pumping Lemma

Take an infinite regular language L
(contains an infinite number of strings)

There exists a DFA that accepts L
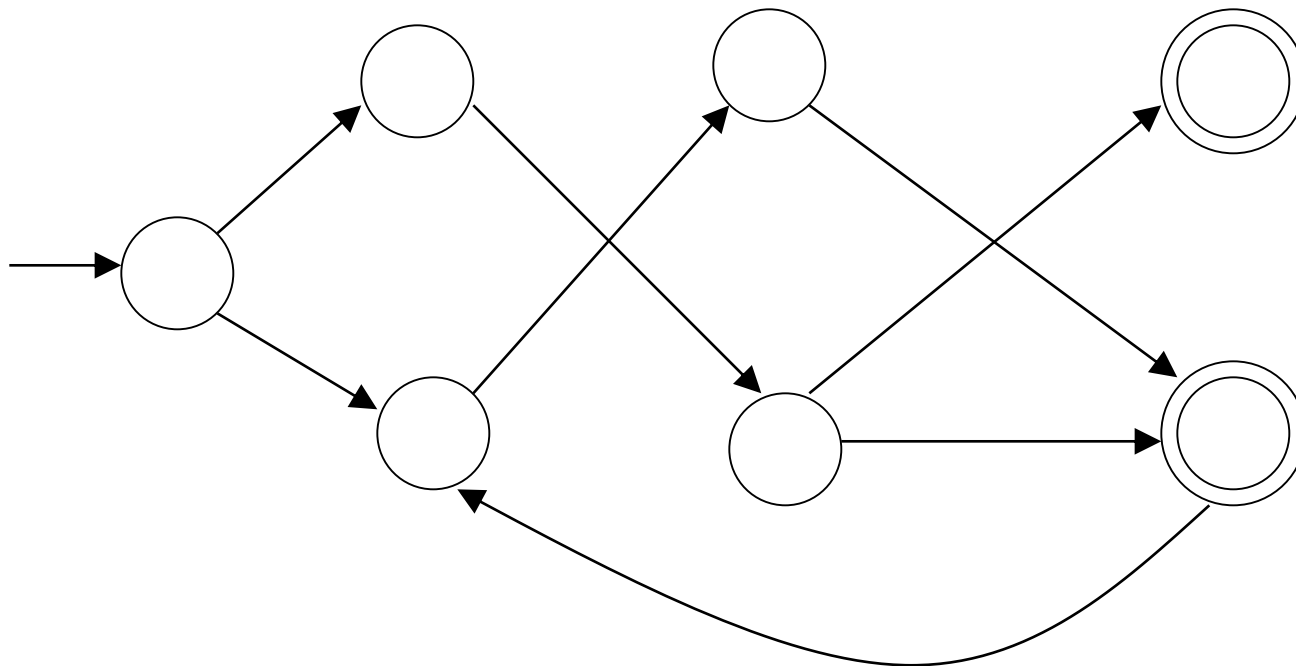
m
states

Take string $w \in L$ with $|w| \geq m$

(number of states of DFA)

then, at least one state is repeated in the walk of w

Walk in DFA of

$$w = \sigma_1 \sigma_2 \cdots \sigma_k$$

Repeated state in DFA
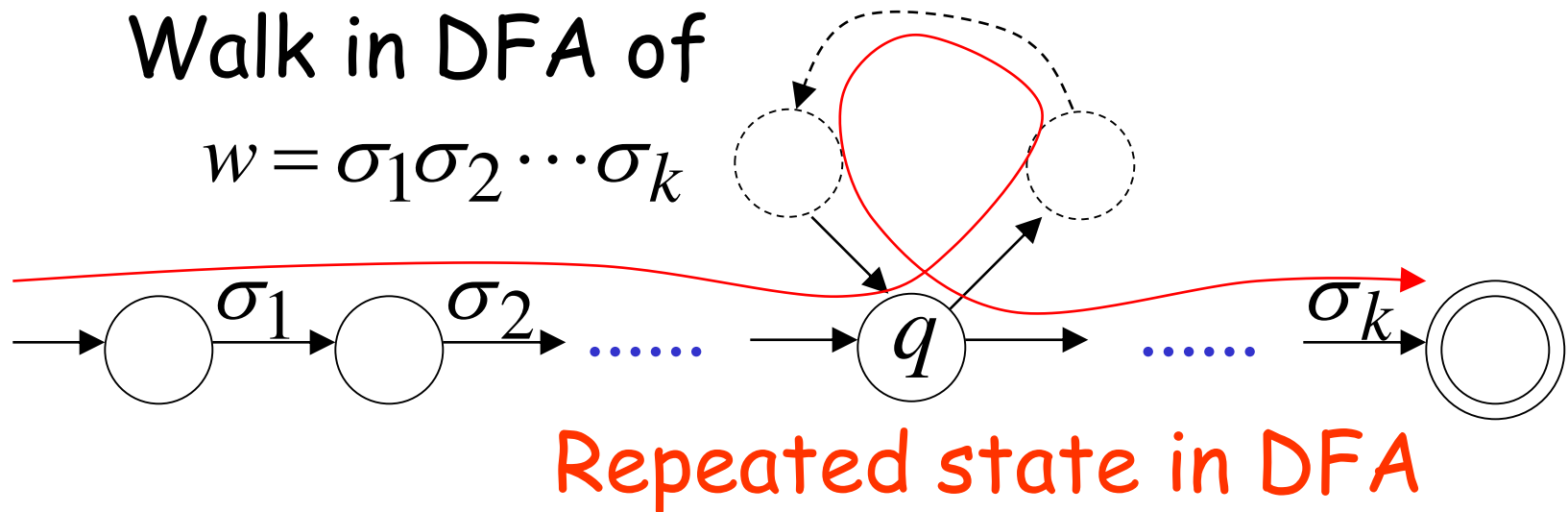
There could be many states repeated

Take q to be the first state repeated

One dimensional projection of walk w:

First occurrence

Second occurrence

Unique states
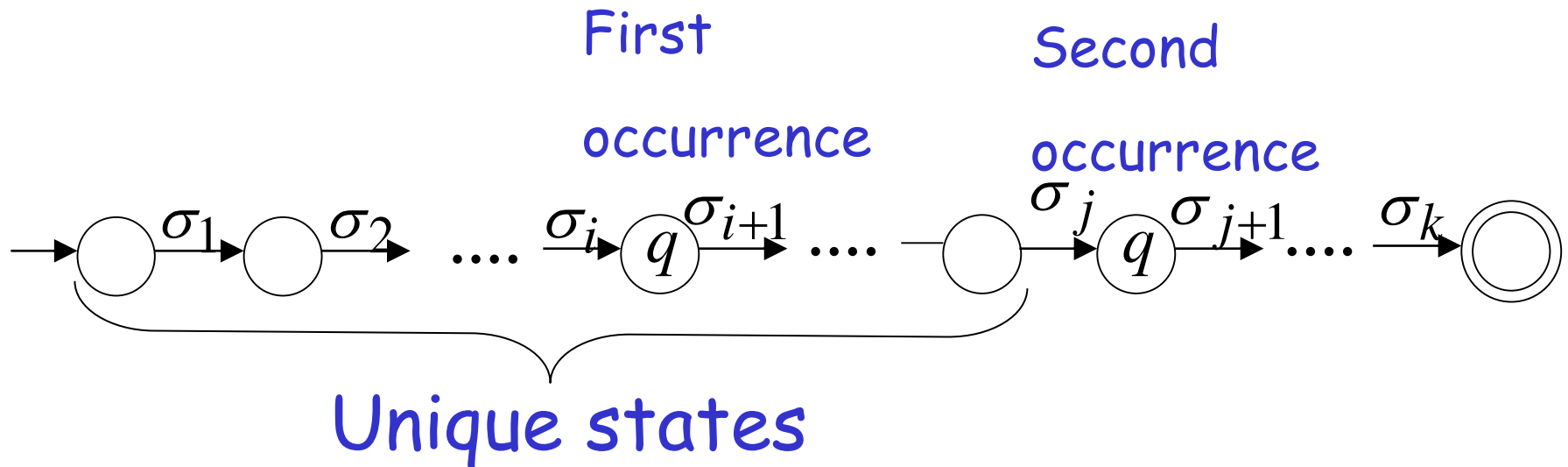
We can write $w = xyz$

xin de y*nin* 0
de *uzunlugu*
olabilir

q'den *q* *ya*
s ile *gesilirse*
... *+ ya*

One dimensional projection of walk w:

