COM2109

Automata

Rob Hierons

Example 1

Write DFA or NFA to recognize:

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

$$L = \{w : w \text{ has equal number of 0s and 1s}\}$$

Example 1

Language

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

Consider the following prefixes of strings and sequences of 'b' then accepted

- a: goes to state where only 'b' accepted
- aa: goes to state where only 'bb' accepted
- aaa: goes to state where only 'bbb' accepted

… - would need **infinitely many** states

Example 1

Write DFA or NFA to recognize:

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

$$L = \{w : w \text{ has equal number of 0s and 1s}\}$$

- Machines would have to keep track of an unlimited number of possibilities
- Can't be done with finite number of states

Example 2

Write DFA or NFA to recognize:

$$L = \{w : w \text{ has equal number of occurrences of 01 and 10 as substrings}\}$$

Note that 01 cannot follow 01 without introducing 10 (for example, 0101 has a 10).

Same for 1010 or 1110001100

Example 2

Write DFA or NFA to recognize:

$$L = \{w : w \text{ has equal number of occurrences of } 01 \text{ and } 10 \text{ as substrings}\}$$

- Number of possible strings is also infinite, but:
- CAN be done with finite number of states

Non-regular languages

Non-regular languages

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

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

Regular languages

$$a^*b$$

$$b^*c+a$$

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

$$etc...$$

How can we prove that a language L is regular?

How can we prove that a language L is regular?

Prove that there is a DFA or NFA or RE that accepts L

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

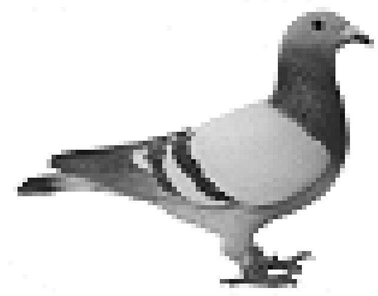
(there is an infinite number of strings)

Solution: use the Pumping Lemma !!!

Pumping lemma

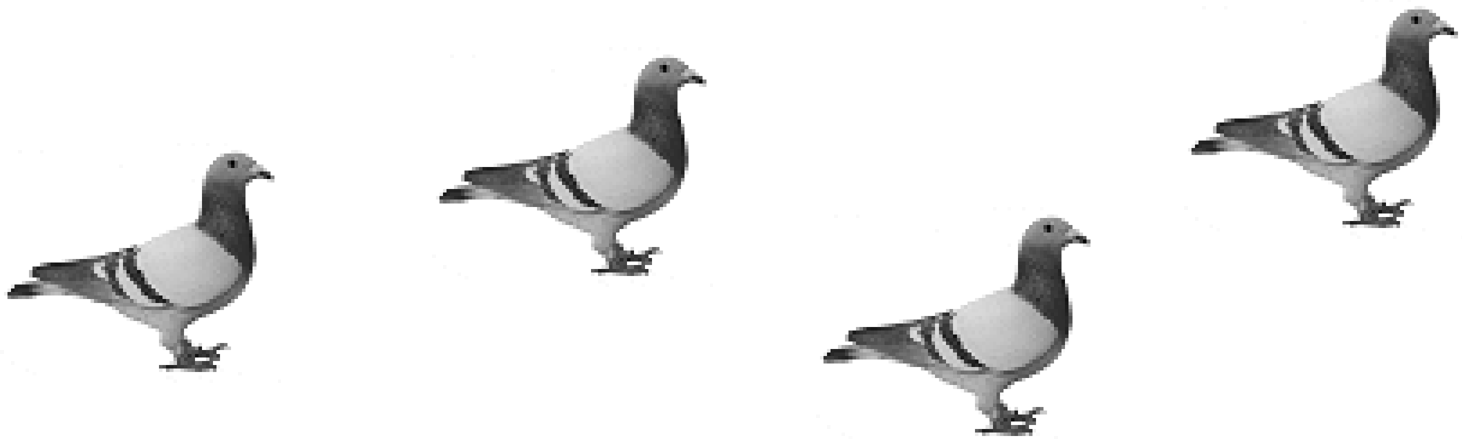
- Theorem that states that all regular languages have a special property

- If we can show that a language does not have that property, we are guaranteed that it is **not regular**

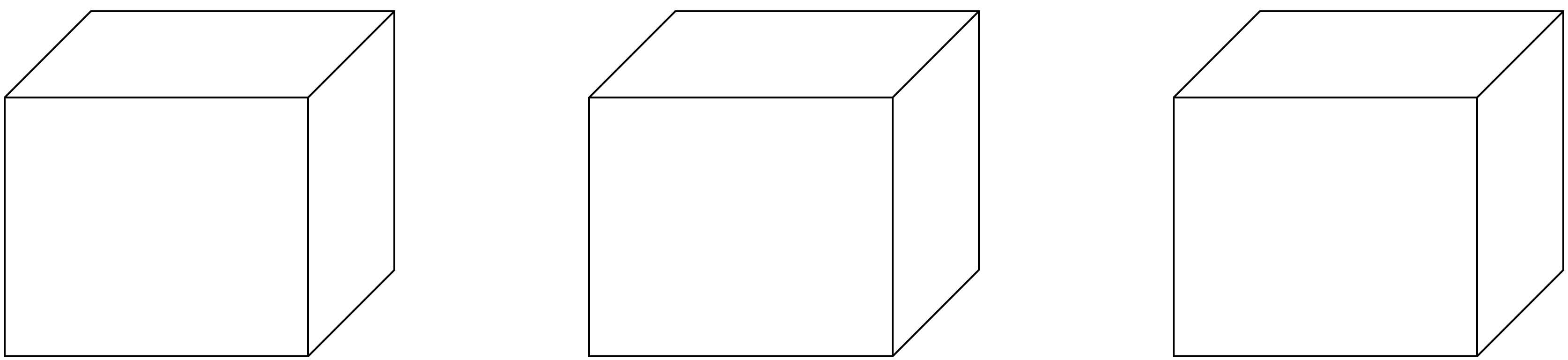


The Pigeonhole Principle

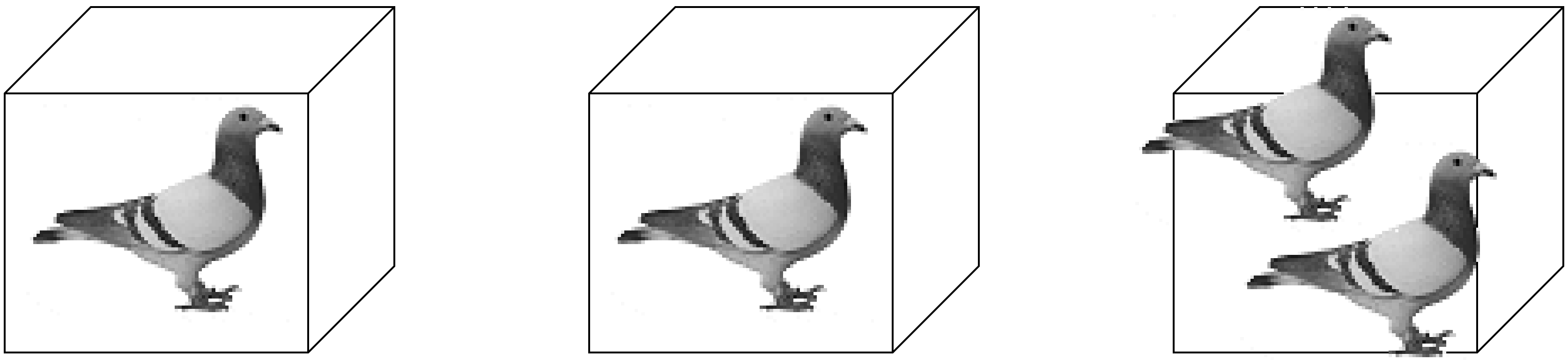
4 pigeons



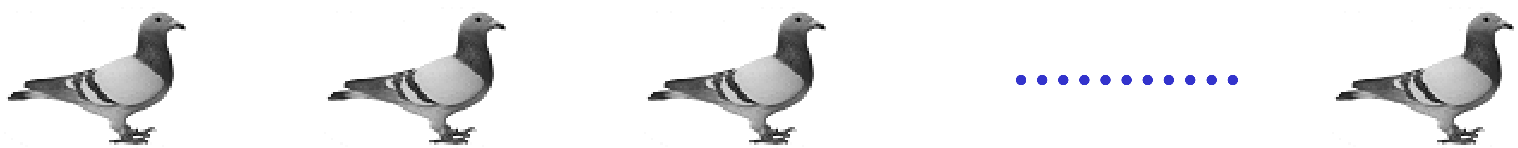
3 pigeonholes



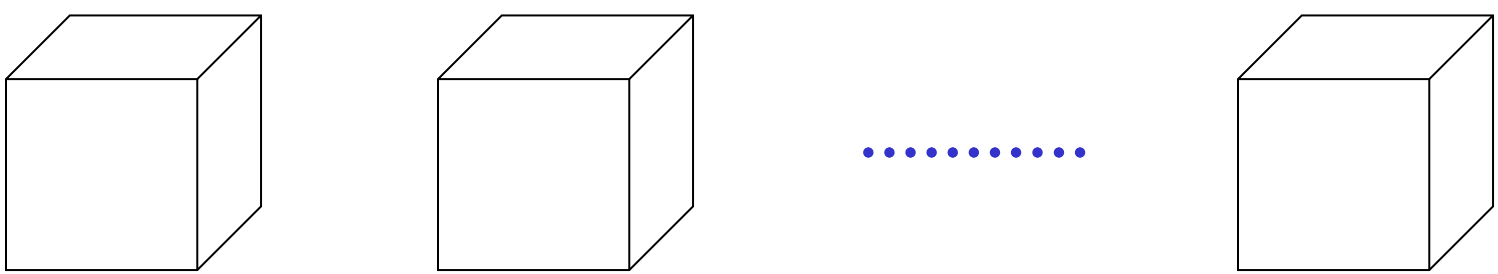
A pigeonhole must contain at least two pigeons



