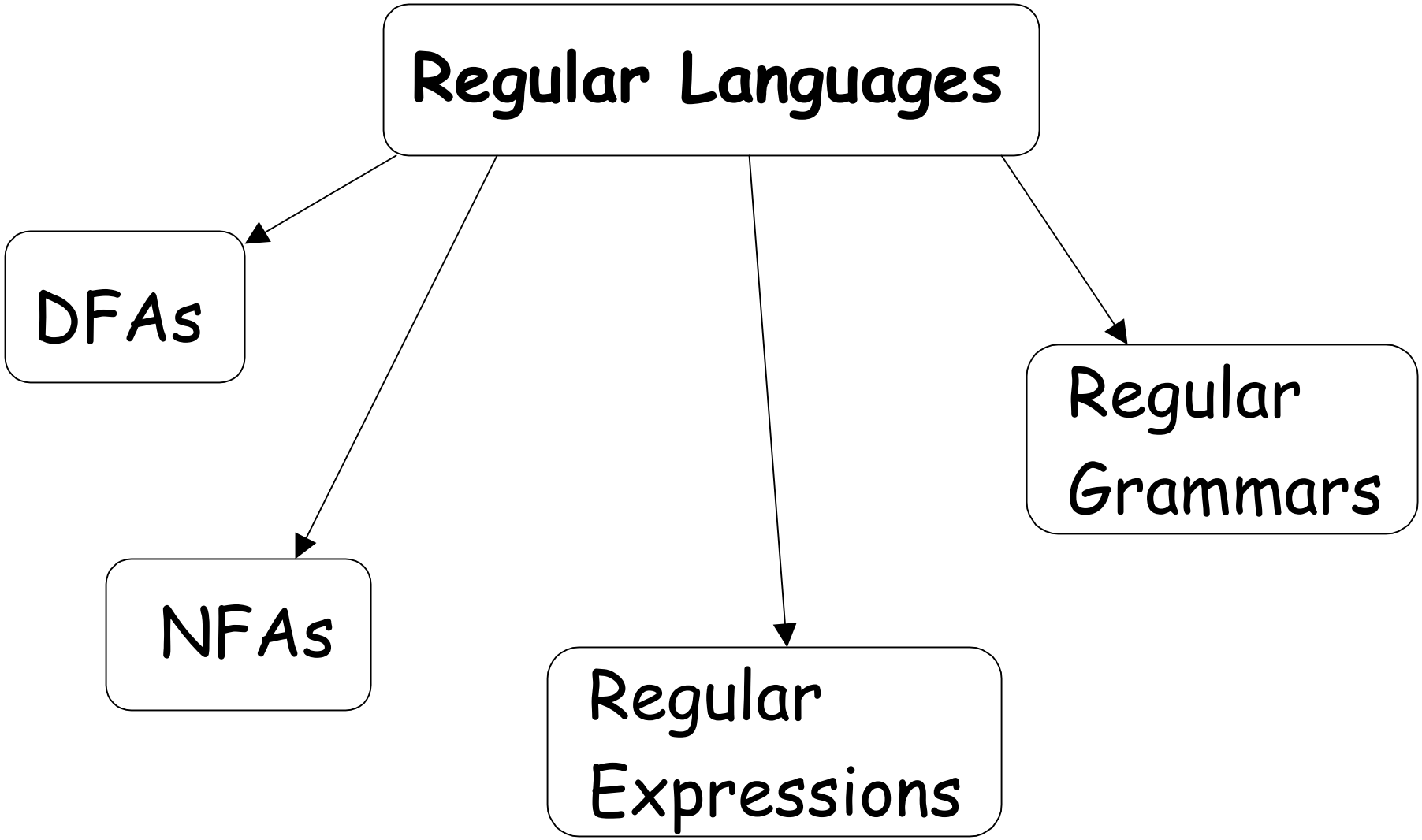
Let's Recapitulate


Regular Languages
DFAs
NFAs
Regular Expressions
Regular Grammars

A standard representation
of a regular language L:

A DFA M that accepts L

A NFA M that accepts L

A regular expression R
that generates L

A regular grammar G
that generates L

When we say:

"We are given a Regular Language L"

We mean:

Language in a standard representation

Elementary Questions

about

Regular Languages

Question:

Given regular language L

how can we check

if a string $w \in L$?

Question:

Given regular language L

how can we check

if a string $w \in L$?

Answer:

Take the DFA that accepts L

and check if w is accepted

Question:

Given regular language L

how can we check

if L is empty, $(L = \emptyset)$

finite,

infinite ?

Answer:

Take the DFA that accepts L

Then check the DFA

If there is a walk from the start state to a final state then:

L is not empty

Otherwise empty

If the walk contains a cycle then:

L is infinite

Otherwise finite

Question:

Given regular languages L_1 and L_2

how can we check if $L_1 = L_2$?

Question:

Given regular languages L_1 and L_2
how can we check if $L_1 = L_2$?

Answer:

take $L = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2)$

And find if $L = \emptyset$

Question:

Given language L

how can we check

if L is not a regular language ?

Question:

Given language L

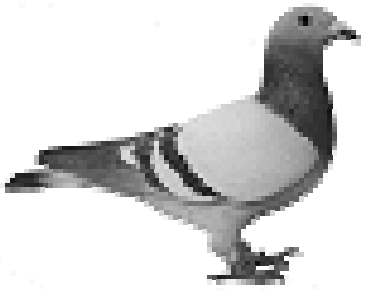
how can we check

if L is not a regular language **?**

Answer:

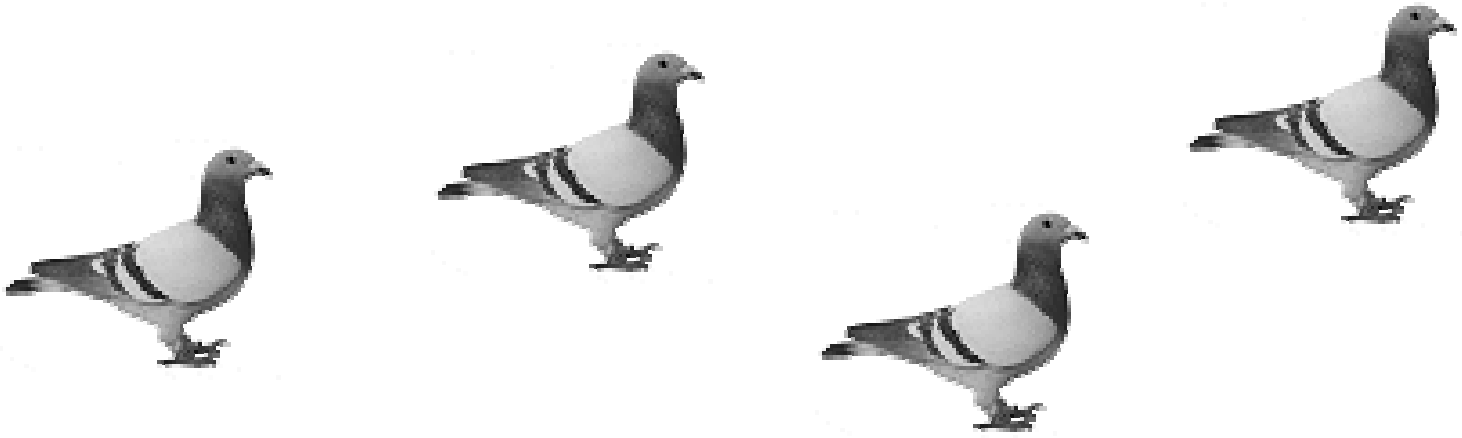
The answer is not obvious

We need the **Pumping Lemma**

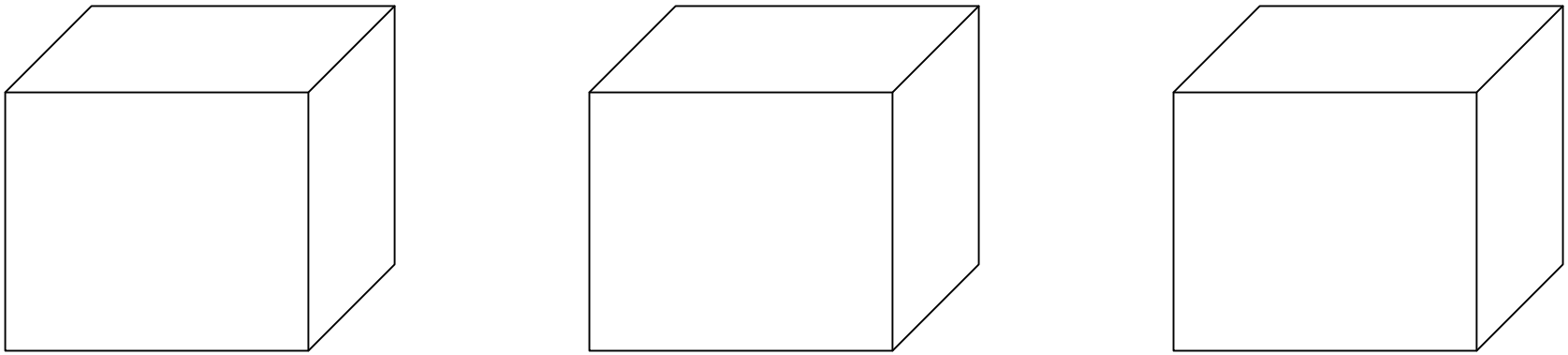


The Pigeonhole Principle

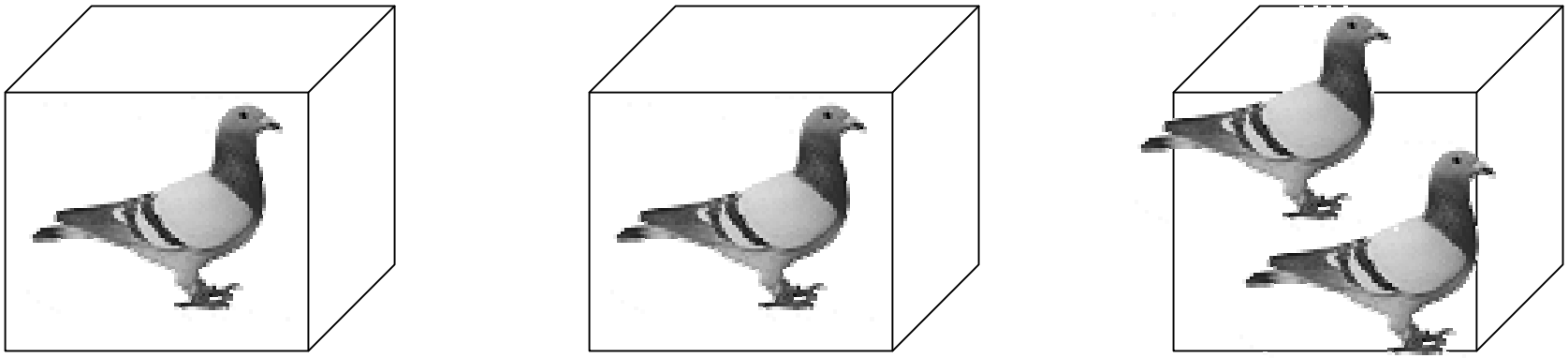
4 pigeons



3 pigeonholes

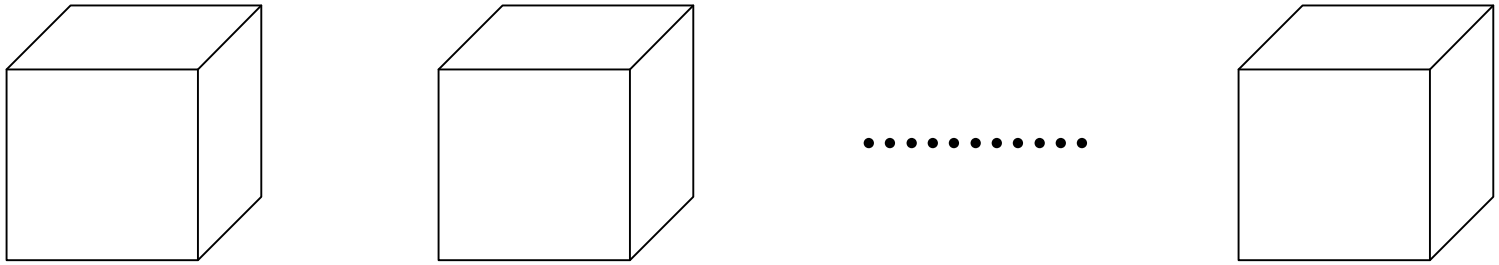


A pigeonhole must have two pigeons



n pigeons $n > m$

m pigeonholes



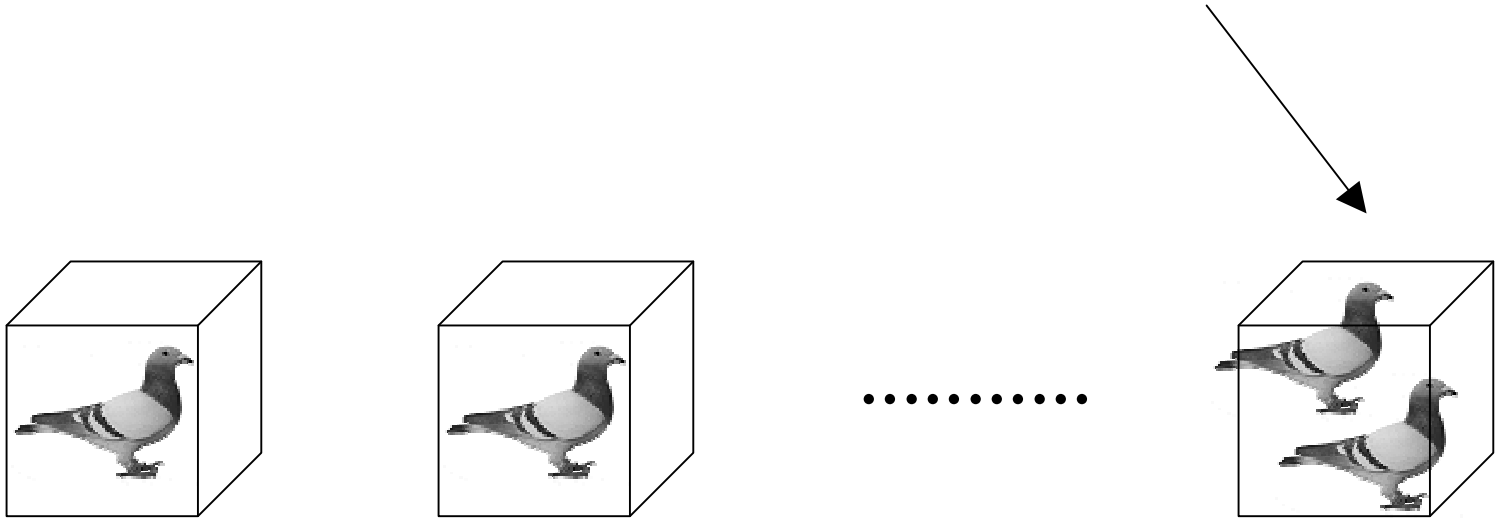
The Pigeonhole Principle