First
occurrence

Second
occurrence

$$x = \sigma_1 \cdots \sigma_i \qquad y = \sigma_{i+1} \cdots \sigma_j \qquad z = \sigma_{j+1} \cdots \sigma_k$$
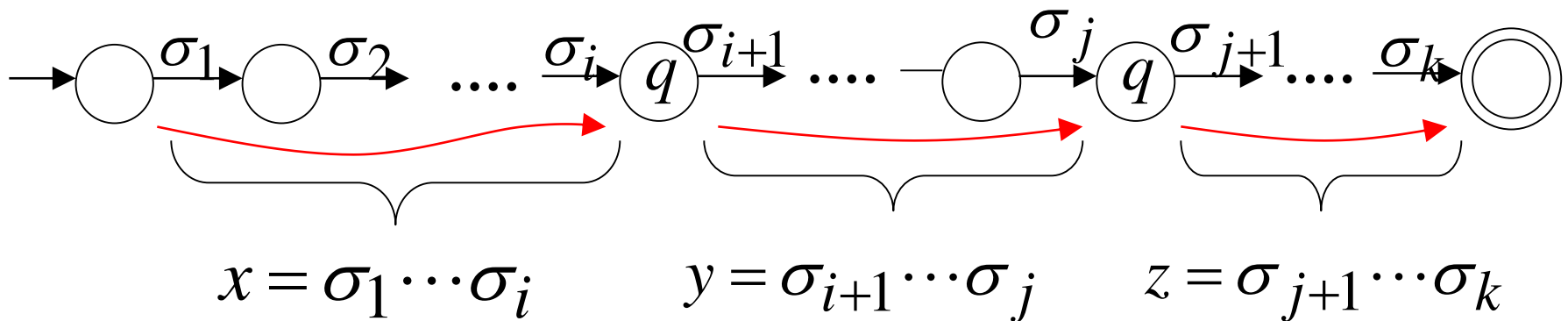
In DFA: $w = x\ y\ z$

$x\ y < m$

contains only
first occurrence of q

y

x

$z.$
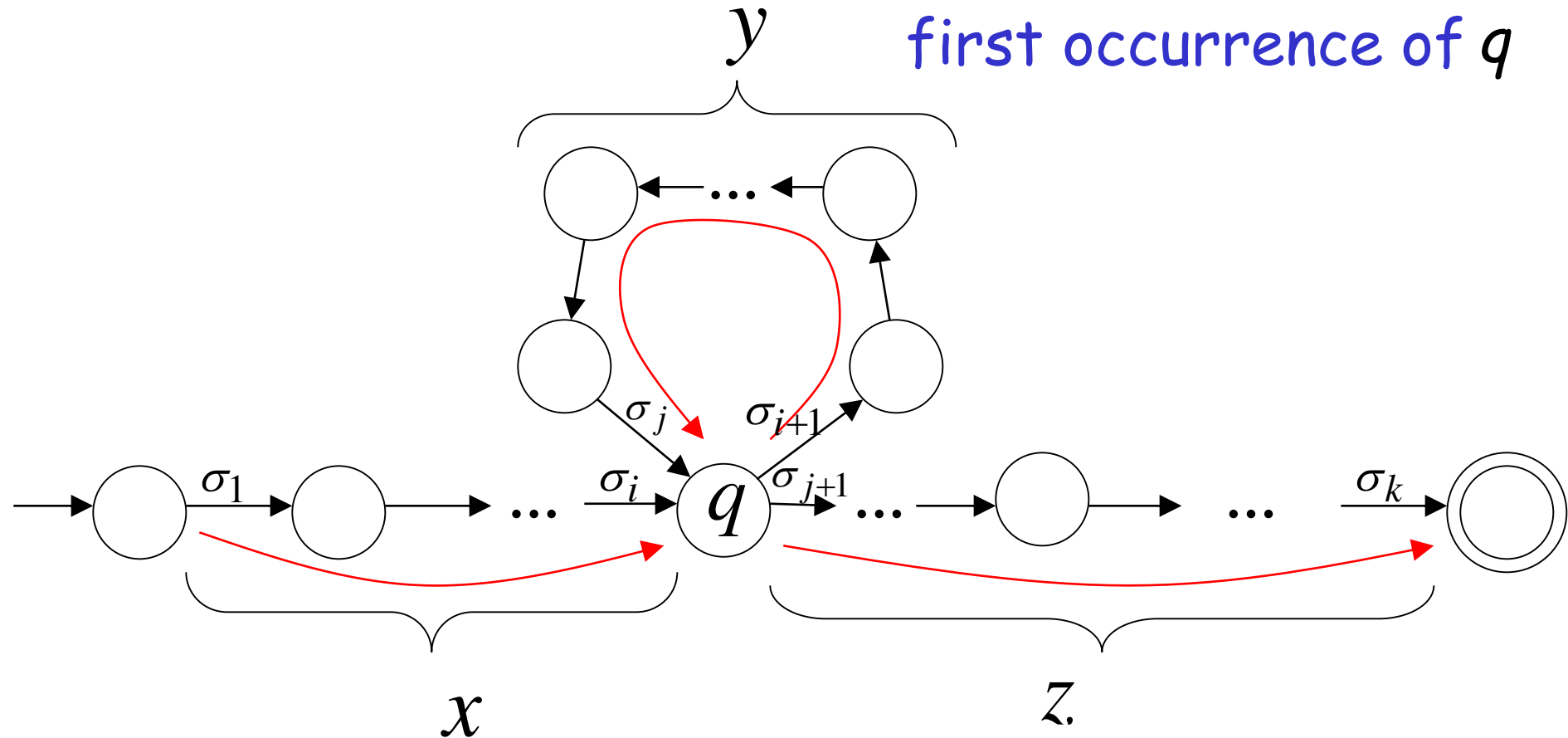
Observation: length $|x\,y| \leq m$ number of states of DFA
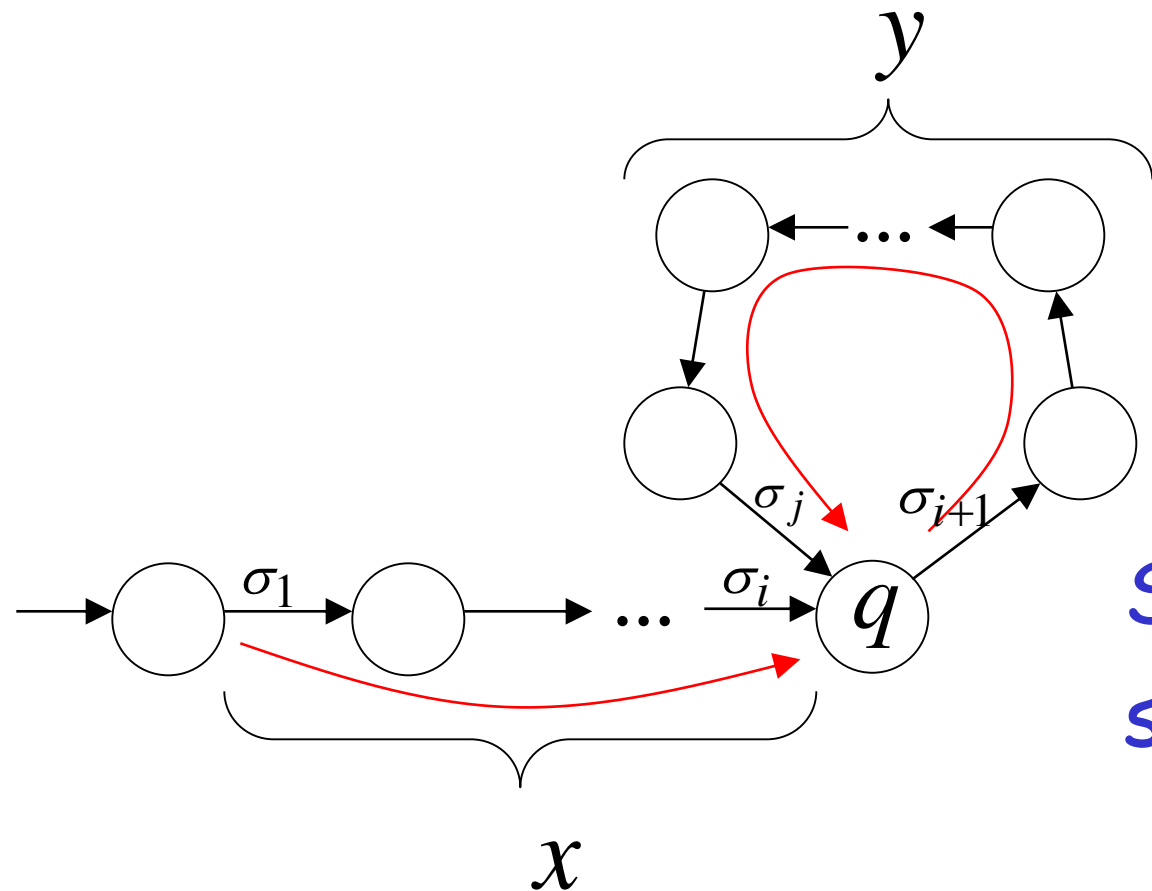
Unique States

