# Pumping Lemma

## CSCI 338

If all regular languages have property $\boldsymbol{P}$, and some new language $\boldsymbol{L}$ does not have property $\boldsymbol{P}$, then...?

If all regular languages have property $\boldsymbol{P}$, and some new language $\boldsymbol{L}$ does not have property $\boldsymbol{P}$, then $\boldsymbol{L}$ cannot be a regular language.

# Properties That Imply Language is Regular?

<u>Claim</u>: Languages that are finite in size are regular.

<u>Claim</u>: Languages where all strings have bounded size (each string has size $\leq$ some $n$) are regular.

# Properties That Imply Language is Regular?

Claim: Languages that are finite in size are regular.

Proof: Since there are a finite number of strings, build a DFA for each individual string in the language.

Then connect all start states to a new start state via $\varepsilon$-transitions.

Claim: Languages where all strings have bounded size (each string has size $\leq$ some $n$) are regular.

# Properties That Imply Language is Regular?

Claim: Languages that are finite in size are regular.

Proof: Since there are a finite number of strings, build a DFA for each individual string in the language.

Then connect all start states to a new start state via $\varepsilon$-transitions.

Claim: Languages where all strings have bounded size (each string has size $\leq$ some $n$) are regular.

Proof: Since the alphabet is finite, there is a finite number of strings constructible with $n$ characters.

Thus, the language is finite and regular.

# Quest for Regular Language Properties

What do we know about non-regular languages?

- They must be infinite in size.
- They must have arbitrarily long strings in them.

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

**?**

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.

Let $p =$ number of states.
Let $s$ be <u>any</u> string in $L$ such that $|s| \geq p$.
Then…

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.

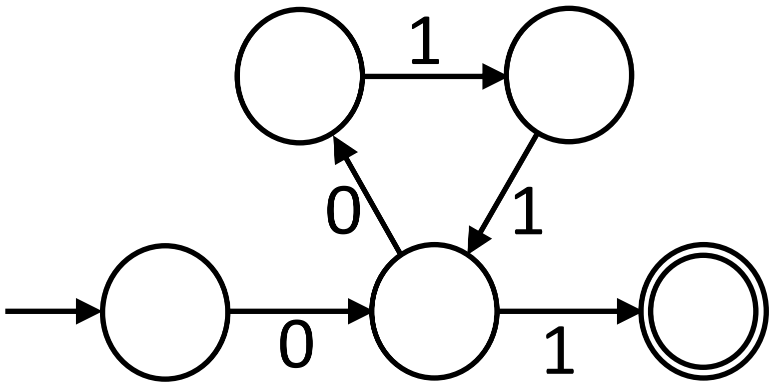
Let $p =$ number of states.
Let $s$ be <u>any</u> string in $L$ such that $|s| \geq p$.
Then, $s$ must visit repeated states (i.e. loops).

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.

Let $p =$ number of states.
Let $s$ be <u>any</u> string in $L$ such that $|s| \geq p$.
Then, $s$ must visit repeated states (i.e. loops).

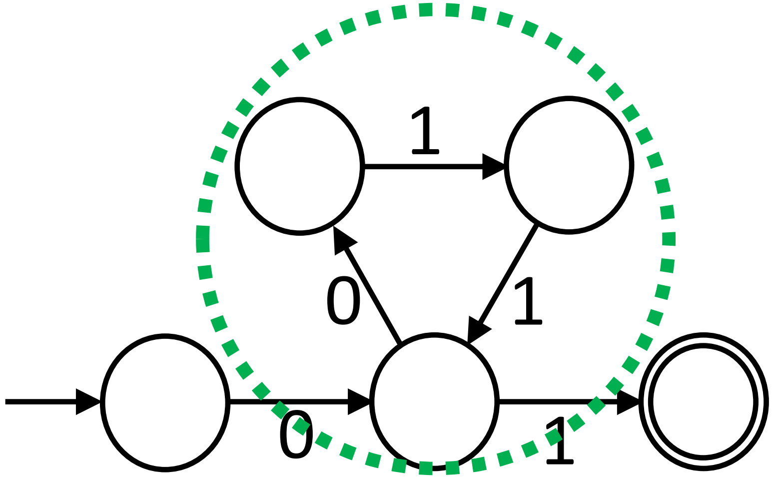



e.g. $s = 00111 \in L$

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.