n pigeons

m pigeonholes

$n > m$

There is a pigeonhole with at least 2 pigeons

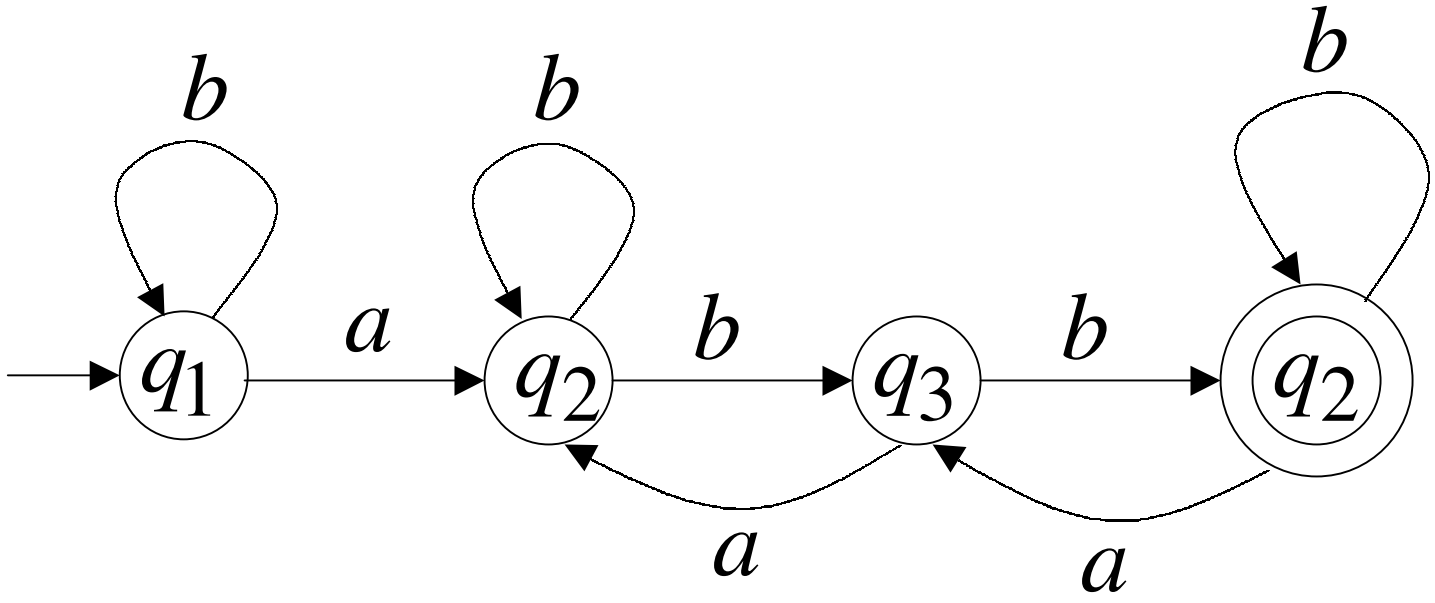


The Pigeonhole Principle

and

DFAs

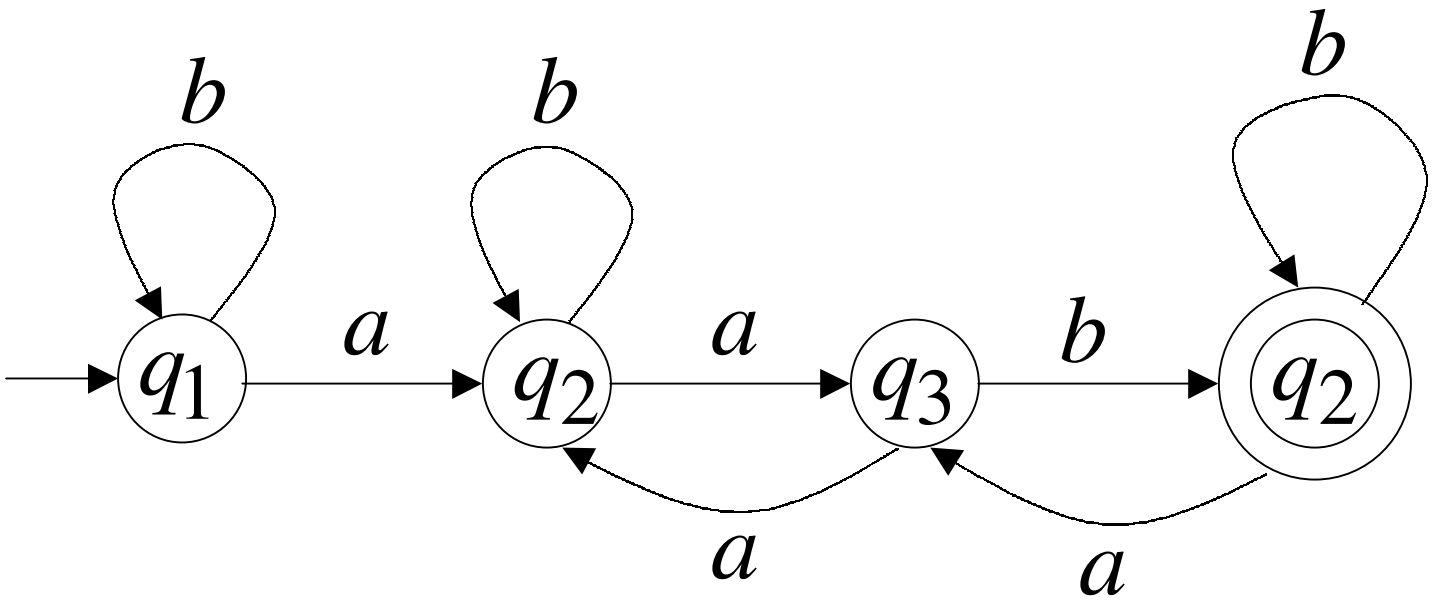
DFA with 4 states



In walks of strings: a no state

aa is repeated

aab



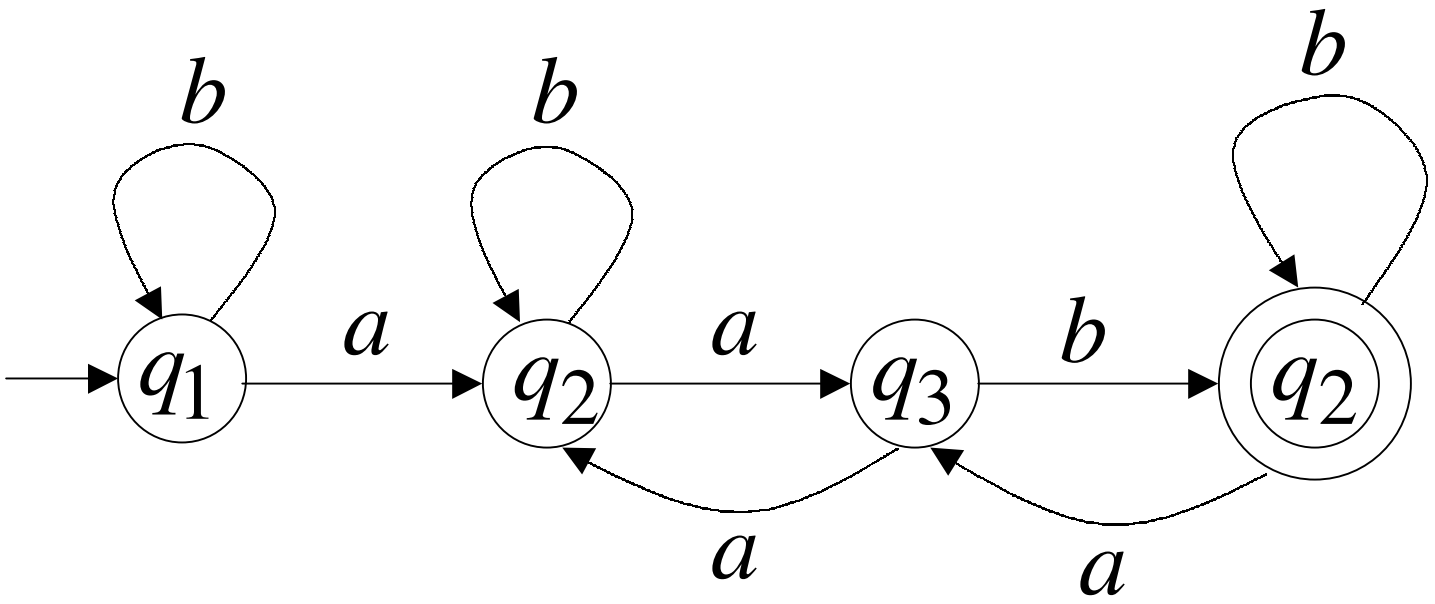
In walks of strings: *aabb*

bbaa

abbabb

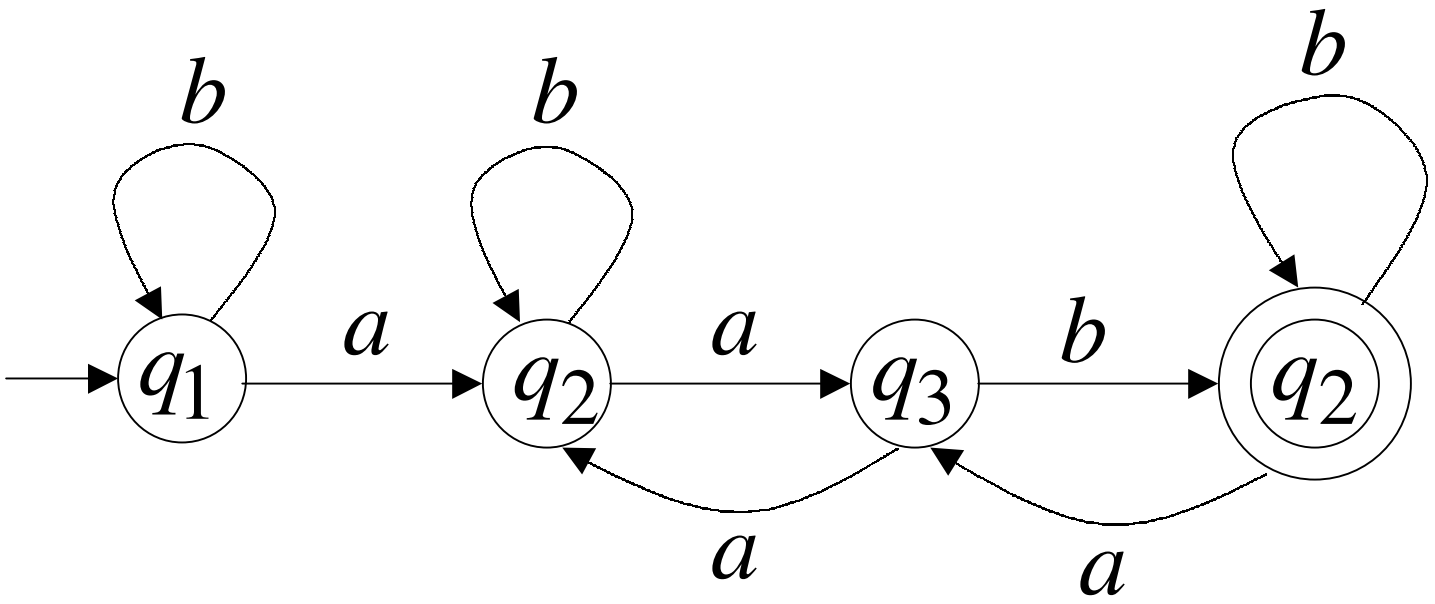
abbbabbbabb...