Since, in xy no state is repeated

(except q)

Observation: length $|y| \geq 1$

Since there is at least one transition in loop
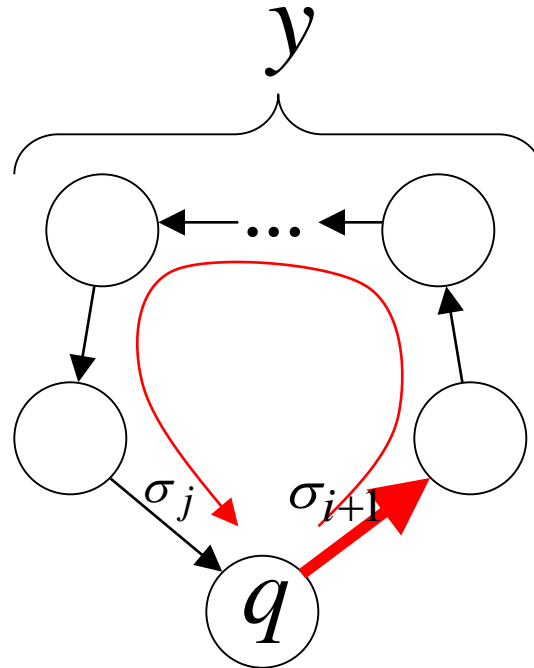
We do not care about the form of string z.

z may actually overlap with the paths of x and y
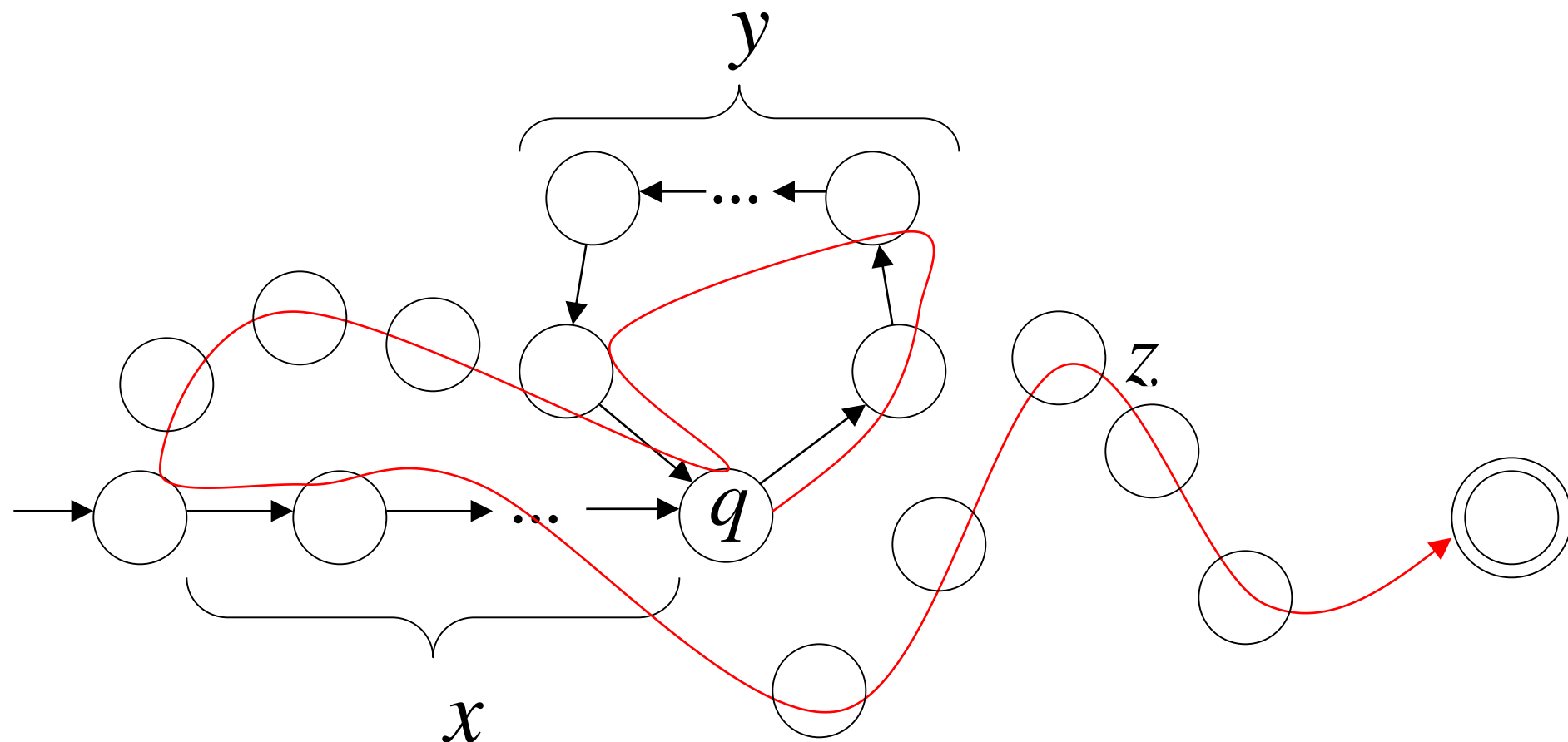
Additional string: The string $x\ z$ is accepted