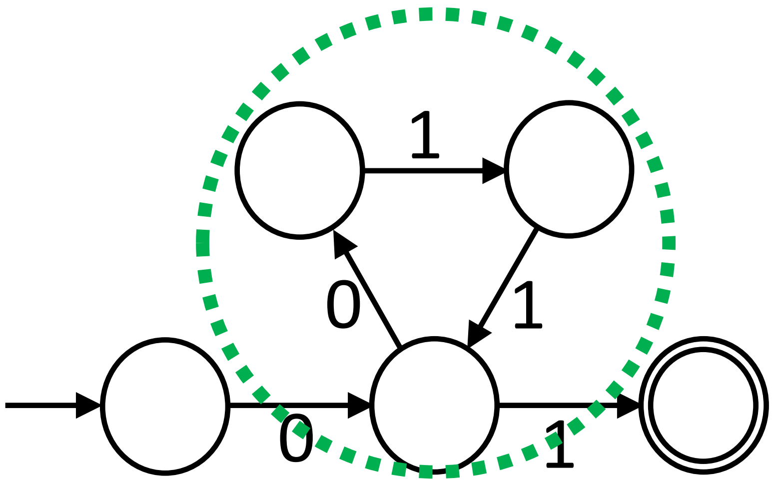

Let $p =$ number of states.
Let $s$ be <u>any</u> string in $L$ such that $|s| \geq p$.
Then, $s$ must visit repeated states (i.e. loops).

e.g. $s = 0|011|1 \in L$

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.




Let $p =$ number of states.
Let $s$ be <u>any</u> string in $L$ such that $|s| \geq p$.
Then, $s$ must visit repeated states (i.e. loops).

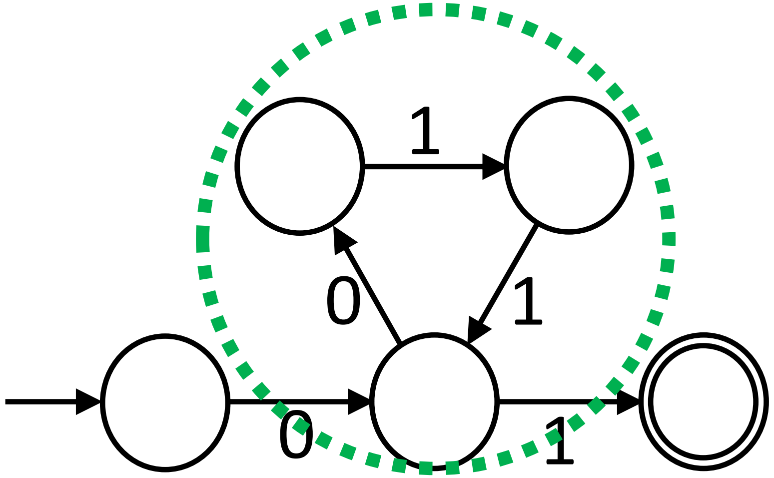
e.g. $s = 0|011|1 \in L$

Is $s = 0|011|011|1 \in L$?

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.




Let $p =$ number of states.
Let $s$ be <u>any</u> string in $L$ such that $|s| \geq p$.
Then, $s$ must visit repeated states (i.e. loops).

e.g. $s = 0|011|1 \in L$

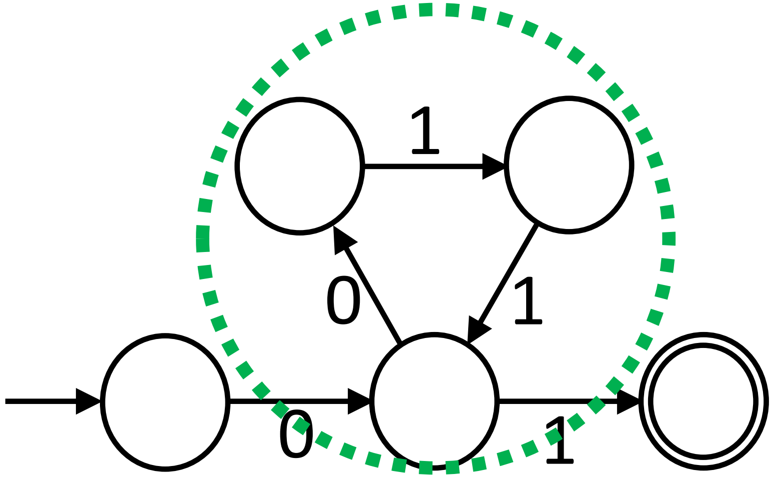
Is $s = 0|011|011|1 \in L$?