a state is repeated



If the walk of string w has length $|w| \geq 4$

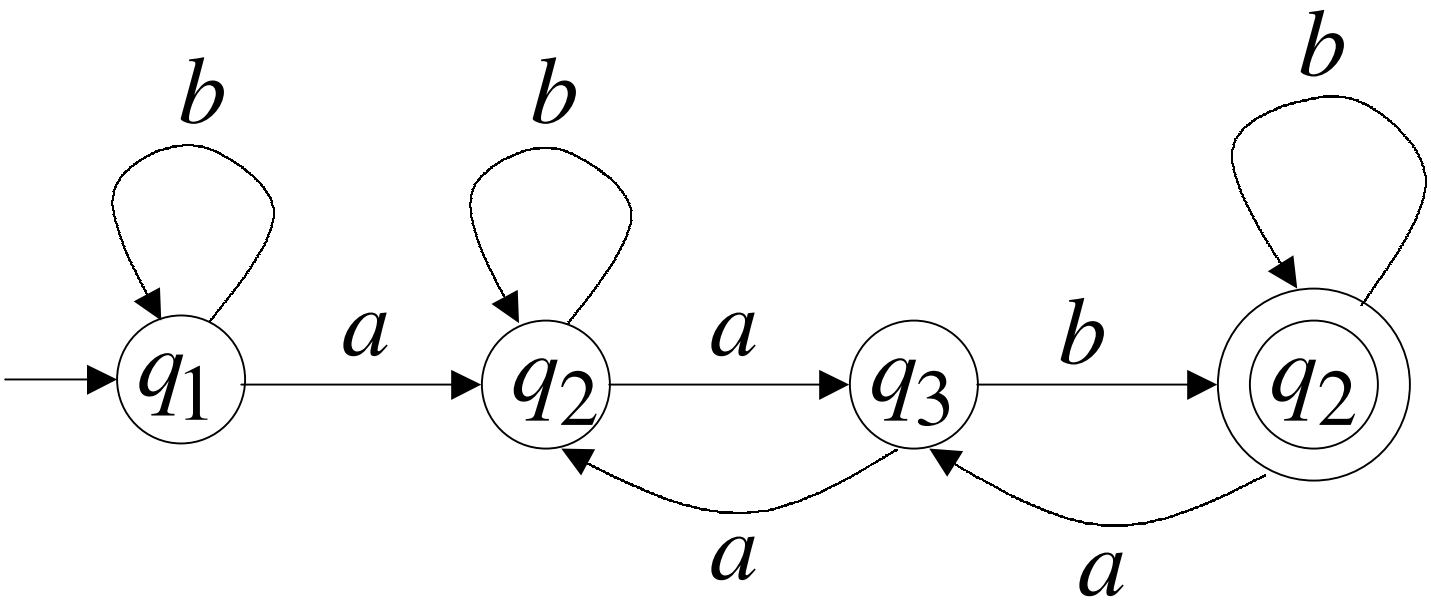
Then a state is repeated



The pigeonhole principle:

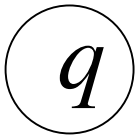
If in a walk: transitions $\geq$ states

Then: A state is repeated

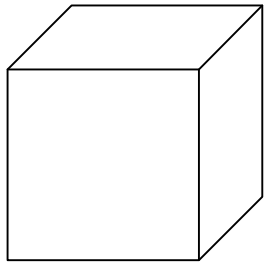


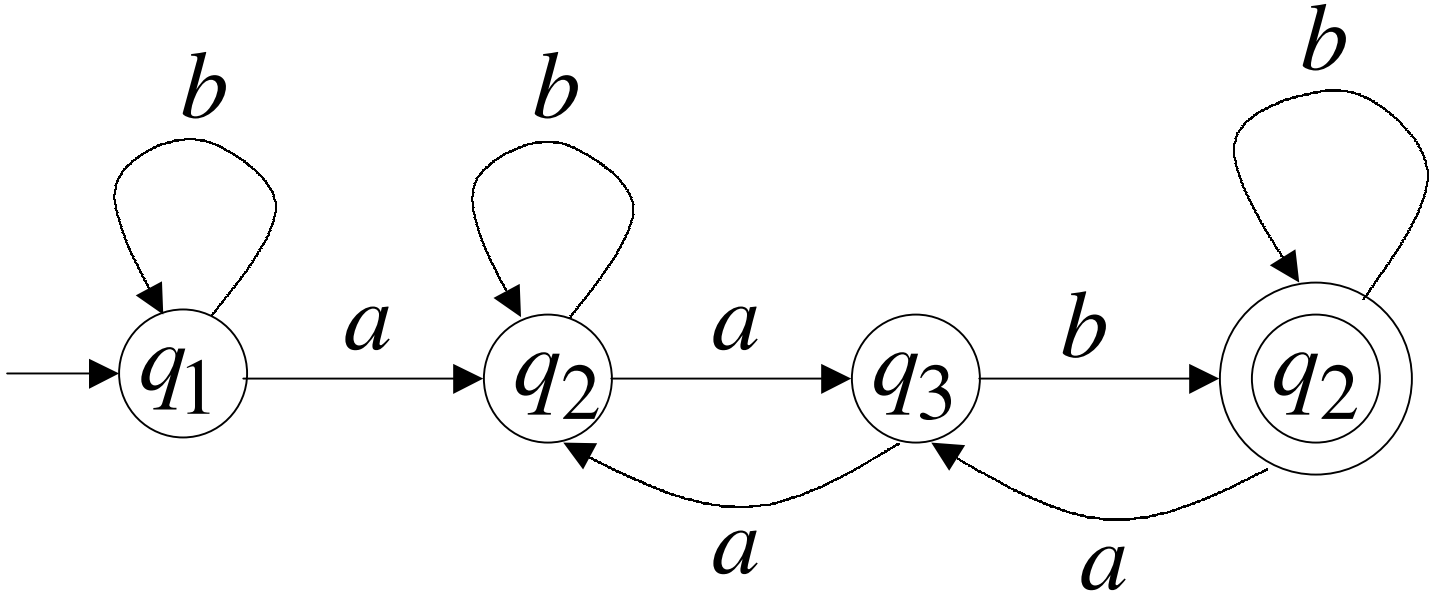
In other words:

transitions are pigeons



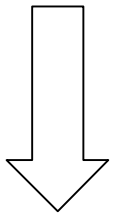
states are pigeonholes



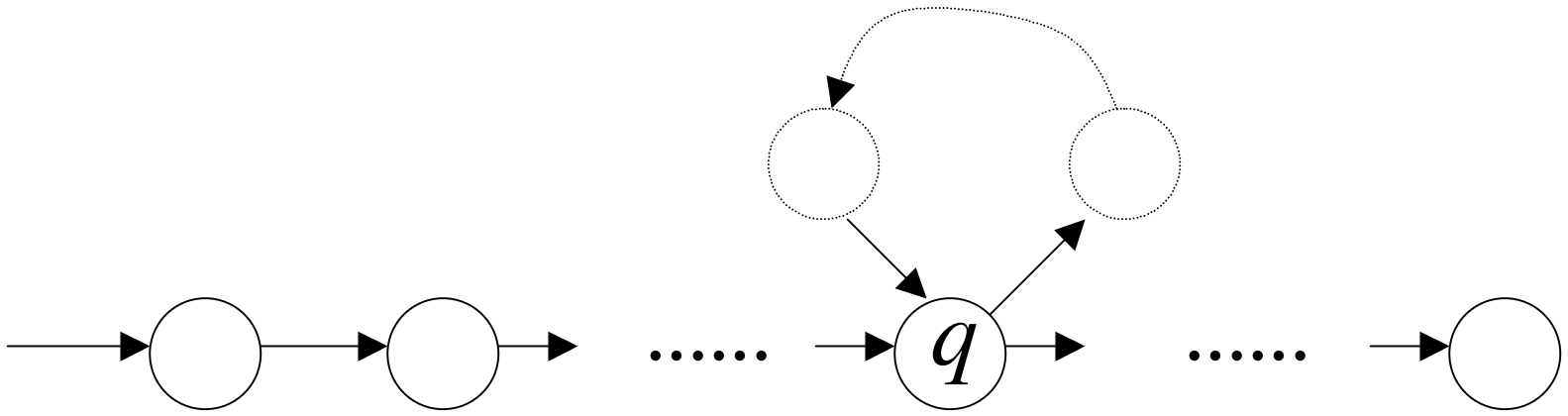


In general:

A string w has length $\geq$ number of states



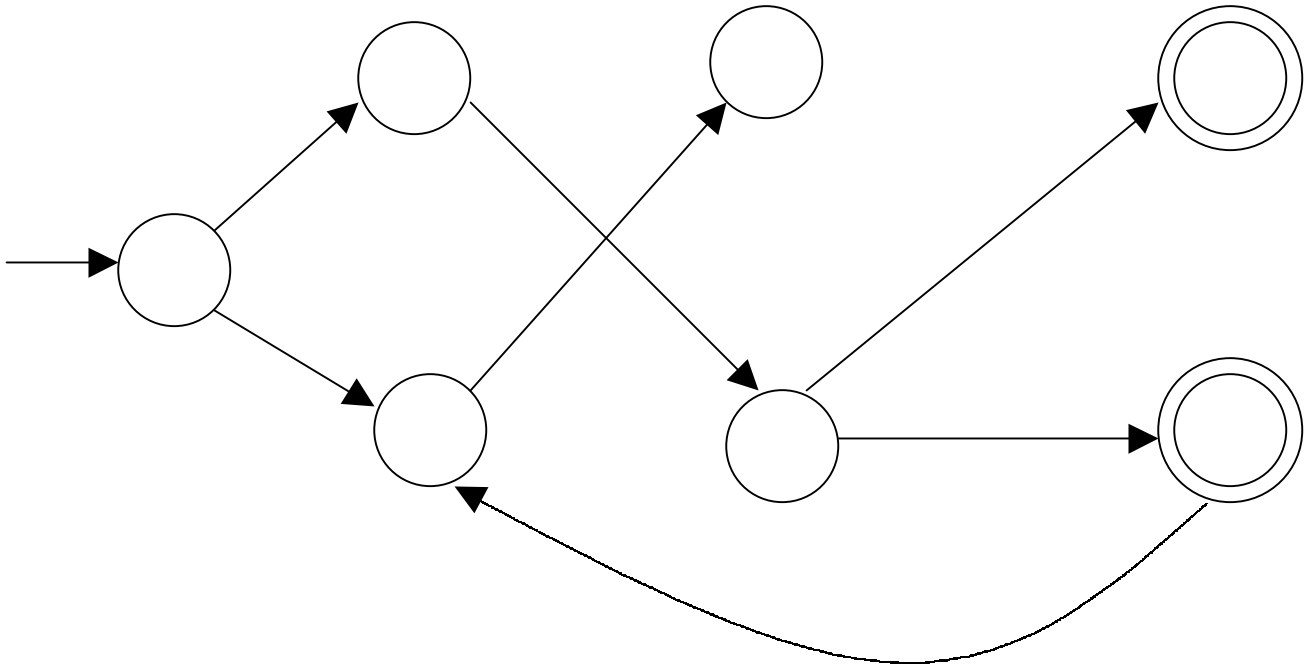
A state q must be repeated in the walk w



The Pumping Lemma

Take an infinite regular language L

DFA that accepts L



m states

Take string w

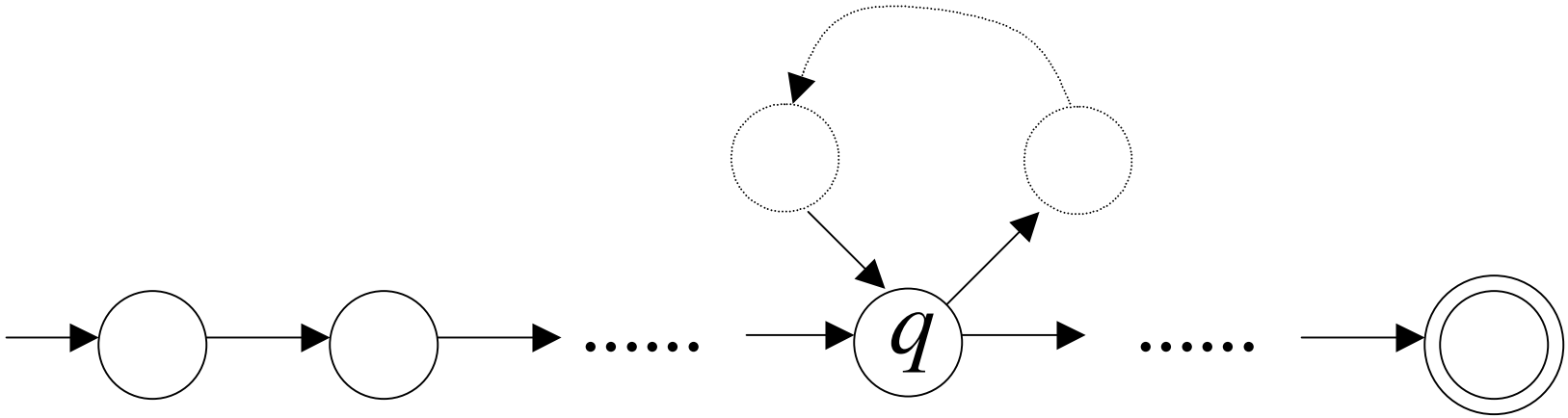
with $w \in L$

There is a walk with label w :

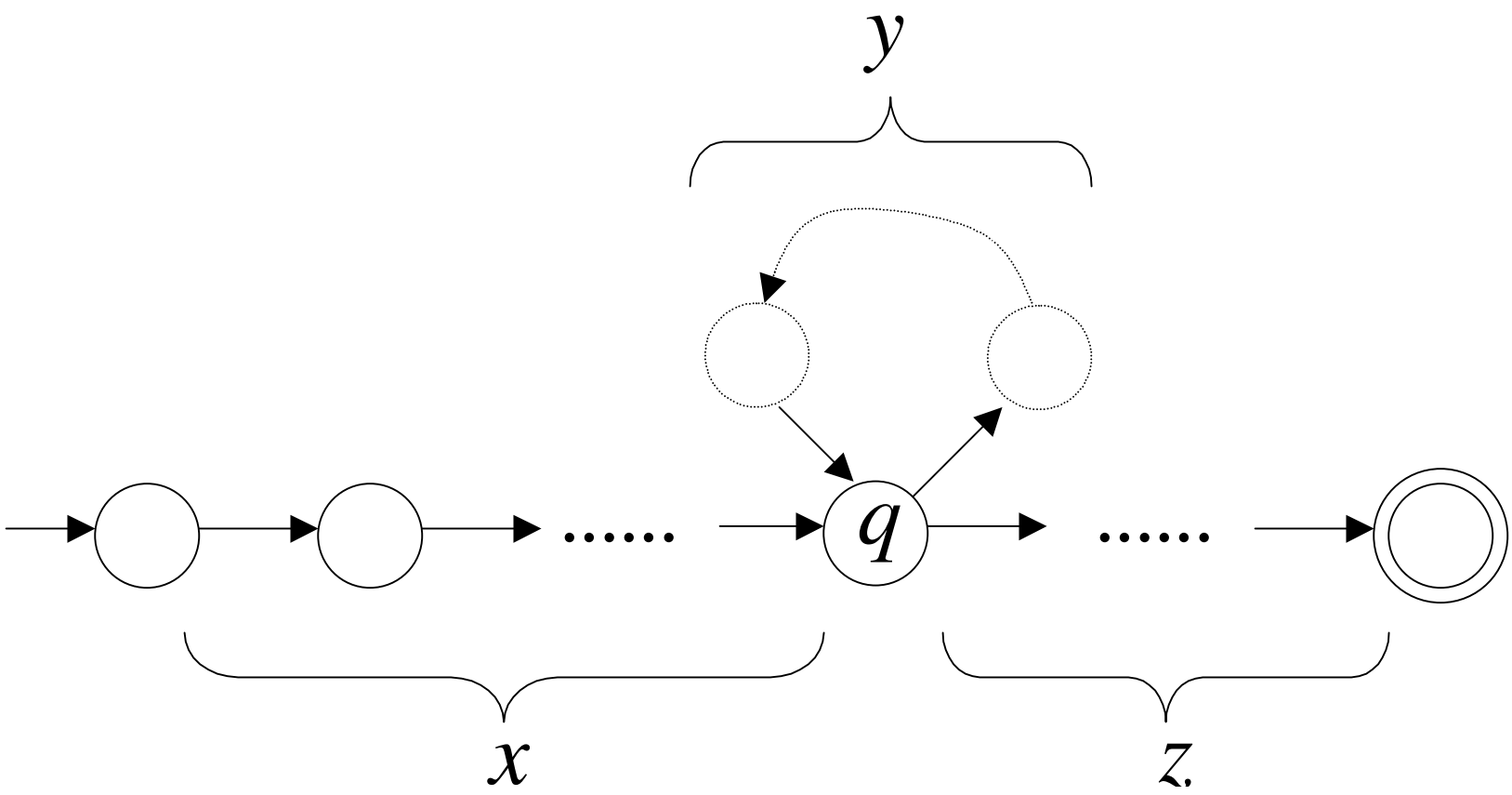
If string w has length $|w| \geq m$ number of states