Do not follow loop

y

σ_j σ_{i+1}

σ_1 σ_i q σ_{j+1} σ_k

x

$z.$
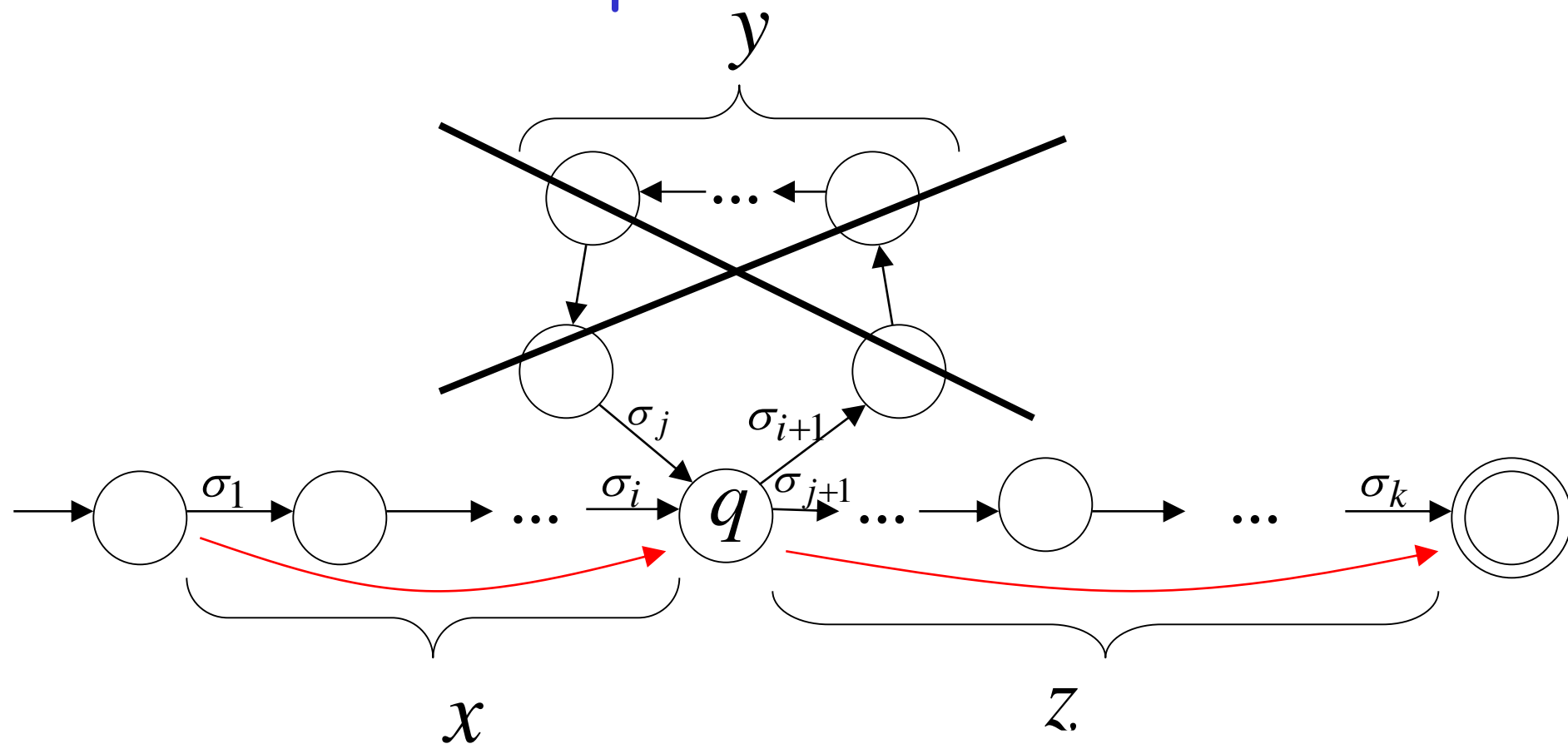
Additional string: The string $x\ y\ y\ z$ is accepted

Follow loop 2 times
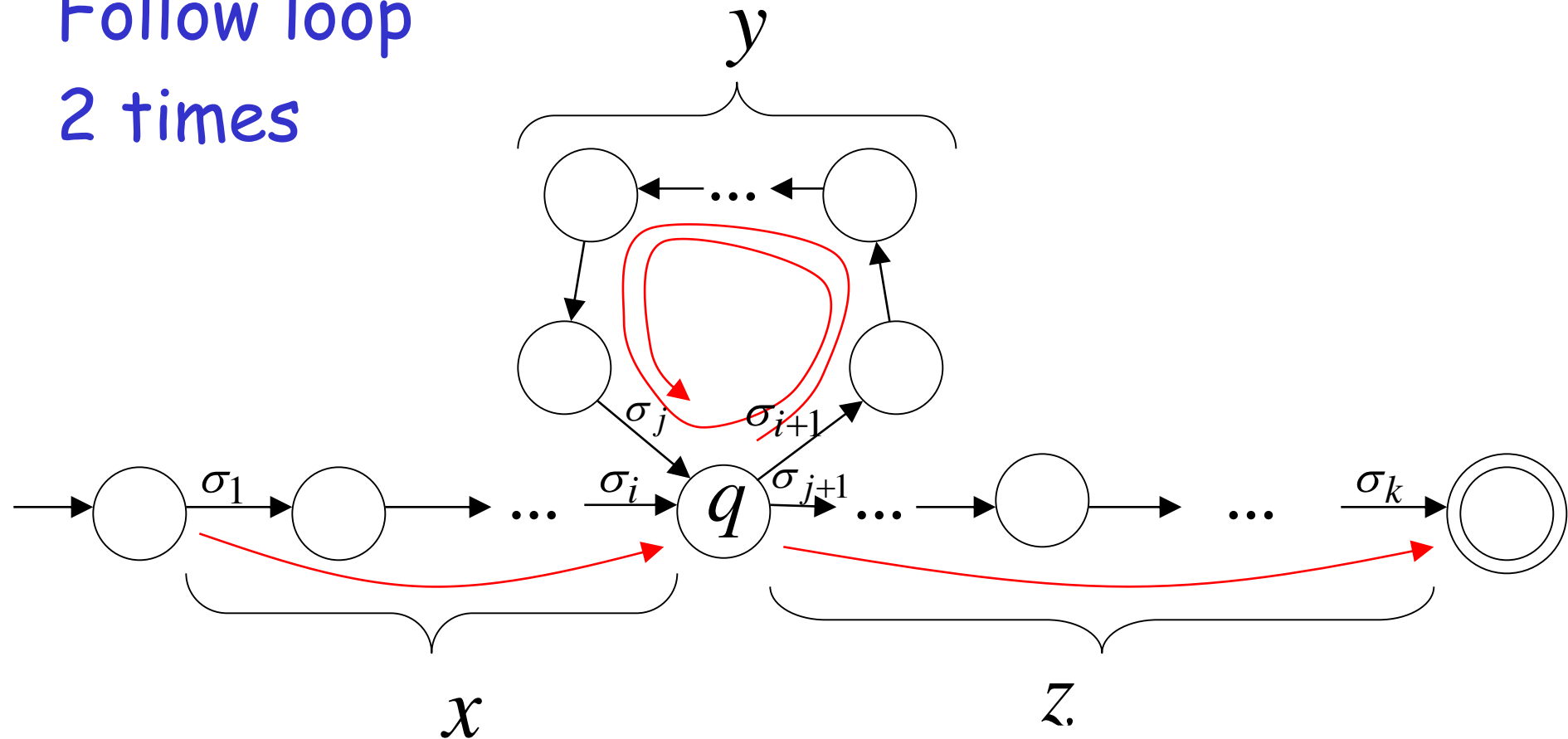
Additional string: The string $x\ y\ y\ y\ z$ is accepted