n pigeons



m pigeonholes $\quad n > m$



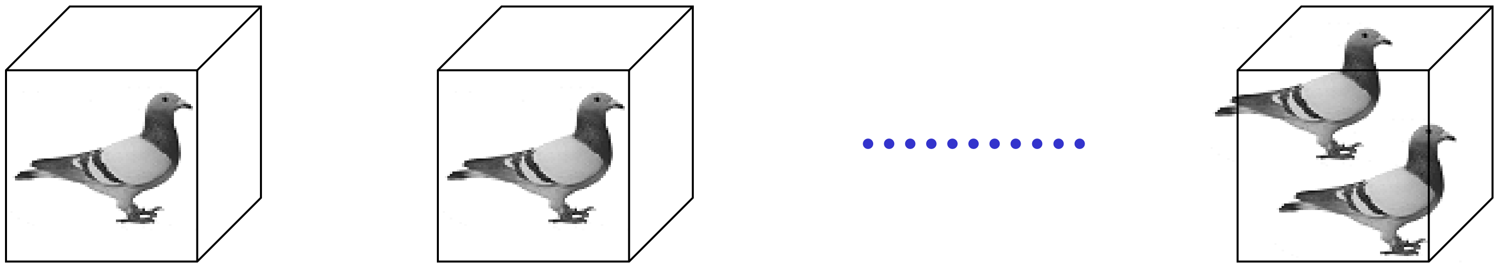
The Pigeonhole Principle

n pigeons

m pigeonholes

$$n > m$$

There is a pigeonhole with at least 2 pigeons

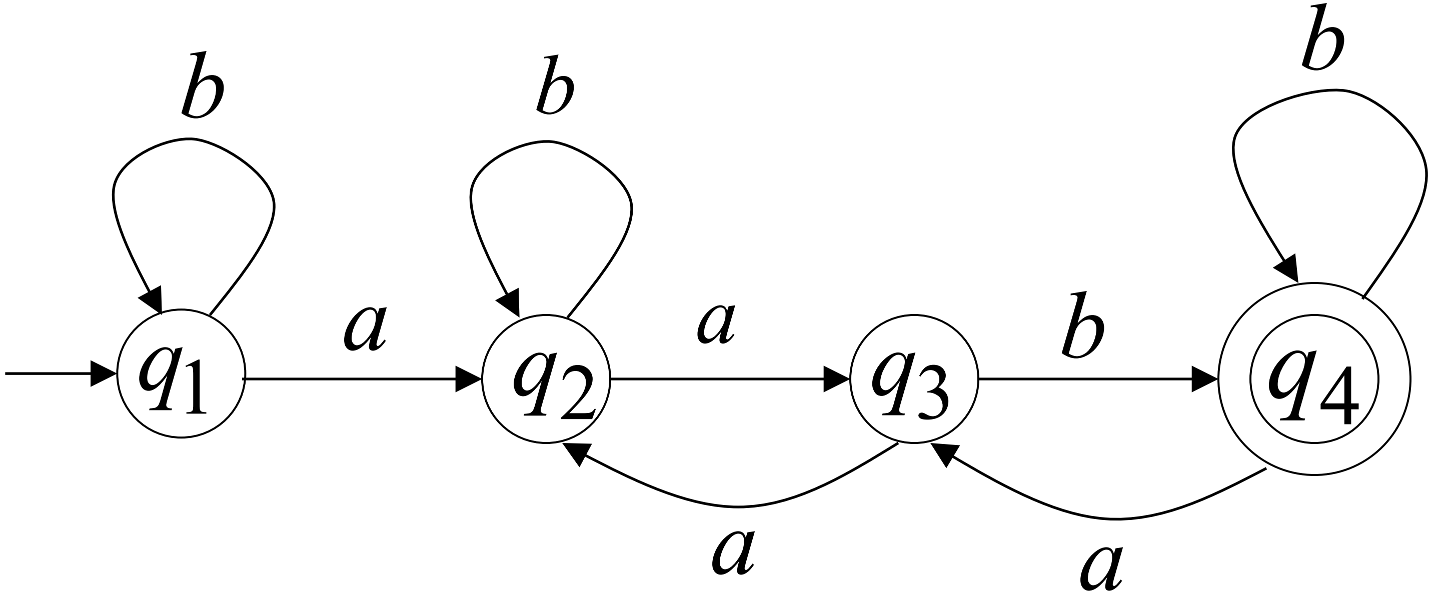


The Pigeonhole Principle

and

DFAs

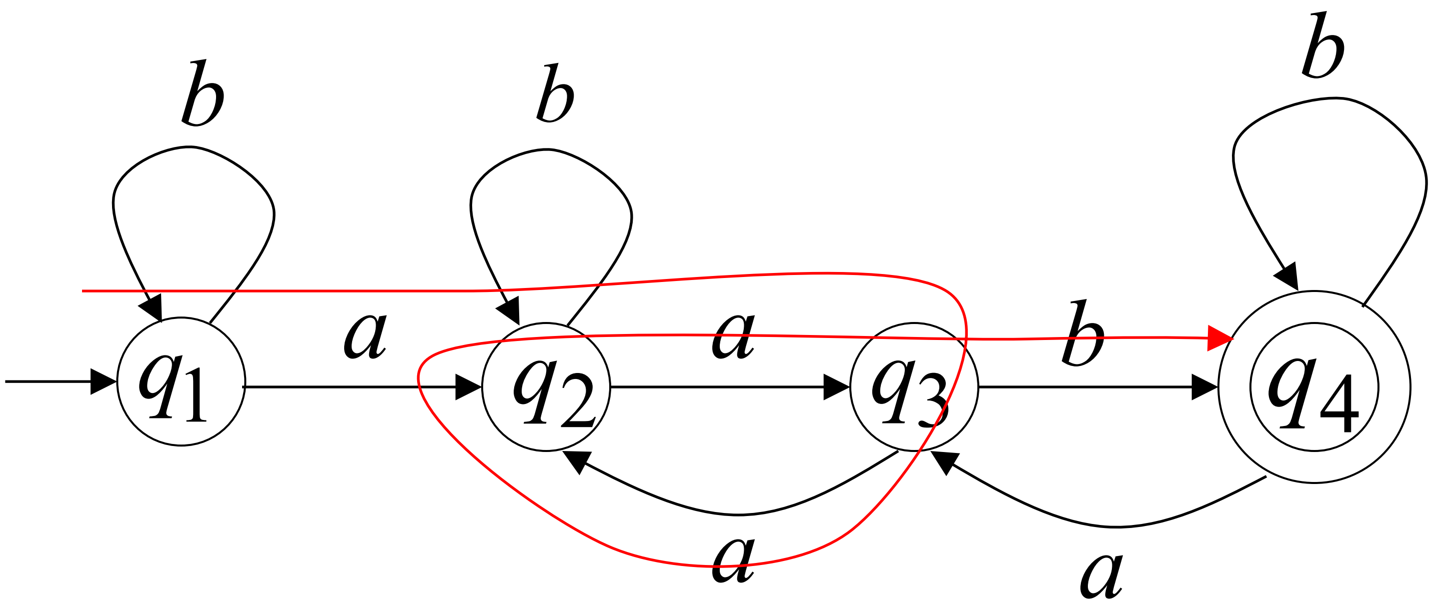
Consider a DFA with 4 states



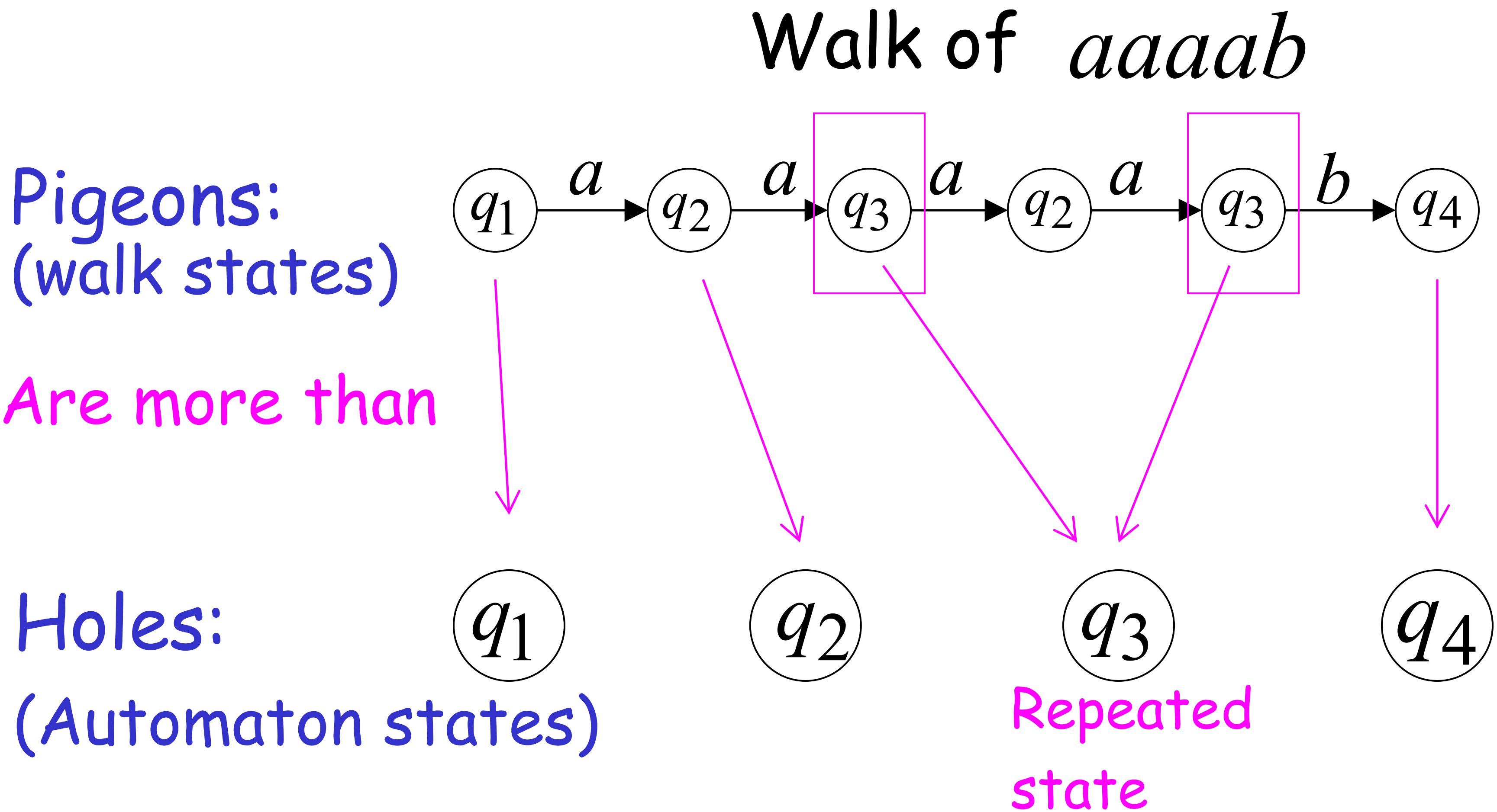
Consider the walk of a "long" string: $aaaab$
(length at least 4)

A state is repeated in the walk of $aaaab$