Then, from the pigeonhole principle:

A state q is repeated in the walk w



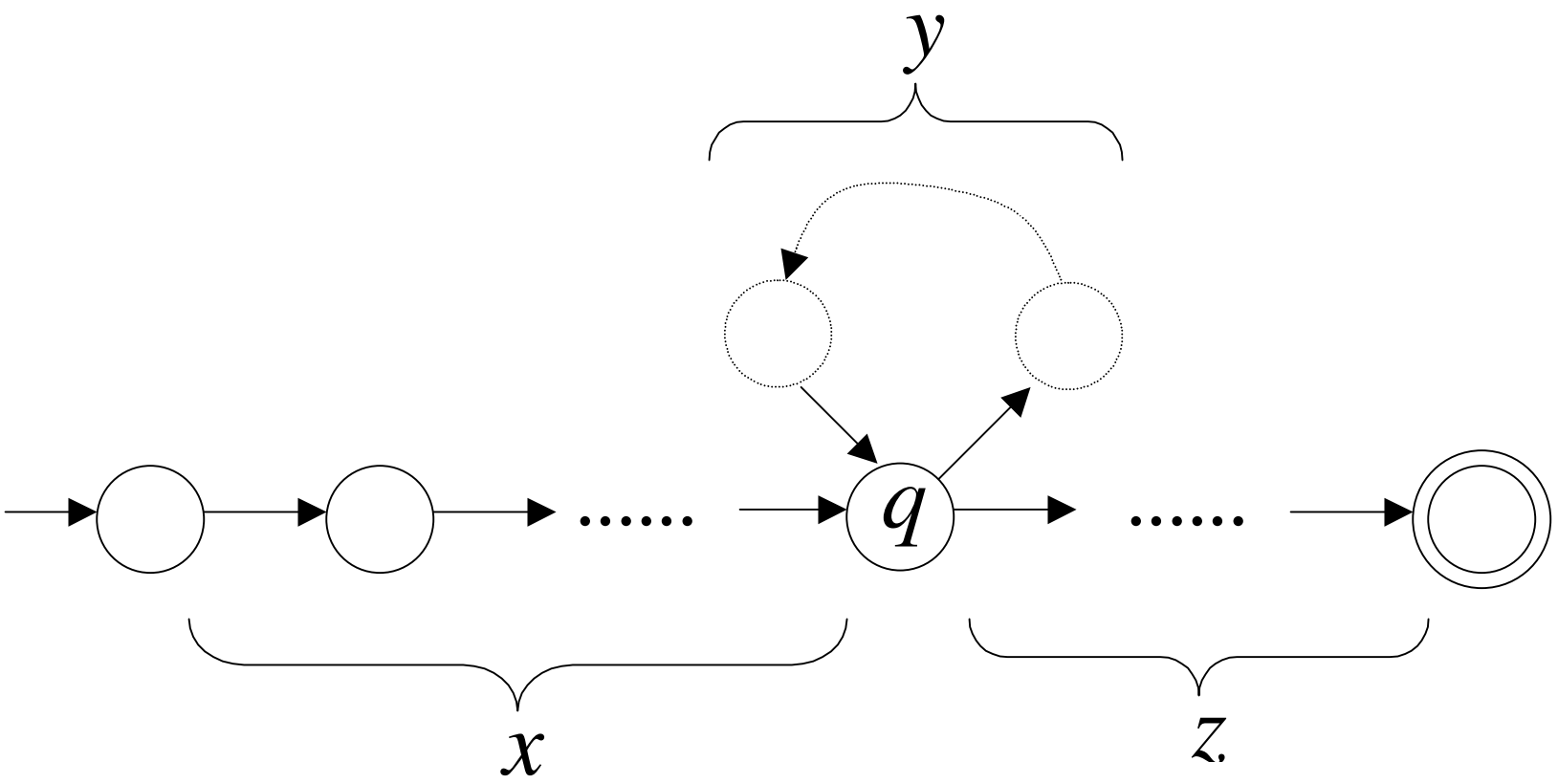
Write $w = x\ y\ z$



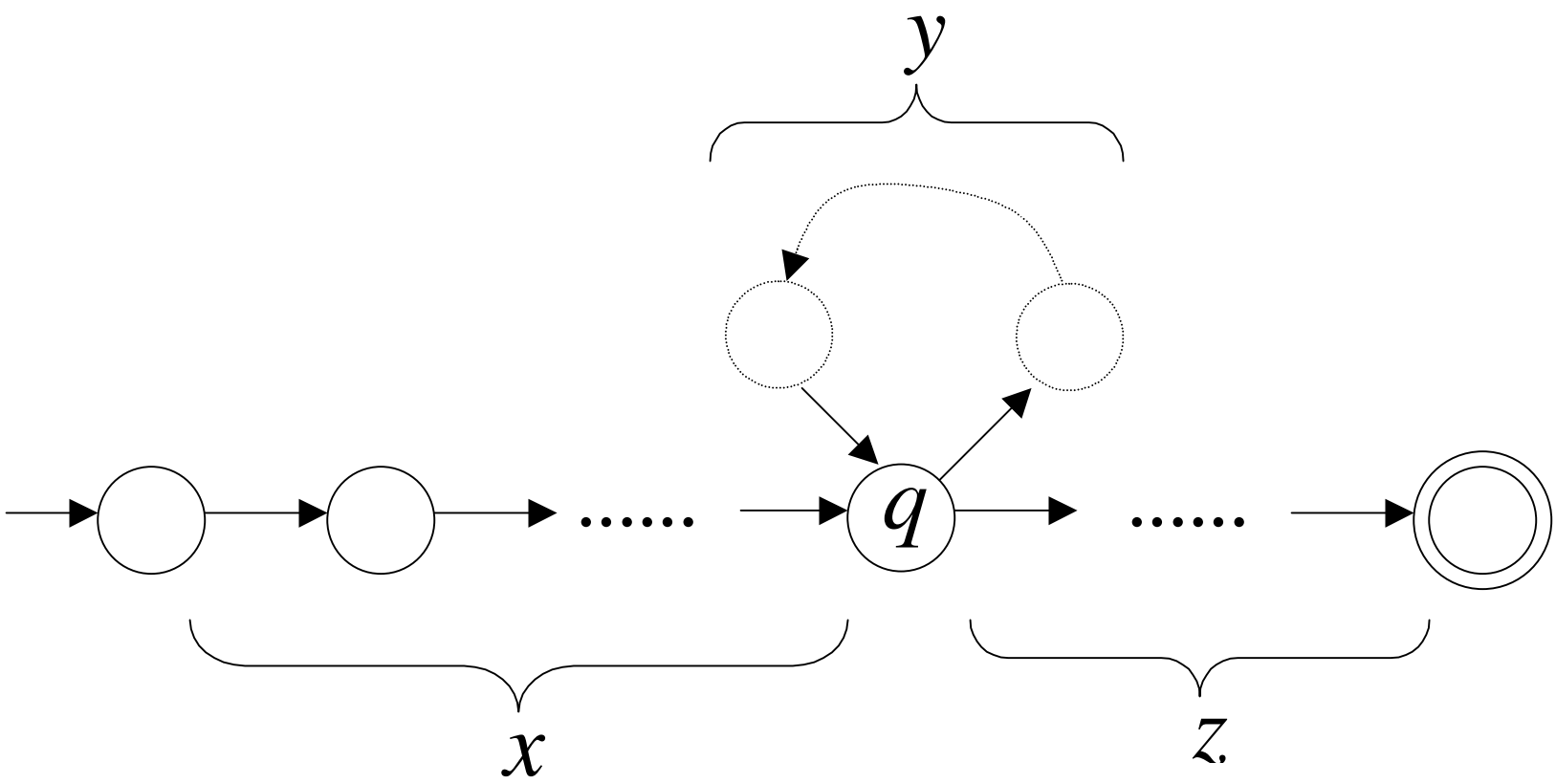
Observations :