Follow loop 3 times
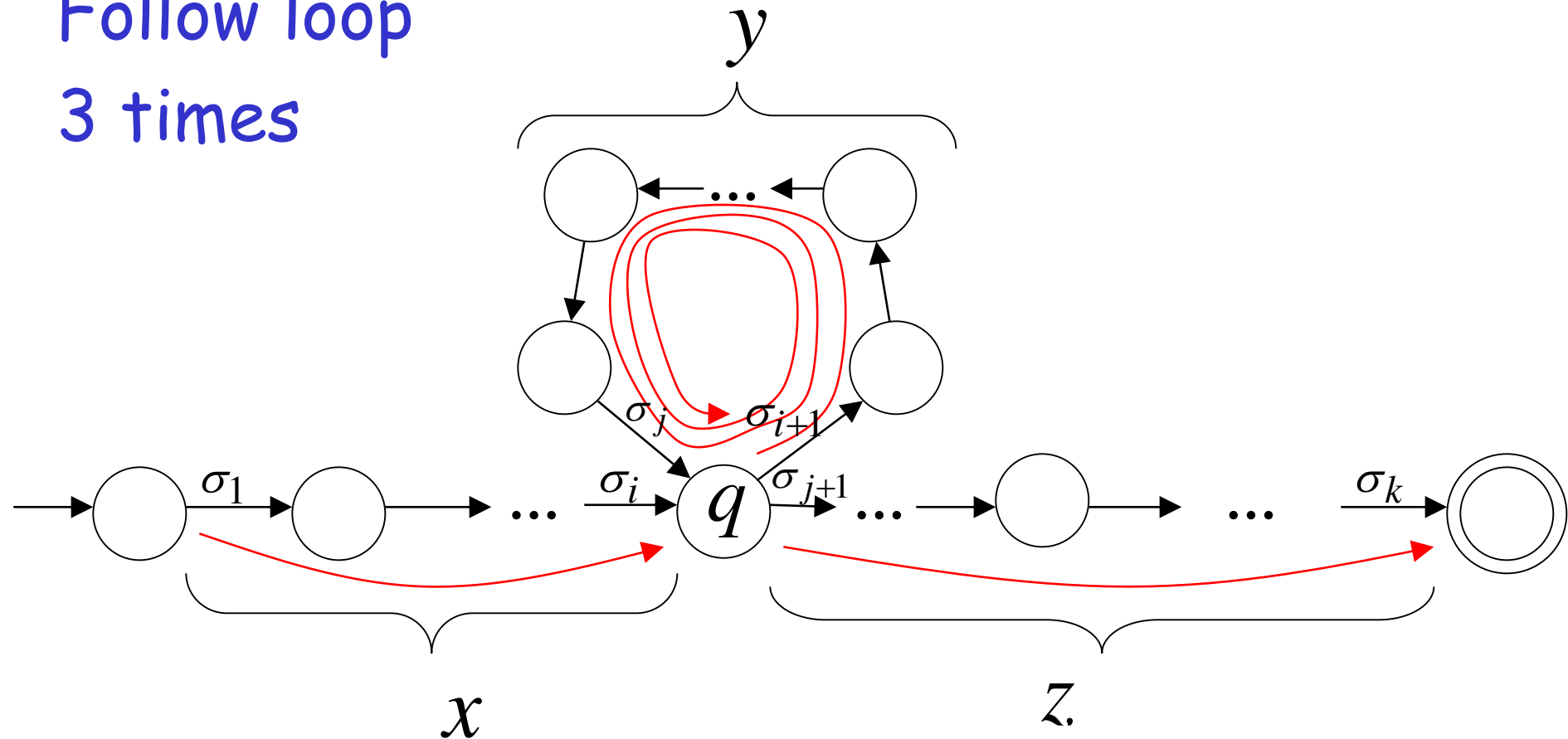
In General: The string $x\, y^i\, z$ is accepted $i = 0, 1, 2, \ldots$

Follow loop i times
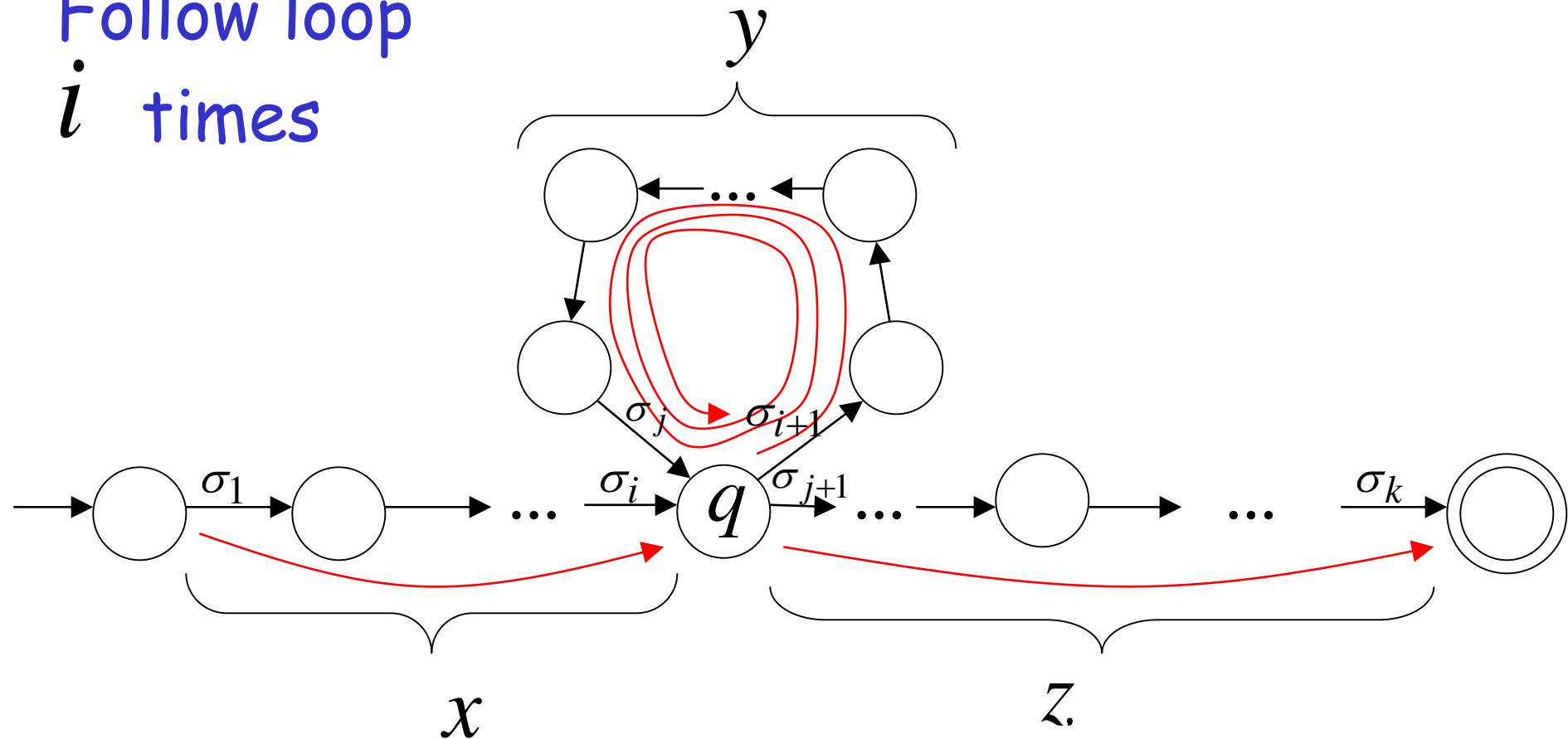
Therefore: $x \, y^i \, z \, \in L$ $i = 0, 1, 2, \ldots$