What about $s = 0|011|011|011|1 \in L$?

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.




Let $p =$ number of states.
Let $s$ be <u>any</u> string in $L$ such that $|s| \geq p$.
Then, $s$ must visit repeated states (i.e. loops).

e.g. $s = 0|011|1 \in L$

Is $s = 0|011|011|1 \in L$?

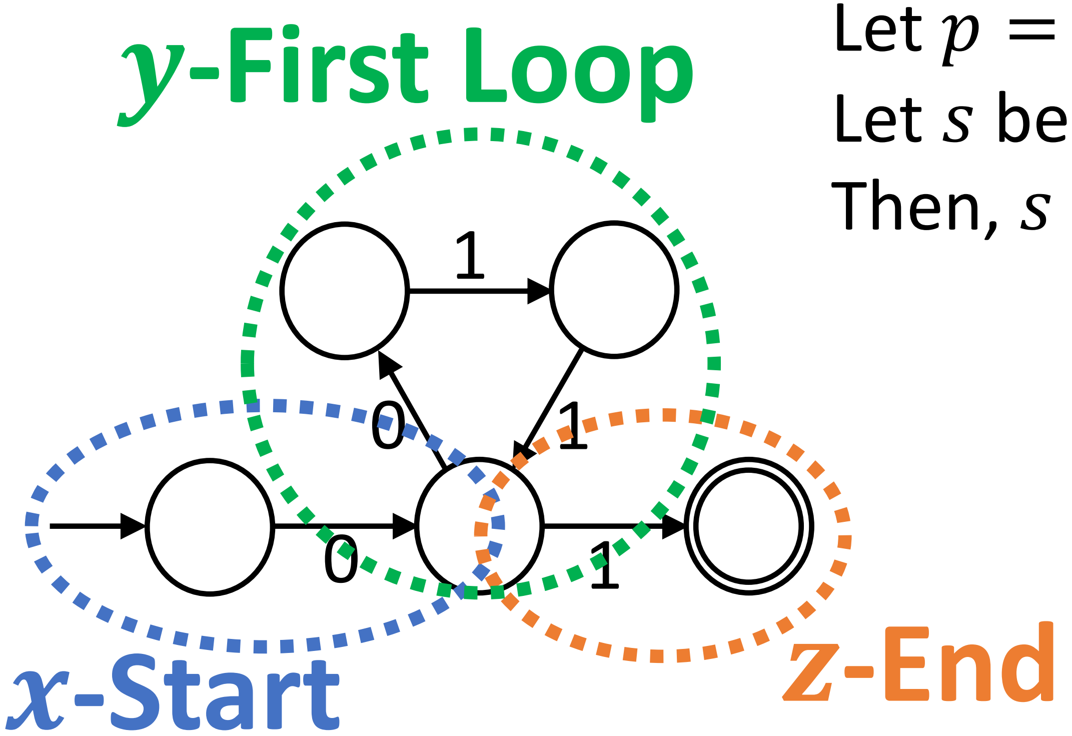
What about $s = 0|011|011|011|1 \in L$?

What about $s = 0|1 \in L$?

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.




Let $p =$ number of states.
Let $s$ be <u>any</u> string in $L$ such that $|s| \geq p$.
Then, $s$ must visit repeated states (i.e. loops).

e.g. $s = \overset{x}{0}|\overset{y}{011}|\overset{z}{1} \in L$

Is $s = 0|011|011|1 \in L$?