length $|x\ y| \leq m$ number of states

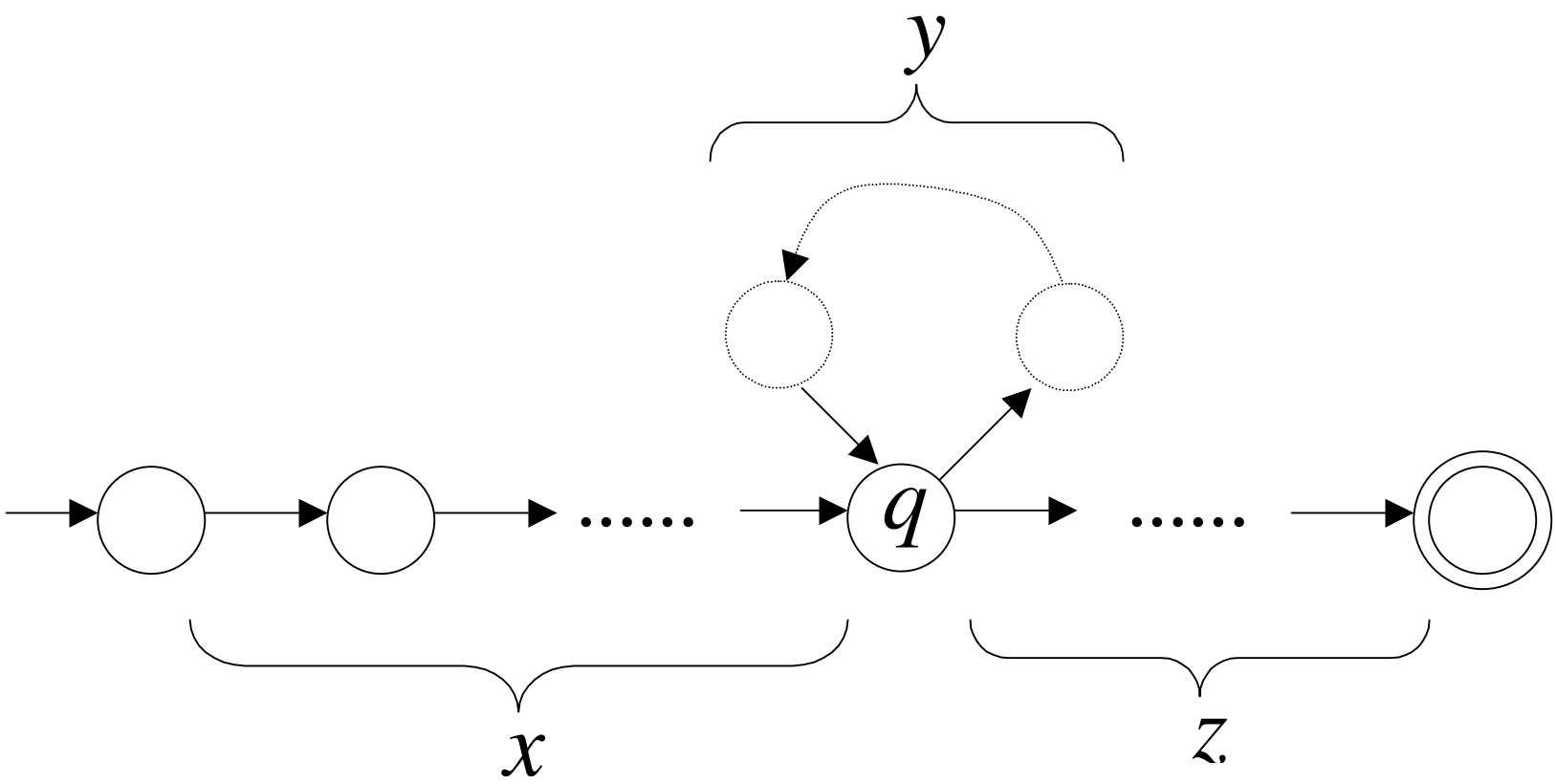
length $|y| \geq 1$



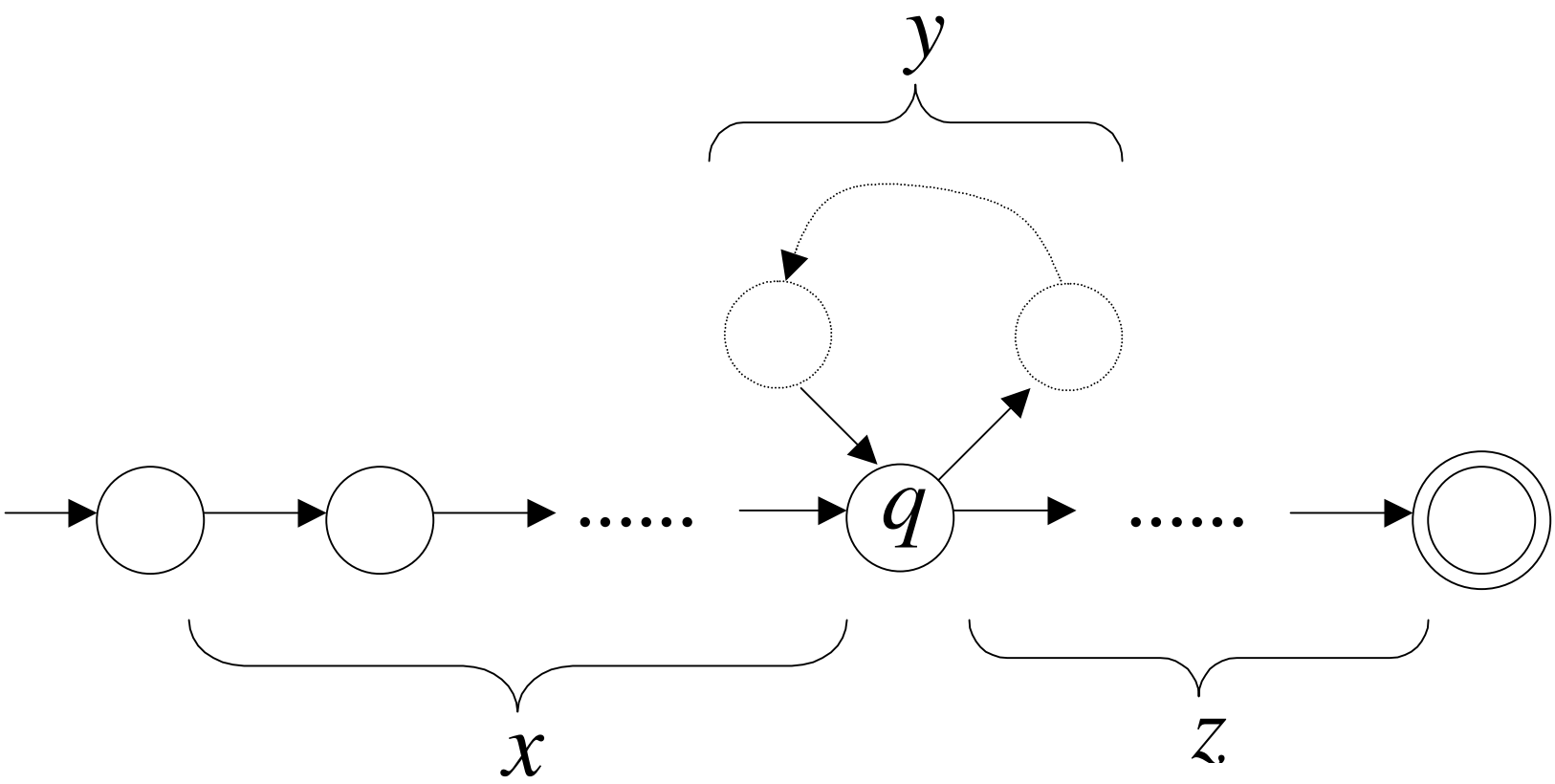
Observation: The string $x\,z$ is accepted