Language accepted by the DFA
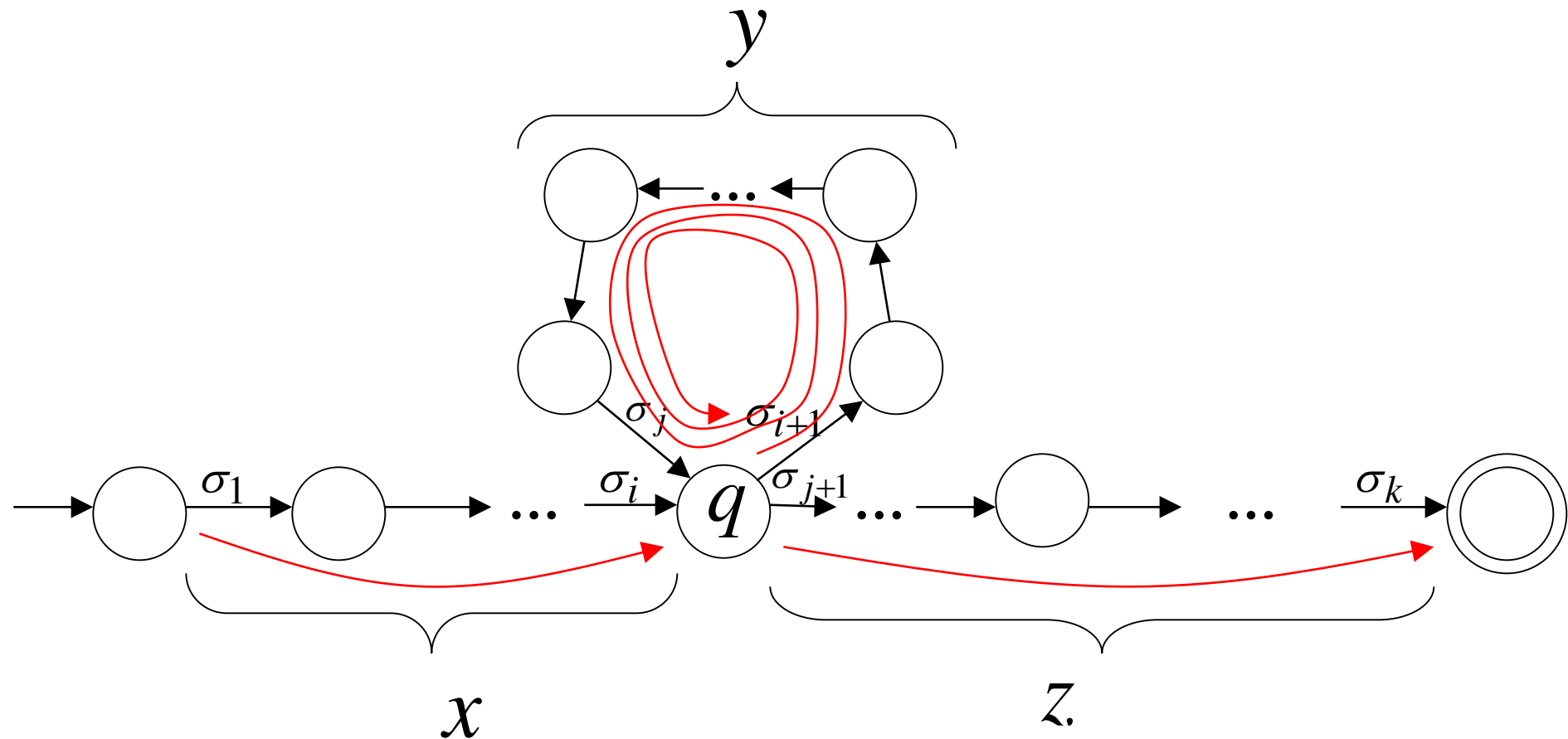
In other words, we described:

The Pumping Lemma !!!

The Pumping Lemma:

- Given a infinite regular language L

- there exists an integer m (critical length)

- for any string $w \in L$ with length $|w| \geq m$

- we can write $w = x\,y\,z$

- with $|x\,y| \leq m$ and $|y| \geq 1$

- such that: $x\,y^i\,z \in L$ $\qquad i = 0, 1, 2, \dots$

In the book:

Critical length m = Pumping length p

Applications

of

the Pumping Lemma

Observation:

→ n tane
state
sayısı

Every language of finite size has to be regular

(we can easily construct an NFA
that accepts every string in the language)

Therefore, every non-regular language

has to be of infinite size

(contains an infinite number of strings)

Suppose you want to prove that
An infinite language L is not regular

1. Assume the opposite: L is regular

2. The pumping lemma should hold for L

3. Use the pumping lemma to obtain a contradiction
   ↳ çelişki

4. Therefore, L is not regular

Explanation of Step 3: How to get a contradiction

1. Let m be the critical length for L

2. Choose a particular string $w \in L$ which satisfies the length condition $|w| \geq m$

3. Write $w = xyz$

   not y besinlikle
   döngü olmalı.

4. Show that $w' = xy^i z \notin L$ for some $i \neq 1$

5. This gives a contradiction, since from pumping lemma $w' = xy^i z \in L$

Note: It suffices to show that
only one string $w \in L$
gives a contradiction

You don't need to obtain
contradiction for every $w \in L$

Example of Pumping Lemma application

Theorem: The language $L = \{a^n b^n : n \geq 0\}$

is not regular

$a^m b^m$

$xyz = aabb$

$y = a$

$xy^k z$

$xy^4 z$

$a^{m+k} b^m \neq a^m b^m$

Proof: Use the Pumping Lemma

$$L = \{a^n b^n : n \geq 0\}$$

Assume for contradiction
that L is a regular language

Since L is infinite
we can apply the Pumping Lemma

$$L = \{a^n b^n : n \geq 0\}$$

Let m be the critical length for L

Pick a string w such that: $w \in L$

and length $|w| \geq m$

We pick $w = a^m b^m$

From the Pumping Lemma:

we can write $w = a^m b^m = x\ y\ z$

with lengths $|x\ y| \leq 2m, \quad |y| \geq 1$

$$w = xyz = a^m b^m = \overbrace{a...a}^{\quad\quad m \quad\quad} \underbrace{a...ab...b}_{} \overbrace{}^{m}$$

$$\underbrace{a...a}_{x}\underbrace{a...a}_{y}\underbrace{a...ab...b}_{z}$$

Thus: $y = a^k, \quad 1 \leq k \leq m$

$$x \, y \, z = a^m b^m \qquad y = a^k, \quad 1 \leq k \leq m$$

From the Pumping Lemma: $x \, y^i \, z \in L$

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

Thus: $x \, y^2 \, z \in L$

$$x \, y \, z = a^m b^m \qquad y = a^k, \quad 1 \le k \le m$$

From the Pumping Lemma: $x \, y^2 \, z \in L$

$$\overbrace{\qquad\qquad m+k \qquad\qquad}^{} \quad \overbrace{m}^{}$$

$$xy^2 z = \underbrace{a...a}_{x} \underbrace{a...a}_{y} \underbrace{a...a}_{y} \underbrace{a...ab...b}_{z} \in L$$

Thus: $a^{m+k} b^m \in L$

$$a^{m+k}b^m \in L \qquad k \geq 1$$

BUT: $\quad L = \{a^n b^n : n \geq 0\}$

$$\Downarrow$$

$$a^{m+k}b^m \notin L$$

CONTRADICTION!!!