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



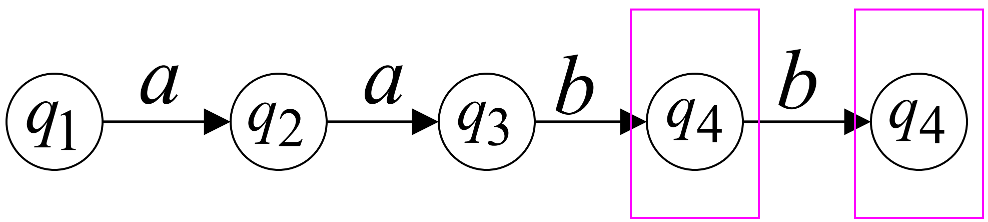
The state is repeated as a result of the pigeonhole principle

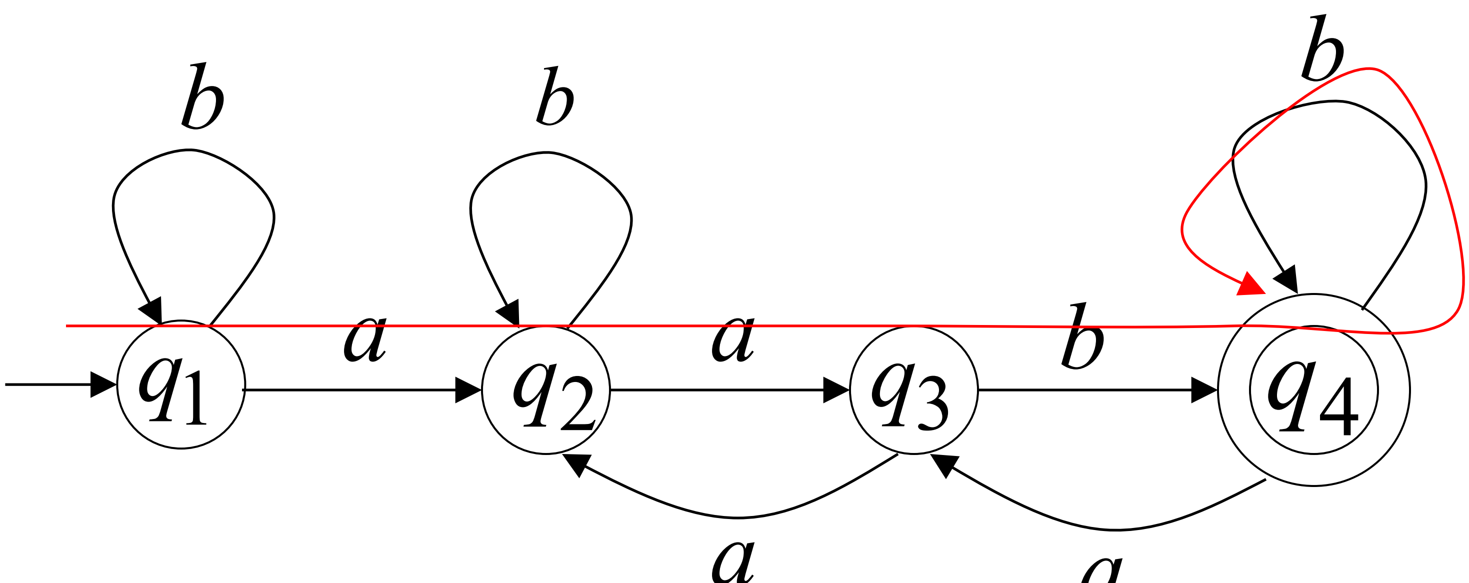


Consider the walk of a “long” string: $aabb$
(length at least 4)

Due to the pigeonhole principle:

A state is repeated in the walk of $aabb$





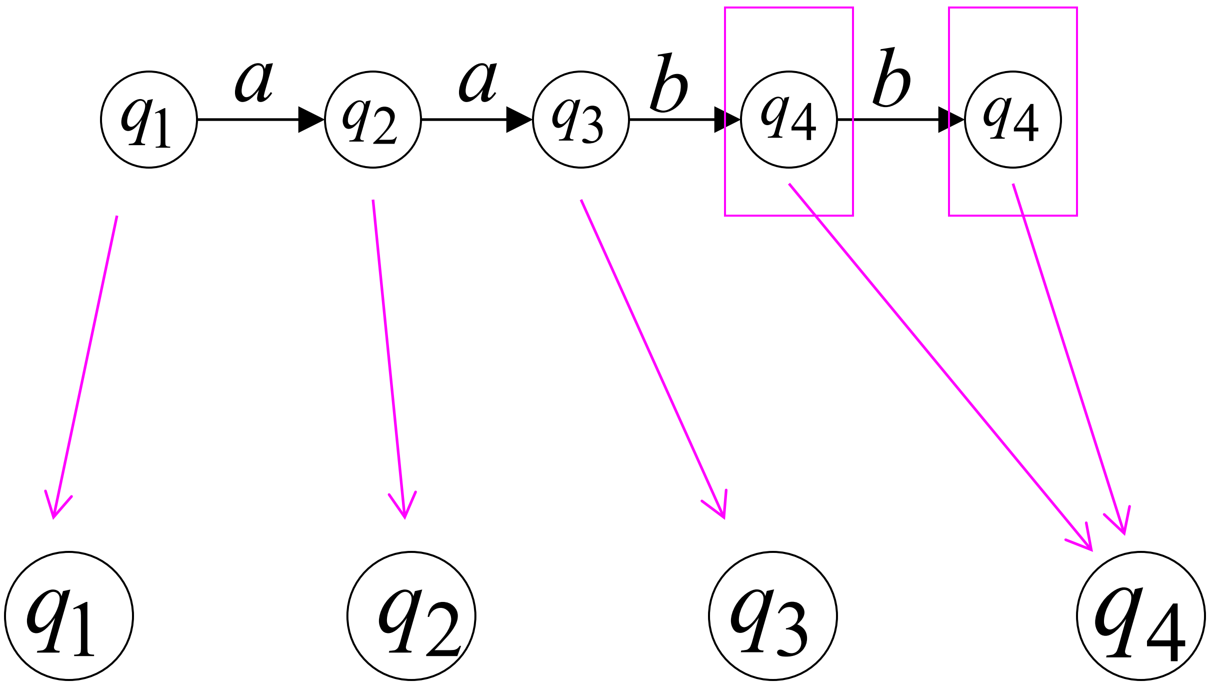
The state is repeated as a result of the pigeonhole principle

Walk of $aabb$

Pigeons:
(walk states)

Are more than

Holes:
(Automaton states)