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



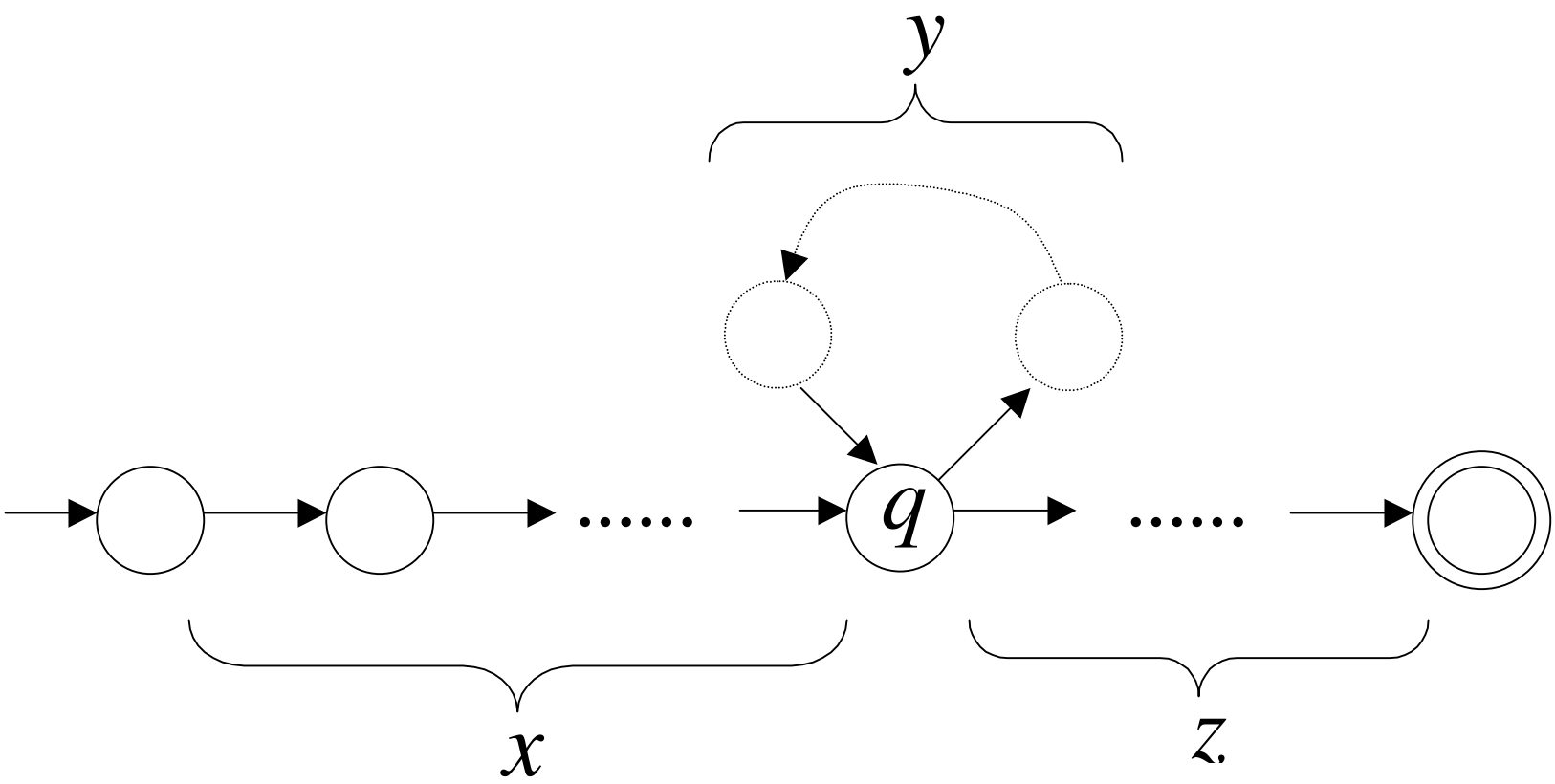
Observation: The string $x\ y\ y\ y\ z$ is accepted



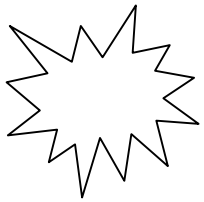
In General:

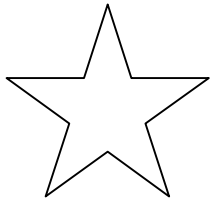
The string $x\ y^i\ z$ is accepted

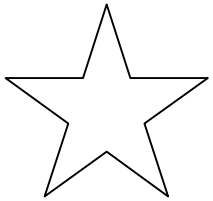
$$i = 0, 1, 2, \ldots$$



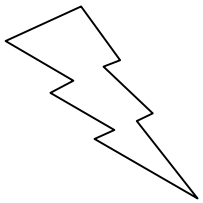
In other words, we described:

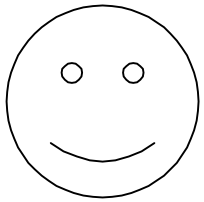


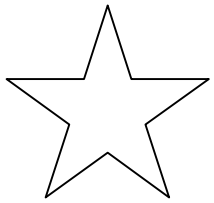




The Pumping Lemma







The Pumping Lemma:

1. Given a infinite regular language L

2. There exists an integer m

3. For any string $w \in L$ with length $|w| \geq m$

4. We can write $w = x\ y\ z$

5. With $|x\ y| \leq m$ and $|y| \geq 1$

6. Such that: string $x\ y^i\ z \in L$

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

Applications of

the Pumping Lemma

Claim:

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

is not regular

Proof:

Use the Pumping Lemma

Proof:

Assume for contradiction
that L is a regular language

Since L is infinite
we can apply the Pumping Lemma

Let m be the integer
in the Pumping Lemma

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

length $|w| \geq m$

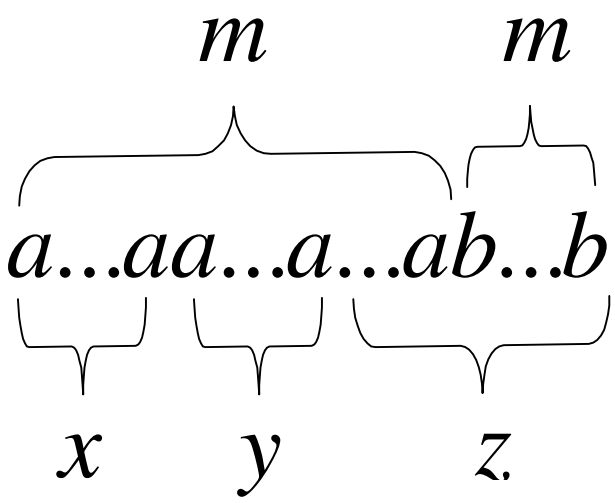
Example: pick $w = a^m b^m$

Write $a^m b^m = x\ y\ z$

From the Pumping Lemma

it must be that length $|x\ y| \leq m$

Therefore:



$y = a^k$

$$x\ y\ z = a^m b^m \qquad\qquad y = a^k$$

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

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

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

$$x\ y^2\ z = x\ y\ y\ z = a^{m+k} b^m\ \in\ L$$

Therefore, $a^{m+k}b^m \in L$

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

and

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

CONTRADICTION!!!

Therefore:

Our assumption that L
is a regular language
cannot be true

CONCLUSION:

L is not a regular language

Claim:

The language $L = \{ww^R : w \in \Sigma^*\}$

is not regular

Proof:

Use the Pumping Lemma

Proof:

Assume for contradiction
that L is a regular language

Since L is infinite
we can apply the Pumping Lemma

Let m be the integer
in the Pumping Lemma

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

length $|w| \geq m$

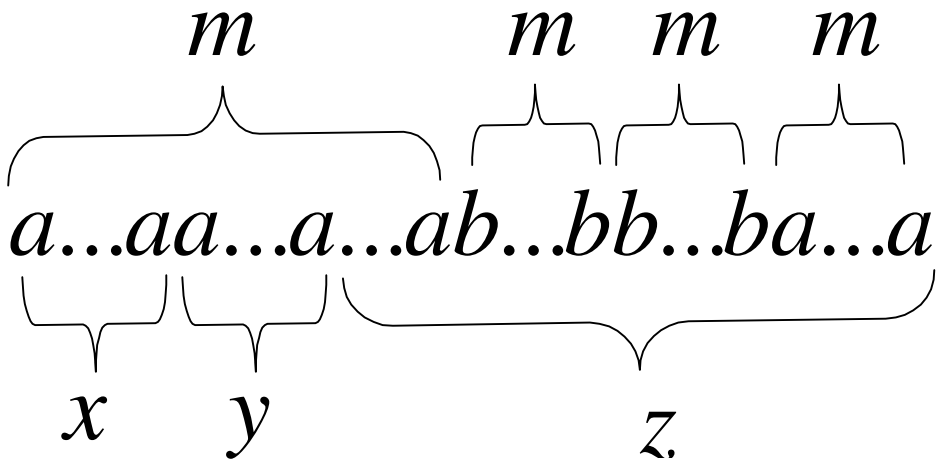
Example: pick $w = a^m b^m b^m a^m$

Write $a^m b^m b^m a^m = x\ y\ z$

From the Pumping Lemma

it must be that length $|x\ y| \leq m$

Therefore:



$y = a^k$

$$x\ y\ z = a^m b^m b^m a^m \qquad y = a^k$$

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

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

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

$$x\ y^2\ z = x\ y\ y\ z = a^{m+k} b^m b^m a^m \in L$$

Therefore, $a^{m+k}b^m b^m a^m \in L$

BUT: $L = \{ww^R : w \in \Sigma^*\}$

and

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

CONTRADICTION!!!

Therefore:

Our assumption that L is a regular language cannot be true

CONCLUSION:

L is not a regular language