Therefore: Our assumption that L
is a regular language is not true

Conclusion: L is not a regular language

END OF PROOF

Non-regular language $\{a^n b^n : \ n \geq 0\}$

Regular languages

$$L(a^* b^*)$$

More Applications

of

the Pumping Lemma

BLM2502 Theory of Computation – Regular Pumping Examples

The Pumping Lemma:

- Given a infinite regular language L

- there exists an integer m (critical length)

- for any string $w \in L$ with length $|w| \geq m$

- we can write $w = x\, y\, z$

- with $|x\, y| \leq m$ and $|y| \geq 1$

- such that: $x\, y^i\, z \in L$ $\qquad i = 0, 1, 2, \ldots$

Non-regular languages

$$L = \{vv^R : v \in \Sigma*\}$$

$a^m b^m b^m a^m$

Regular languages

$xyz = aa\underline{bba}a^u$

$y = a$

$xy^2z = ab^{m+k}b^m a^m$

Theorem: The language

$$L = \{vv^R : v \in \Sigma^*\} \qquad \Sigma = \{a,b\}$$

is not regular

Proof: Use the Pumping Lemma

$$L = \{vv^R : v \in \Sigma^*\}$$

Assume for contradiction
that L is a regular language

Since L is infinite
we can apply the Pumping Lemma

$$L = \{vv^R : v \in \Sigma*\}$$

Let m be the critical length for L

Pick a string w such that: $w \in L$

and length $|w| \geq m$

$a^n a^m$

$\underset{x}{a \, a} \underset{y}{a} \underset{z}{a \, a}$

$a^{m+k} a^m \neq a^{2n}$

while k odd

We pick $w = a^m b^m b^m a^m$

From the Pumping Lemma:

we can write: $w = a^m b^m b^m a^m = x\ y\ z$

with lengths: $|x\ y| \le m, \quad |y| \ge 1$

$$\overset{\displaystyle m}{\overbrace{\qquad\qquad}} \quad \overset{\displaystyle m}{\overbrace{\quad}}\overset{\displaystyle m}{\overbrace{\quad}}\overset{\displaystyle m}{\overbrace{\quad}}$$

$$w = xyz = \underbrace{a...a}_{x}\underbrace{a...a}_{y}\underbrace{...ab...bb...ba...a}_{z}$$

Thus: $y = a^k, \quad 1 \le k \le m$

$$x \, y \, z = a^m b^m b^m a^m \qquad y = a^k, \quad 1 \le k \le m$$

From the Pumping Lemma: $\quad x \, y^i \, z \ \in L$

$$i = 0, 1, 2, \dots$$

Thus: $x \, y^2 \, z \ \in L$

$$x \, y \, z = a^m b^m b^m a^m \qquad y = a^k, \quad 1 \le k \le m$$

From the Pumping Lemma: $\quad x \, y^2 \, z \; \in L$

$$xy^2z = \overbrace{a...aa...a}^{m+k}a...a...ab...b\overbrace{b...b}^{m}\overbrace{a...a}^{m} \in L$$

$$xy^2z = \underbrace{a...a}_{x}\underbrace{a...a}_{y}\underbrace{a...a}_{y}...\underbrace{ab...bb...ba...a}_{z} \in L$$

Thus: $\quad a^{m+k}b^m b^m a^m \;\; \in L$

$$a^{m+k} b^m b^m a^m \ \in L \qquad k \geq 1$$

BUT: $\quad L = \{ v v^R : v \in \Sigma^* \}$

$$\Downarrow$$

$$a^{m+k} b^m b^m a^m \ \notin L$$

CONTRADICTION!!!

Therefore: Our assumption that L
is a regular language is not true

Conclusion: L is not a regular language

END OF PROOF

Non-regular languages
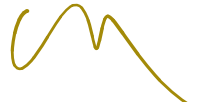
$$L = \{a^n b^l c^{n+l} : n, l \geq 0\}$$

$a^m b^m c^{2m} = \textcircled{b}{\to}\cancel{y}$

$y = c$

$a^{m+2} b^m c^{2m} \neq a^m b^m c^{2m}$

Regular languages

Theorem: The language

$$L = \{a^n b^l c^{n+l} : n, l \geq 0\}$$

is not regular

Proof: Use the Pumping Lemma

$$L = \{a^n b^l c^{n+l} : n, l \geq 0\}$$

Assume for contradiction
that L is a regular language

Since L is infinite
we can apply the Pumping Lemma

$$L = \{a^n b^l c^{n+l} : n, l \geq 0\}$$

Let m be the critical length of L

Pick a string w such that: $w \in L$ and

length $|w| \geq m$

We pick $w = a^m b^m c^{2m}$

From the Pumping Lemma:

We can write $w = a^m b^m c^{2m} = x\,y\,z$

With lengths $|x\,y| \le m, \quad |y| \ge 1$

$$w = xyz = \underbrace{a...aa...aa...a}_{m} \underbrace{b...b}_{m} \underbrace{c...cc...c}_{2m}$$

$$\underbrace{}_{x}\ \underbrace{}_{y} \qquad \underbrace{}_{z}$$

Thus: $y = a^k, \quad 1 \le k \le m$

$$x \, y \, z = a^m b^m c^{2m} \qquad y = a^k, \quad 1 \le k \le m$$

From the Pumping Lemma: $\quad x \, y^i \, z \ \in L$

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

Thus: $\quad x \, y^0 \, z = xz \ \in \ L$

$$x \ y \ z = a^m b^m c^{2m} \qquad y = a^k, \ \ 1 \le k \le m$$

From the Pumping Lemma: $\quad xz \ \in \ L$

$$xz = \overbrace{a...a}^{m+k}\overbrace{a...a}^{}\overbrace{b...b}^{m}\overbrace{c...c}^{2m}c...c \ \in \ L$$

$$\underbrace{a...a}_{x}\underbrace{a...ab...bc...cc...c}_{z}$$

Thus: $\quad a^{m+k}b^m c^{2m} \ \in L$

$$a^{m+k}b^m c^{2m} \in L \qquad k \geq 1$$

BUT: $L = \{a^n b^l c^{n+l} : n, l \geq 0\}$

$$a^{m+k} b^m c^{2m} \notin L$$

CONTRADICTION!!!