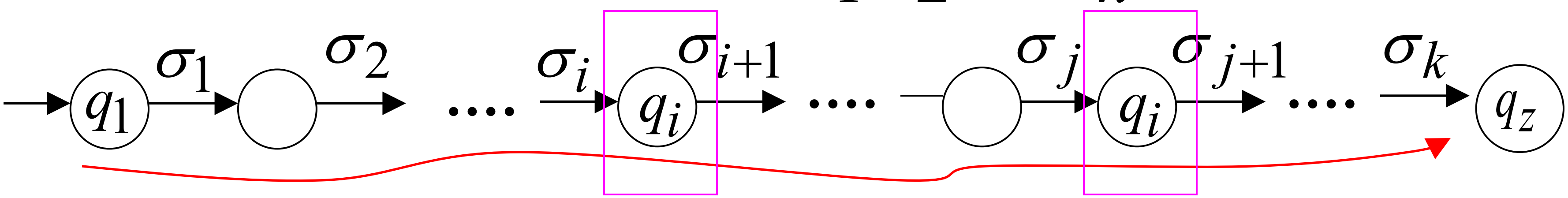


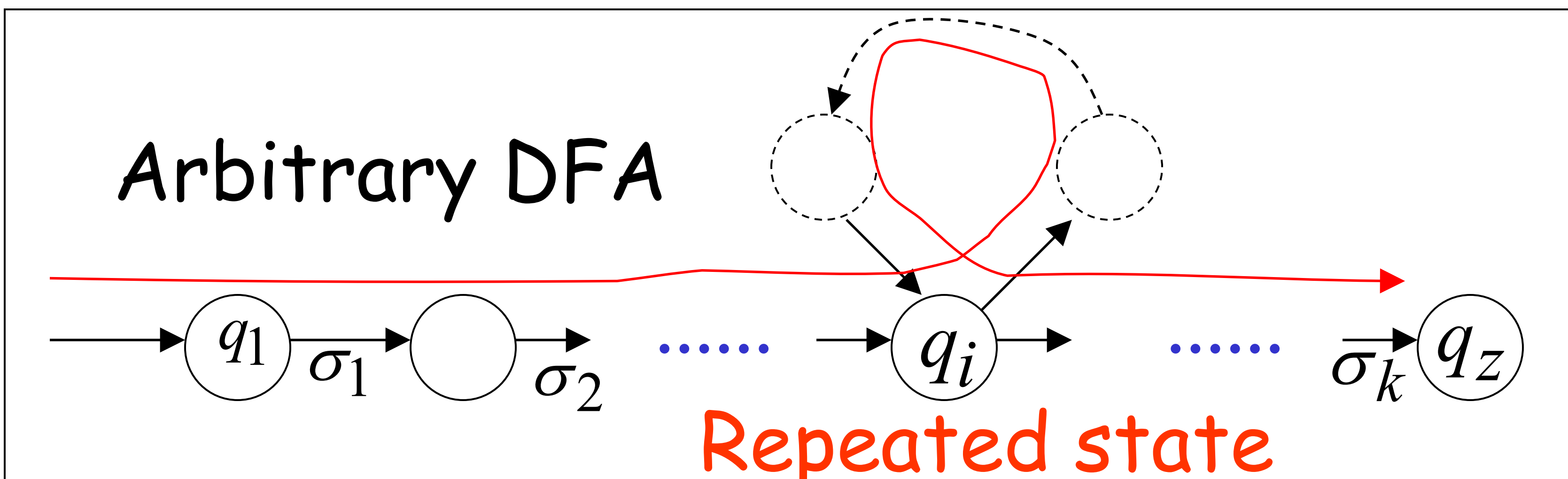
Automaton States

Repeated state

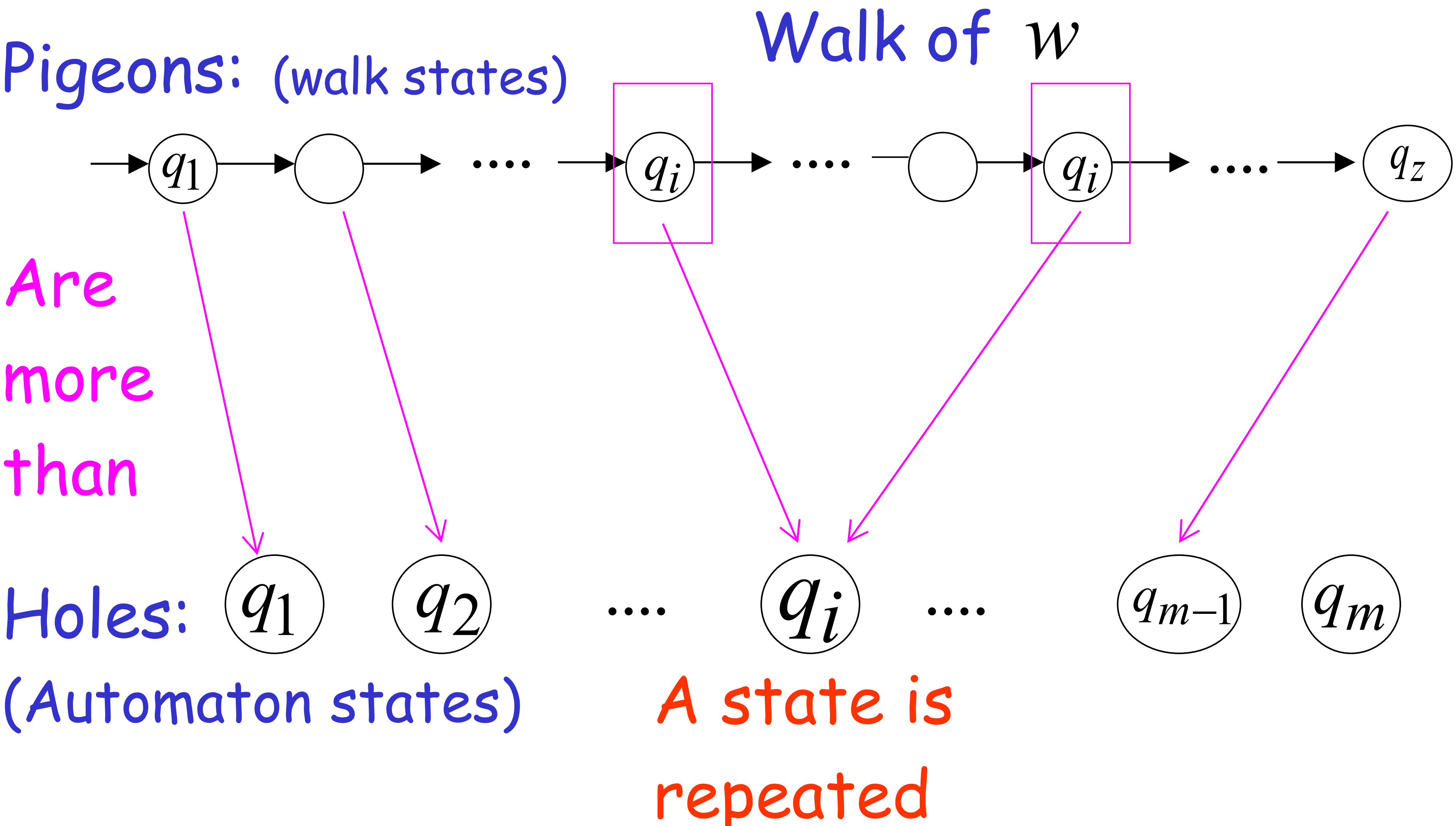
In General: If $|w| \geq$ #states of DFA,
by the pigeonhole principle,
a state is repeated in the walk w

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





$|w| \geq$ #states of DFA = m



The Pumping Lemma

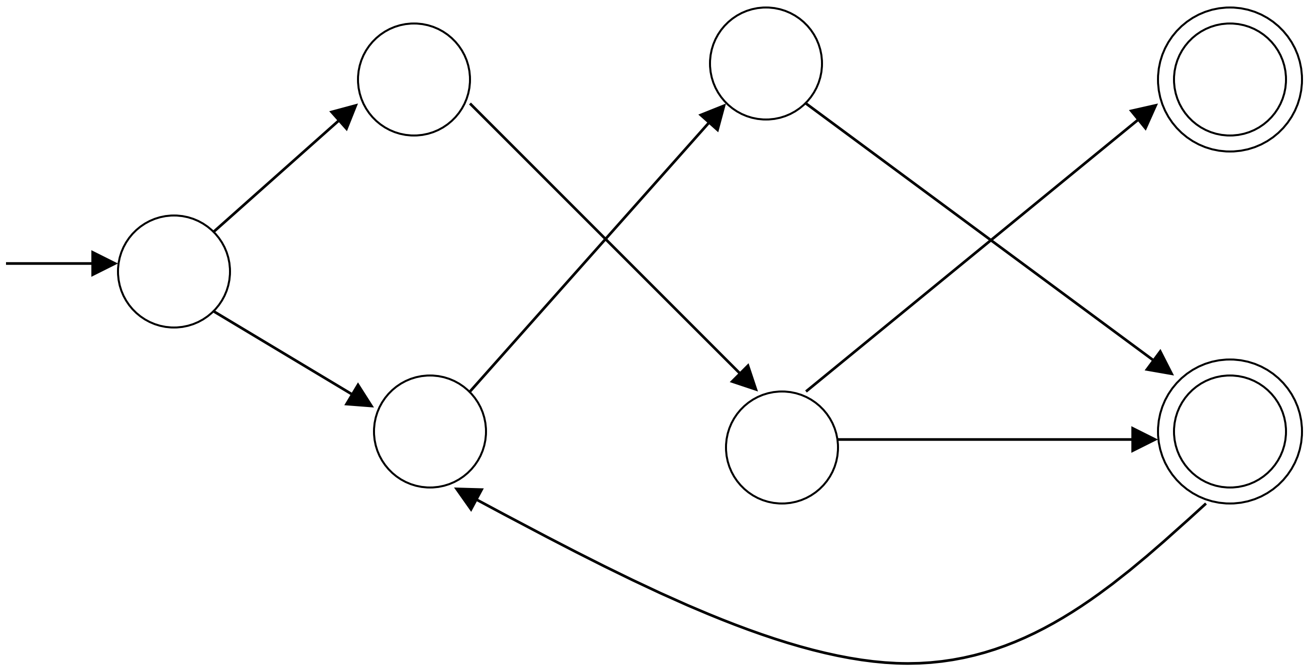
Pumping lemma

- Take an **in**finite language
- All strings in the language can be "pumped" if they are at least as long as a special value (**pumping length**)
- I.e., each such string contains a section that can be repeated any number of times with the resulting string remaining in the language