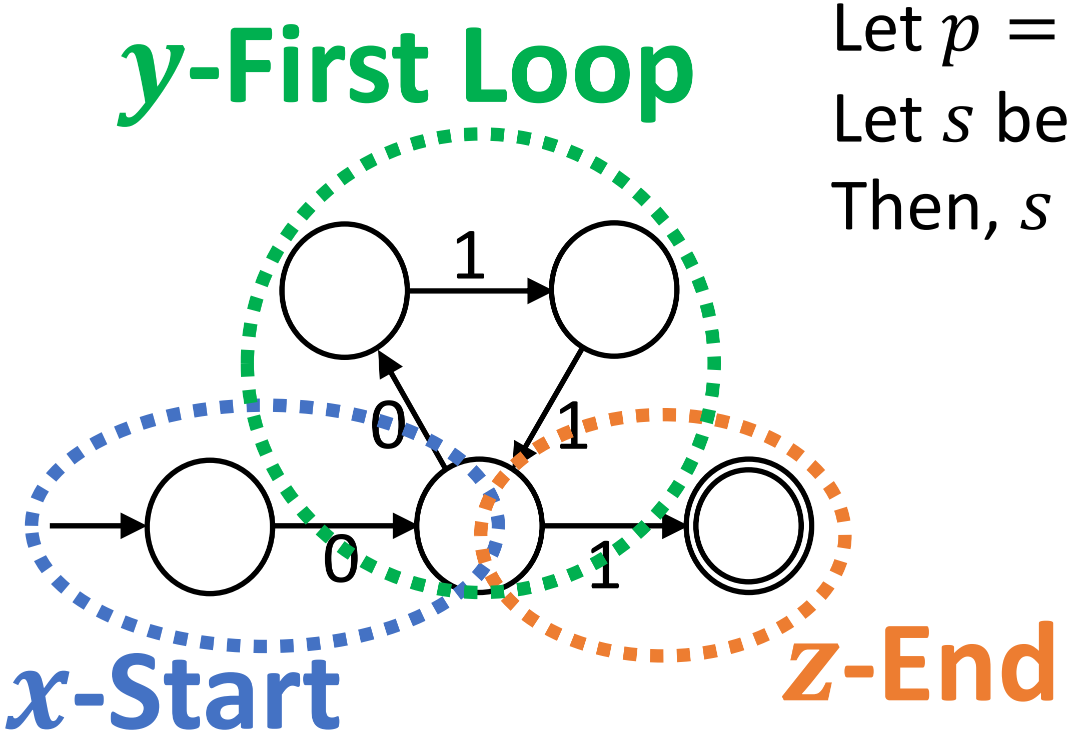
What about $s = 0|011|011|011|1 \in L$?

What about $s = 0|1 \in L$?

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.




Let $p =$ number of states.

Let $s$ be <u>any</u> string in $L$ such that $|s| \geq p$.

Then, $s$ must visit repeated states (i.e. loops).

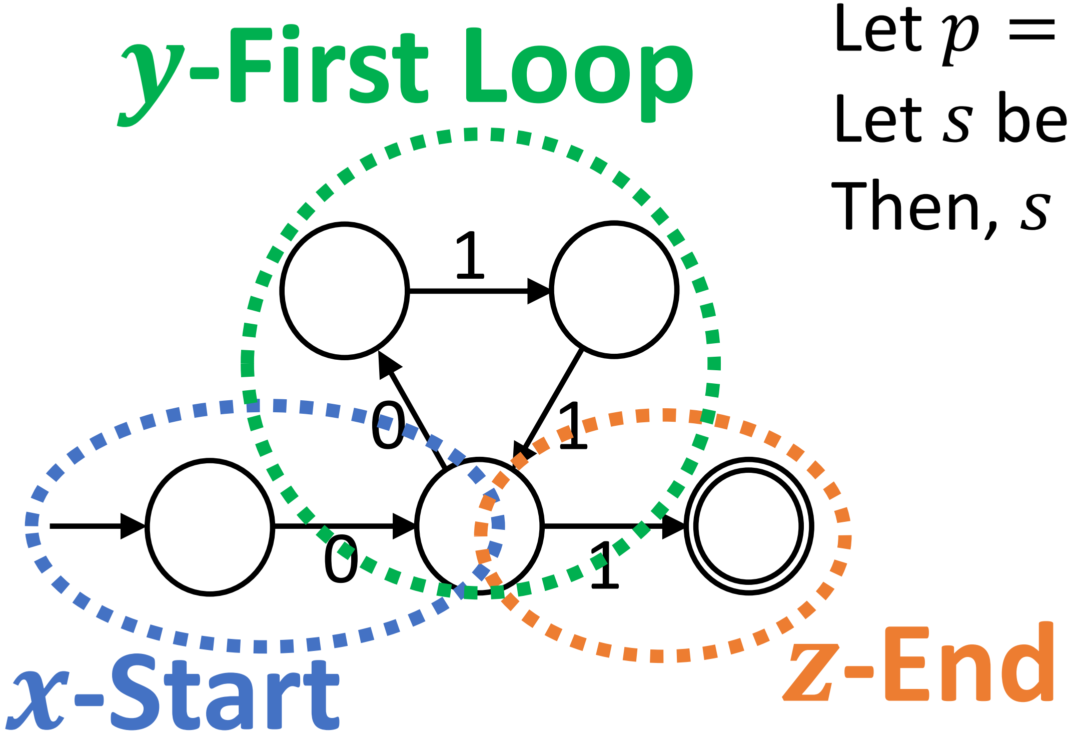
$$\text{e.g. } s = \overset{x}{0}|\overset{y}{011}|\overset{z}{1} \in L$$