Therefore: Our assumption that L
               is a regular language is not true

Conclusion: L is not a regular language

END OF PROOF

Non-regular languages

$$L = \{a^{n!} : n \geq 0\}$$

Regular languages

Theorem: The language $L = \{a^{n!} : n \geq 0\}$ is not regular

$$n! = 1 \cdot 2 \cdots (n-1) \cdot n$$

Proof: Use the Pumping Lemma

$$L = \{a^{n!} : \; n \geq 0\}$$

Assume for contradiction
that L is a regular language

Since L is infinite
we can apply the Pumping Lemma

$$L = \{a^{n!} : n \geq 0\}$$

Let m be the critical length of L

Pick a string w such that: $w \in L$

length $|w| \geq m$

We pick $w = a^{m!}$

From the Pumping Lemma:

We can write $w = a^{m!} = x\ y\ z$

With lengths $|x\ y| \le m, \quad |y| \ge 1$

$$w = xyz = a^{m!} = \overbrace{a...a}^{}\underbrace{a...a}_{}\overbrace{a...a}^{m}\underbrace{a...aa...a}_{m!-m}$$

$$\underbrace{\quad}_{x}\ \underbrace{\quad}_{y}\ \underbrace{\quad}_{z}$$

Thus: $y = a^k, \quad 1 \le k \le m$

$$x\ y\ z = a^{m!} \qquad y = a^{k}, \quad 1 \le k \le m$$

From the Pumping Lemma: $\quad x\ y^{i}\ z\ \in L$

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

Thus: $\quad x\ y^{2}\ z\ \in L$

$$x\,y\,z = a^{m!} \qquad y = a^k, \quad 1 \le k \le m$$

From the Pumping Lemma: $x\,y^2\,z \in L$

$$xy^2 z = \underbrace{a...a}_{x}\underbrace{a...a}_{y}\underbrace{a...a}_{y}\overbrace{\underbrace{a...a}_{}a...aa...a}^{m!-m}... \in L$$

$$xy^2 z = \overbrace{a...aa...aa...a}^{m+k}\overbrace{a...aa...a}^{m!-m} \in L$$

Thus: $a^{m!+k} \in L$

$$a^{m!+k} \in L \qquad 1 \le k \le m$$

Since: $$L = \{a^{n!} : n \ge 0\}$$
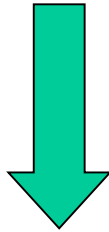
There must exist p such that:

$$m!+k = p!$$

However: $m!+k \leq m!+m$ for $m>1$

$$\leq m!+m!$$

$$< m!m+m!$$

$$= m!(m+1)$$

$$= (m+1)!$$

$$\Downarrow$$

$$m!+k < (m+1)!$$

$$\Downarrow$$

$$m!+k \neq p! \quad \text{for any} \quad p$$

$$a^{m!+k} \in L \qquad 1 \le k \le m$$

BUT: $\quad L = \{a^{n!} : n \ge 0\}$

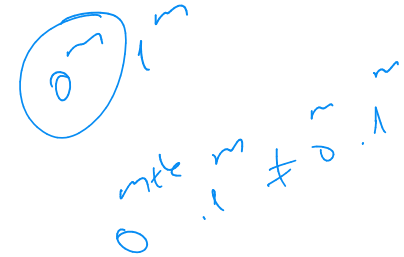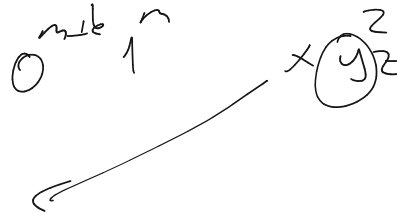$$a^{m!+k} \notin L$$

CONTRADICTION!!!

Therefore: Our assumption that L
            is a regular language is not true

Conclusion: L is not a regular language

END OF PROOF

Example 2

Theorem : $B = \{0^n 1^n \mid n \geq 0\}$ *is not regular.*

Proof. Assume the contrary that B is regular.

Let p be the pumping length. Then, $s = 0^p 1^p$ can be decomposed

as $s = xyz$ with $|y| \geq 1$, and $xy^n z \in B$ for any $n \geq 0$.

There can be three cases $y = 0^k$, $y = 0^k 1^l$, and $y = 1^l$, for some

nonzero k, l. In each case, we can eazily see that $xy^n z \notin B$, which

leads to contradiction. $(n \geq 2)$

Example 3

Theorem : $C = \{w \,|\, w \text{ has an equal number of 0s and 1s}\}$

 is not regular.

Proof.   Assume the contrary that C is regular.

Let p be the pumping length. Then, $s = 0^p 1^p$ can be decomposed

as $s = xyz$ with $|y| \geq 1$ and $|xy| \leq p,$ and $xy^n z \in C$ for any $n \geq 0.$
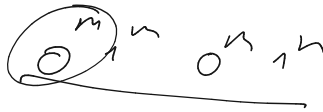
Then we must have $y = 0^k$ for some nonzero $k.$

We can eazily see that $xy^n z \notin C,$ which contradicts the assumption.
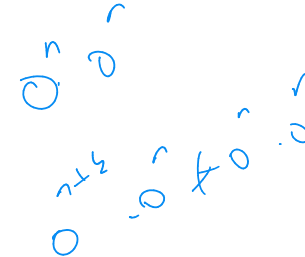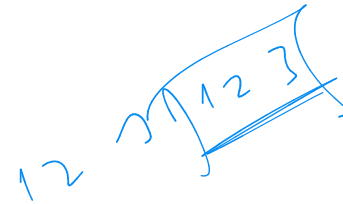
Alternative proof of Example 3

- The class of regular languages is closed under the intersection operation. This is easy to prove if we run two DFAs parallelly and accept only string which are accepted by both of the DFAs.

- Now, assume C is regular.

- Then, $C \cap 0^*1^* = B$ is also regular.

- This contradicts what we proved in Example 1.38.???

Example 4

Theorem : $F = \{ww \mid w \in \{0,1\}^*\}$ *is not regular.*

Proof. Assume the contrary that F is regular.

Let p be the pumping length and let $s = 0^p 1 0^p 1 \in F$. This s can be split

into pieces like $s = xyz$ with $|y| \geq 1$ and $|xy| \leq p$, and $xy^n z \in F$ for any $n \geq 0$.

Then we must have $y = 0^k$ for some nonzero k.

We can eazily see that $xy^n z \notin F$, which contradicts the assumption.

Example 5

1^{m^2+k}

$y = 1^k$

$1^{n^2+k} \neq 1^{n^2}$

$n^2 + k \neq p^2$

Theorem : $D = \{1^{n^2} \mid n \geq 0\}$ *is not regular.*

Proof. Assume the contrary that D is regular.

Let p be the pumping length and let $s = 1^{p^2} \in D$. This s can be split into pieces like $s = xyz$ with $|y| \geq 1$ and $|xy| \leq p$, and $xy^n z \in D$ for any $n \geq 0$.

The length of $xy^n z$◆ grows linearly with n, while the lengths of strings in D grows as $0, 1, 4, 9, 16, 25, 36, 49, \ldots$
These two facts are incompatible as can be easily seen.

Example 6 (Pumping Down)

Theorem: $E = \{0^i 1^j \mid i > j\}$ *is not regular.*

Proof. Assume the contrary that E is regular.

Let p be the pumping length and let $s = 0^{p+1} 1^p$. This s can be split

into $s = xyz$ with $|y| \geq 1$ and $|xy| \leq p$, and $xy^n z \in E$ for any $n \geq 0$.

Then we must have $y = 0^k$ for some nonzero k.

We can eazily see that $xy^0 z = xz \notin E$, which contradicts the assumption.

Example 6 (Differential Encoding)

Claim : $D = \{w \mid w \text{ contains equal number of occurrence s of the substrings } 01 \text{ and } 10\}$ *is regular.*