Take an infinite regular language L

(contains an infinite number of strings)

There exists a DFA that accepts L

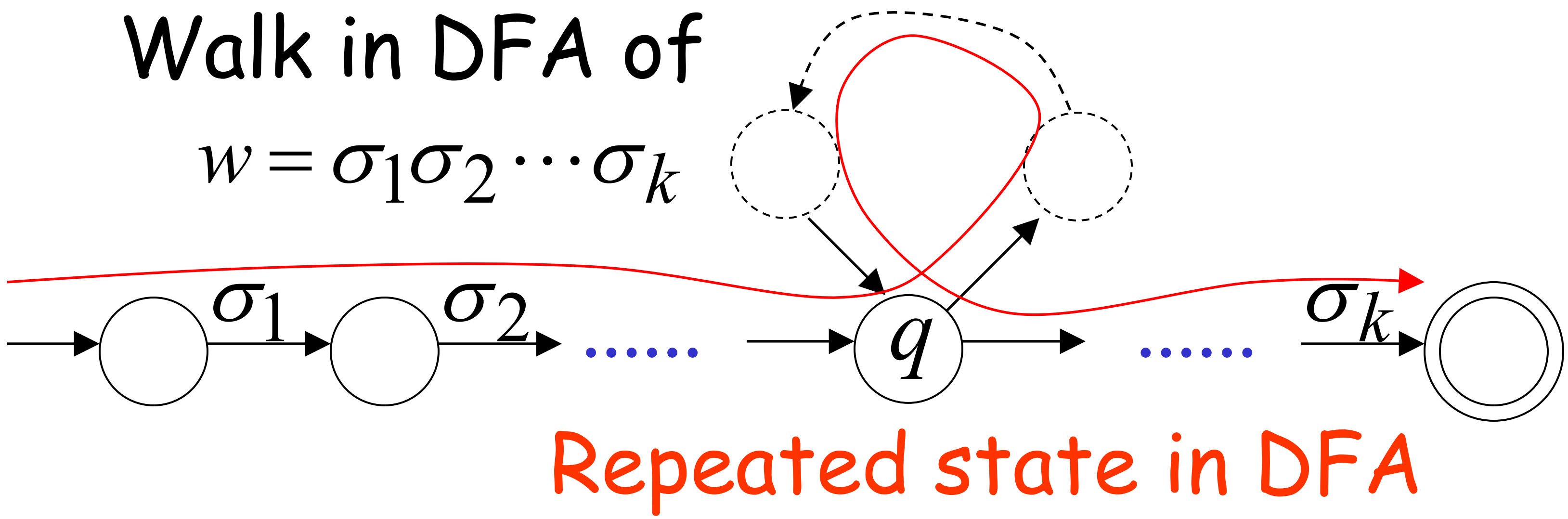


m
states

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

(number of states of DFA)

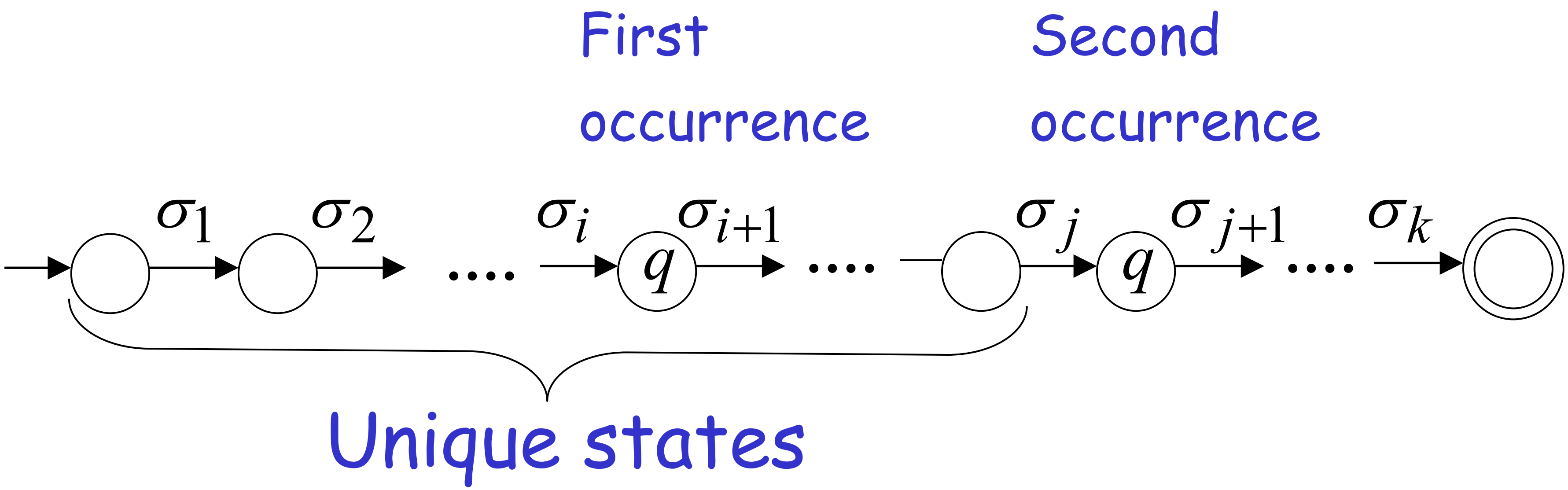
then, **at least** one state is repeated in the walk of w



There could be many states repeated

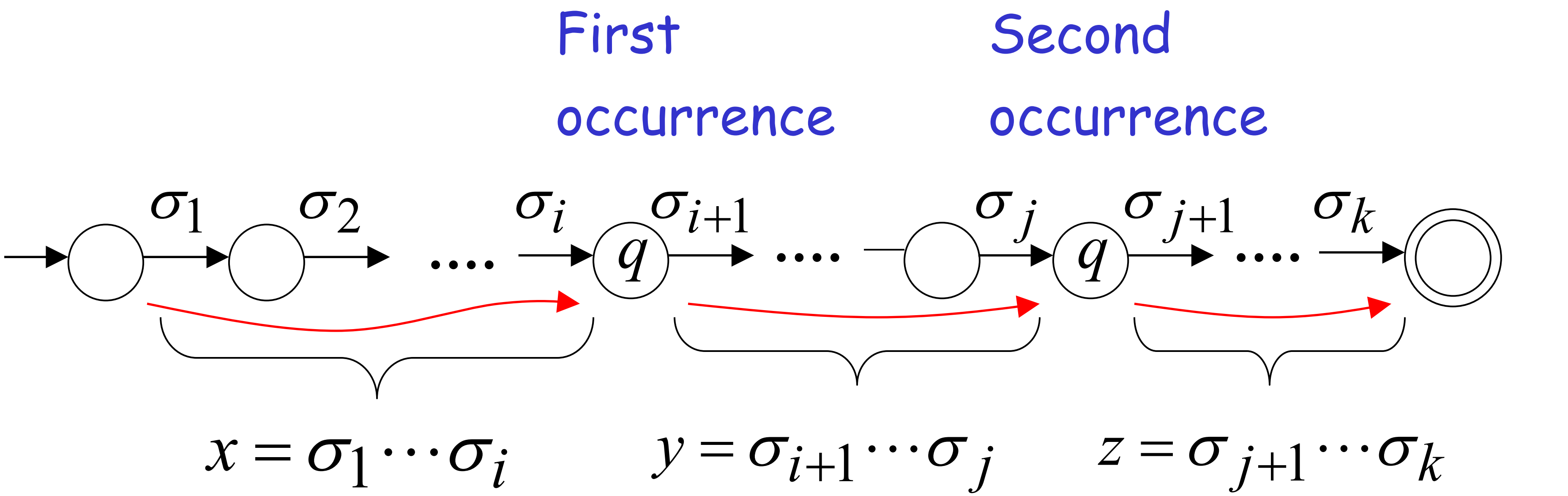
Take q to be the first state repeated

One dimensional projection of walk w:

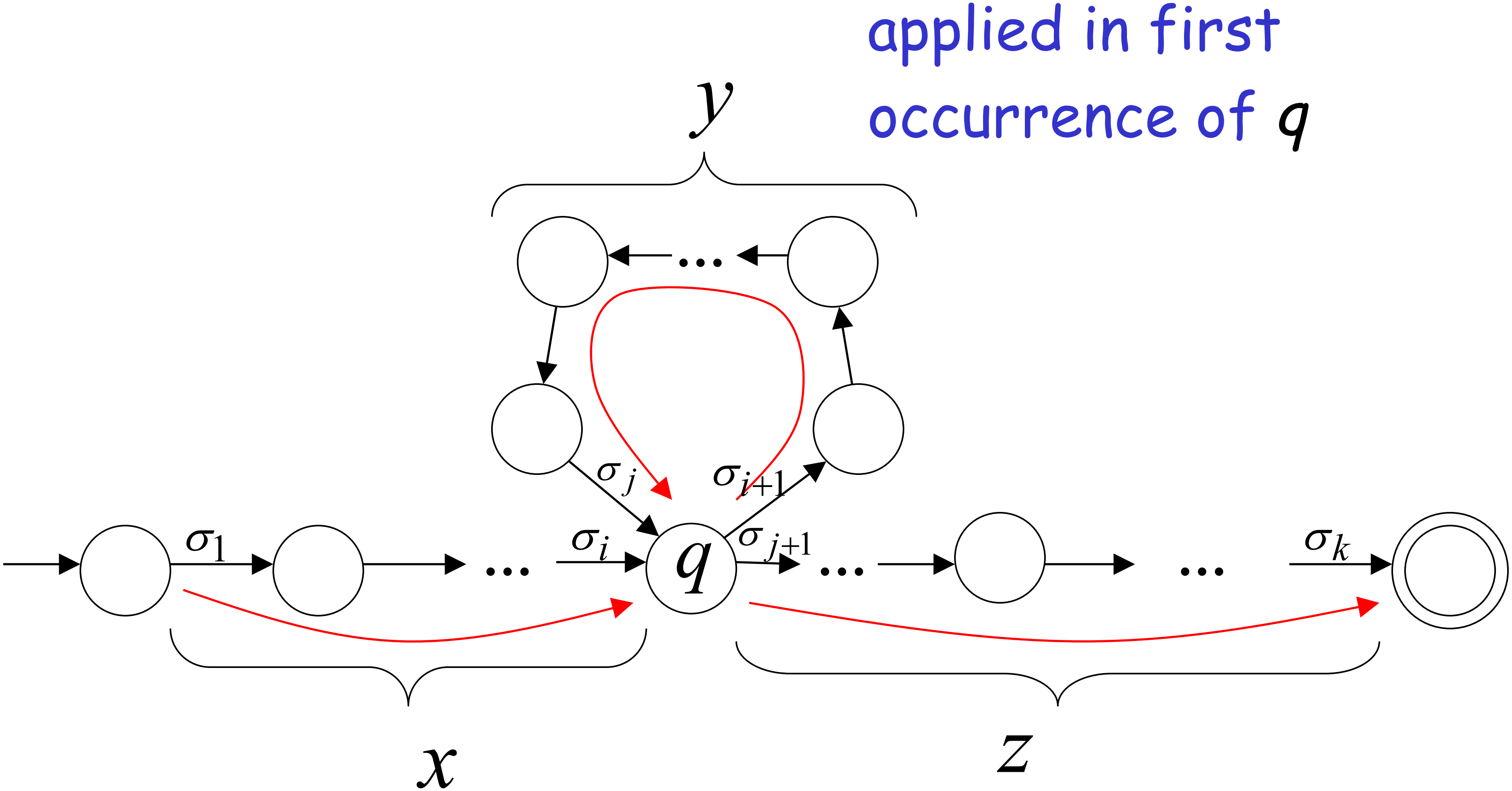


We can write $w = xyz$

One dimensional projection of walk w:

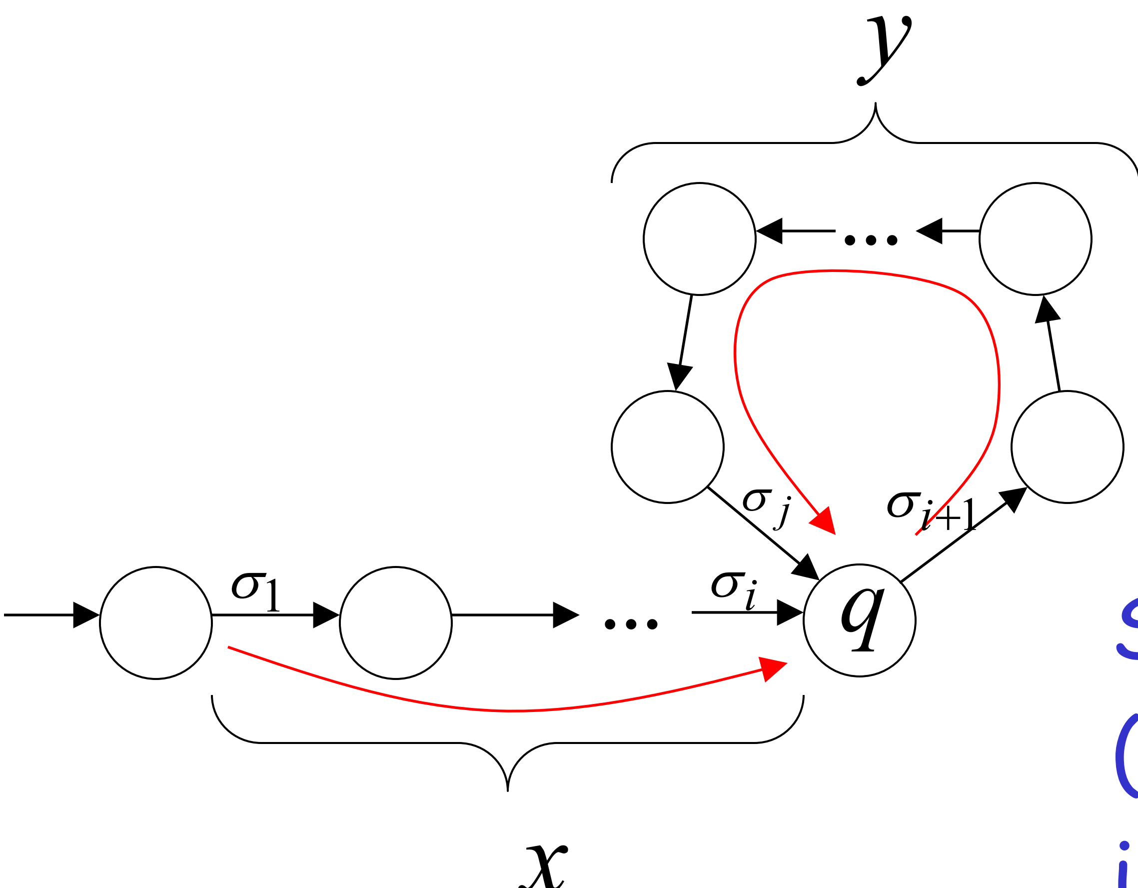


In DFA: $w = x\ y\ z$



applied in first
occurrence of q
y
σ_j
σ_{i+1}
σ_1
σ_i
q
σ_{j+1}
σ_k
x
z

Observation: length $|x\ y| \leq m$ number of states of DFA

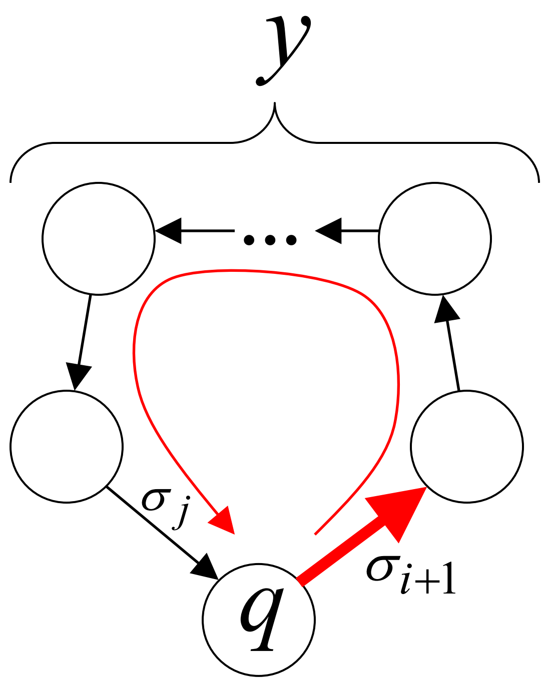


Unique States

Since in xy no (start) state is repeated

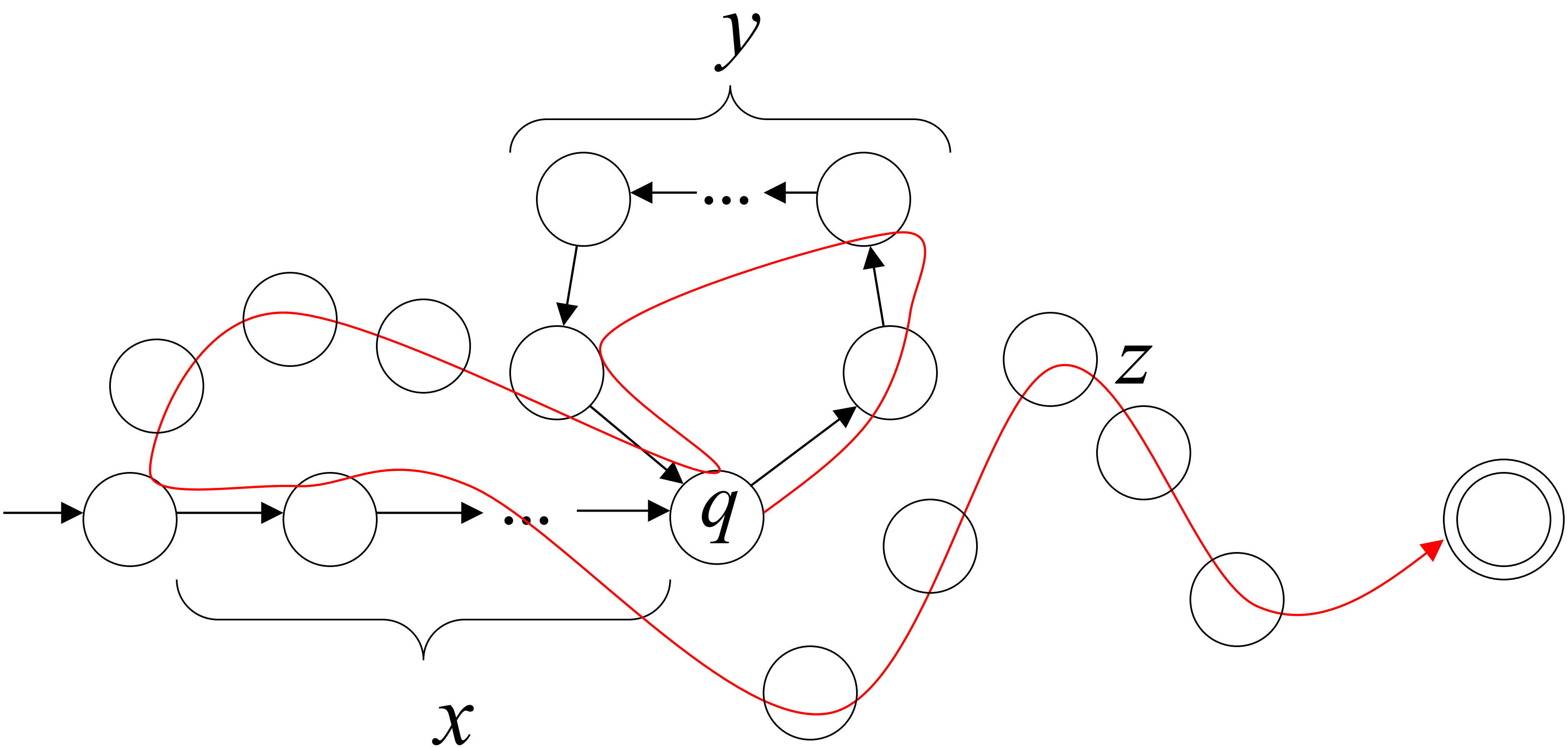
Observation: length $|y| \geq 1$

Since there is at least one transition in loop



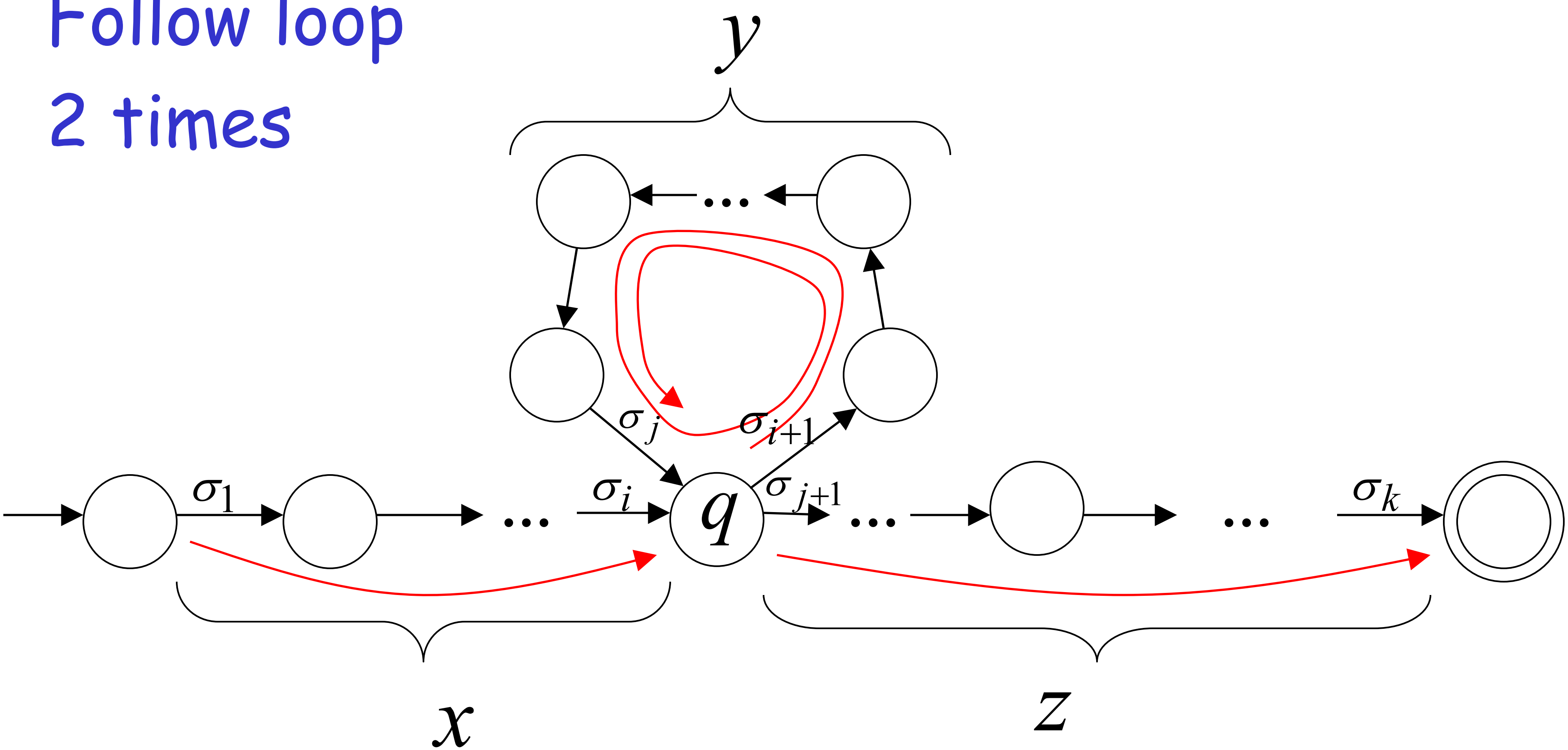
We do not care about the form of string z

z may actually overlap with the paths of x and y



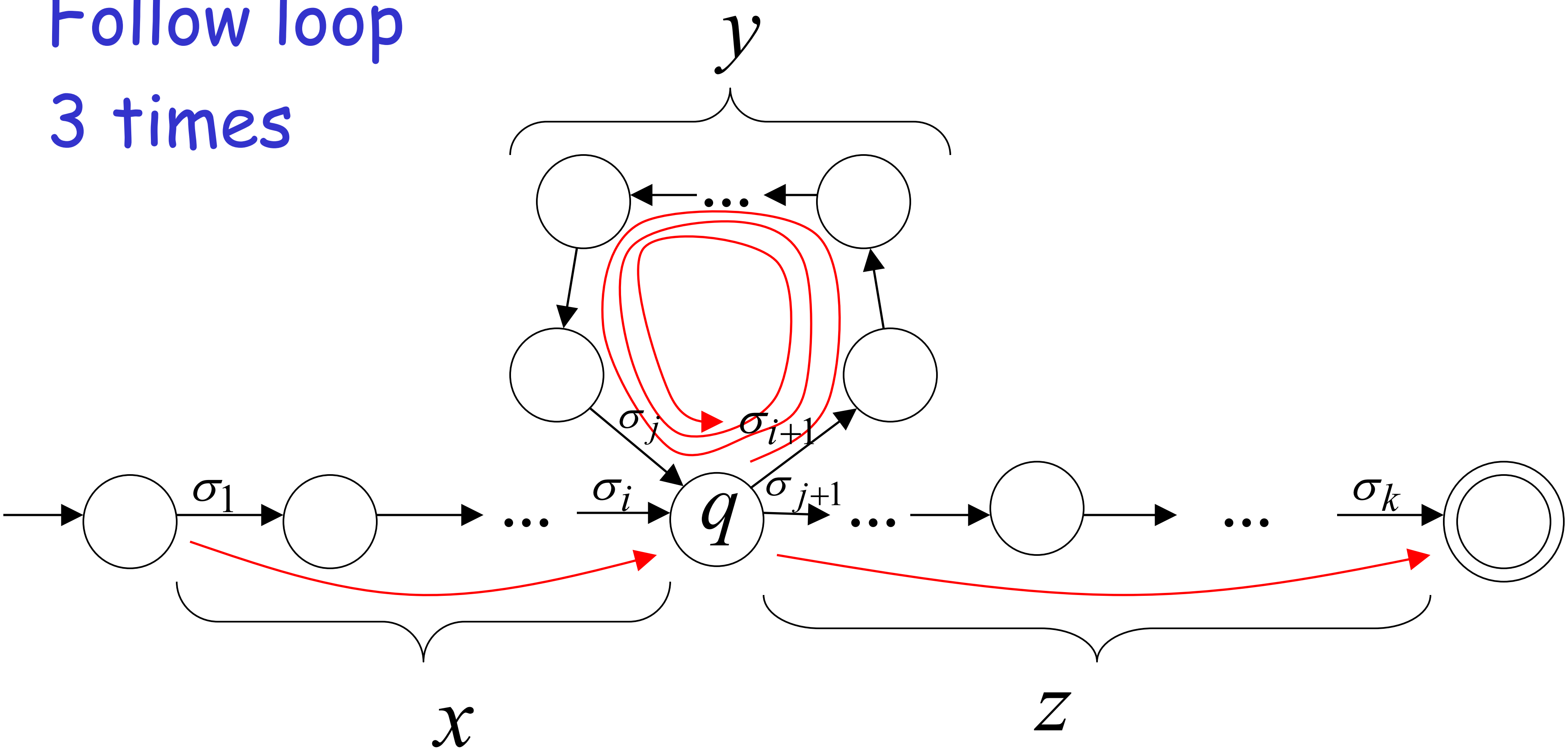
Additional string: The string $x\ y\ y\ z$ is accepted

Follow loop 2 times



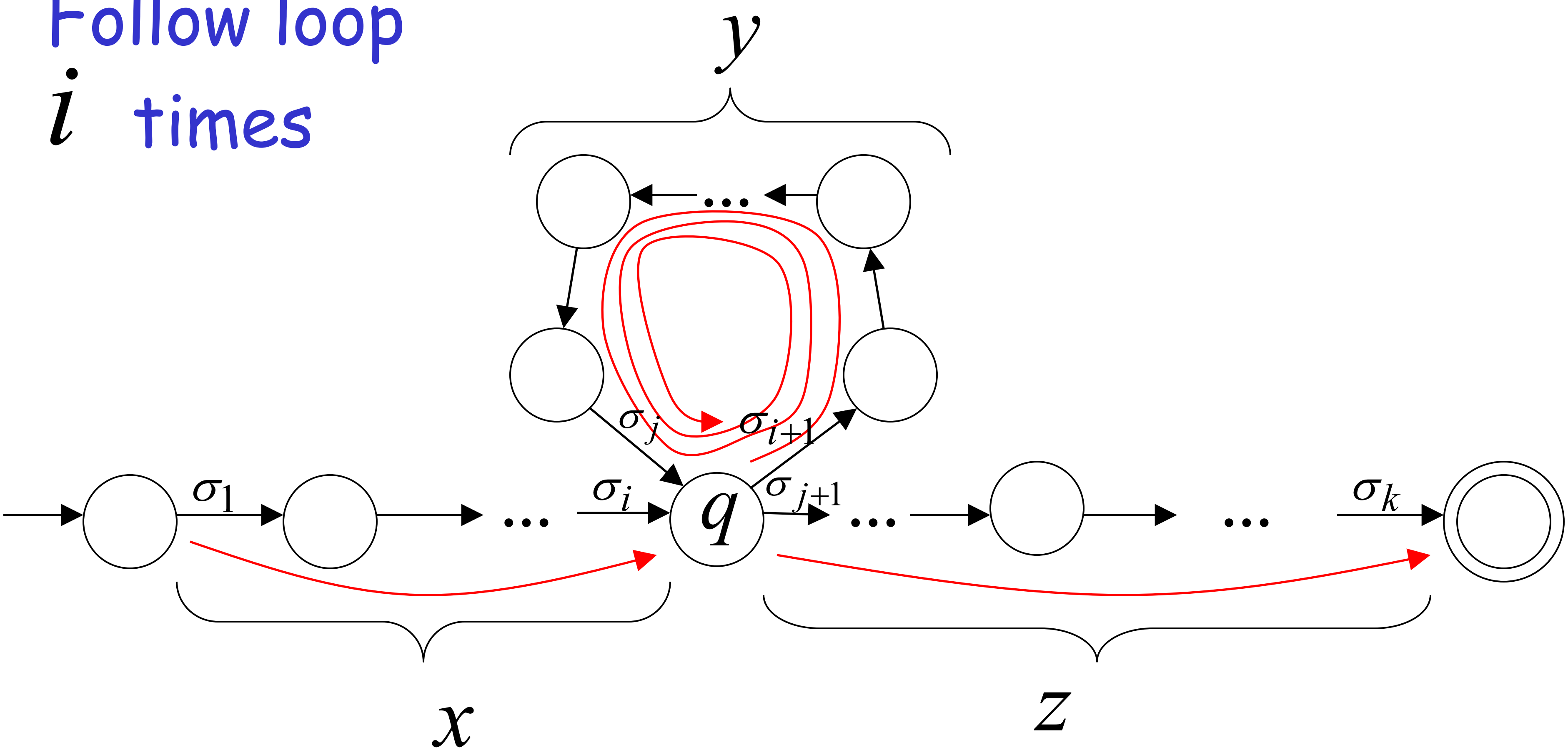
Additional string: The string $x\ y\ y\ y\ z$ is accepted

Follow loop 3 times



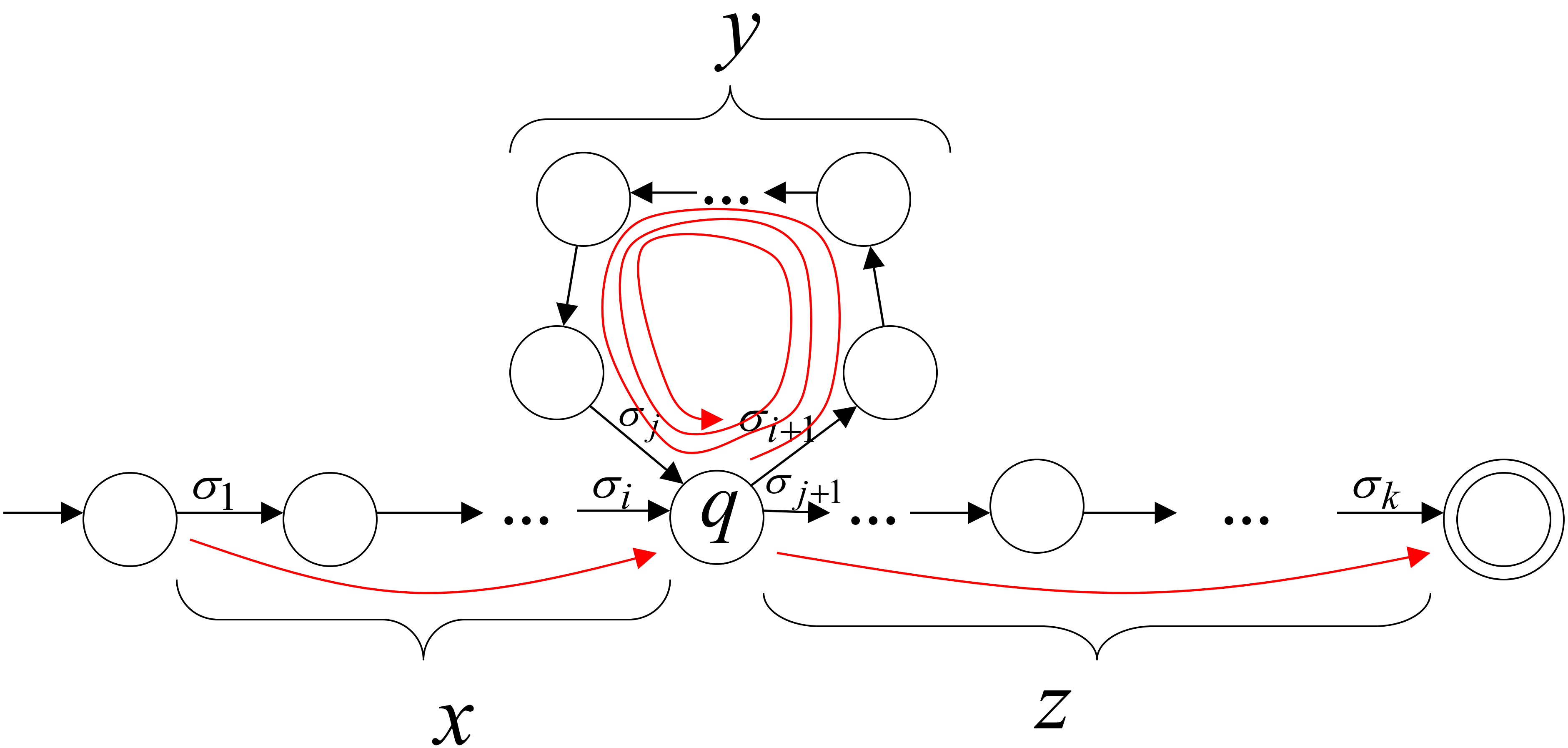
In General: The string $x\, y^i\, z$ is accepted $i = 0, 1, 2, \ldots$

Follow loop i times



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

Language accepted by the DFA



In other words, we described:

The Pumping Lemma !!!

The Pumping Lemma:

- Given an **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: $x\ y^i\ z \in L \qquad i = 0, 1, 2, \ldots$

Critical length m = Pumping length

Applications

of

the Pumping Lemma

Observation:

Every language of finite size has to be regular

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

How: build an NFA with epsilon-transitions from an initial state to automata accepting each individual 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
4. Therefore, L is not regular

Explanation of Step 3: How to get a contradiction

1. Let (unknown) m denote critical length for L

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

3. Write $w = xyz$

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

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 $|x\,y| \leq m, \quad |y| \geq 1$

$$w = xyz = a^m b^m = \overbrace{a...aa...aa...a}^{m}\overbrace{b...b}^{m}$$

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

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

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

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

$$xy^2z = \overbrace{\underbrace{a...a}_{x}\underbrace{a...a}_{y}\underbrace{a...a}_{y}a...a}^{m+k}\overbrace{b...b}^{m} \in L$$

(where $z = a...ab...b$)

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

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

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

$$\Downarrow$$

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

CONTRADICTION!!!

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

Conclusion: L is not a regular language

END OF PROOF