$$\text{What about } s = \overset{x}{0}|\overset{y}{011}|\overset{y}{011}|\overset{y}{011}|\overset{z}{1} \in L?$$

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.




Let $p =$ number of states.

Let $s$ be <u>any</u> string in $L$ such that $|s| \geq p$.

Then, $s$ must visit repeated states (i.e. loops).

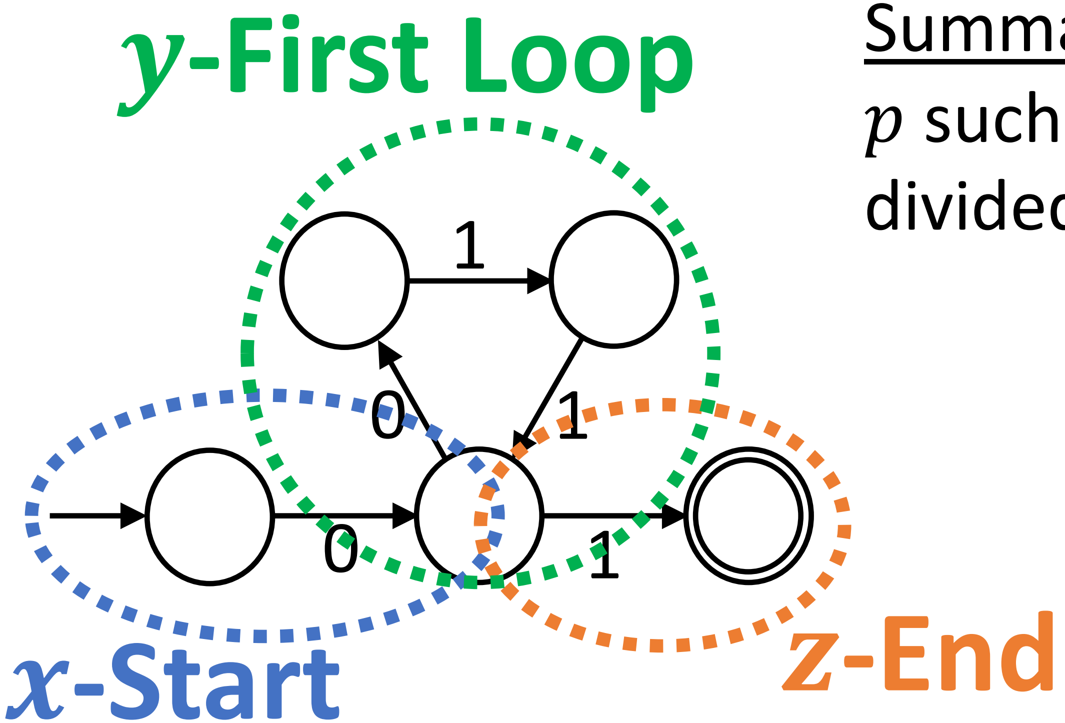
e.g. $s = \overset{x}{0}|\overset{y}{011}|\overset{z}{1} \in L$

What about $s = \overset{x}{0}|\overset{z}{1} \in L$?

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. ∃ strings longer than the number of states.

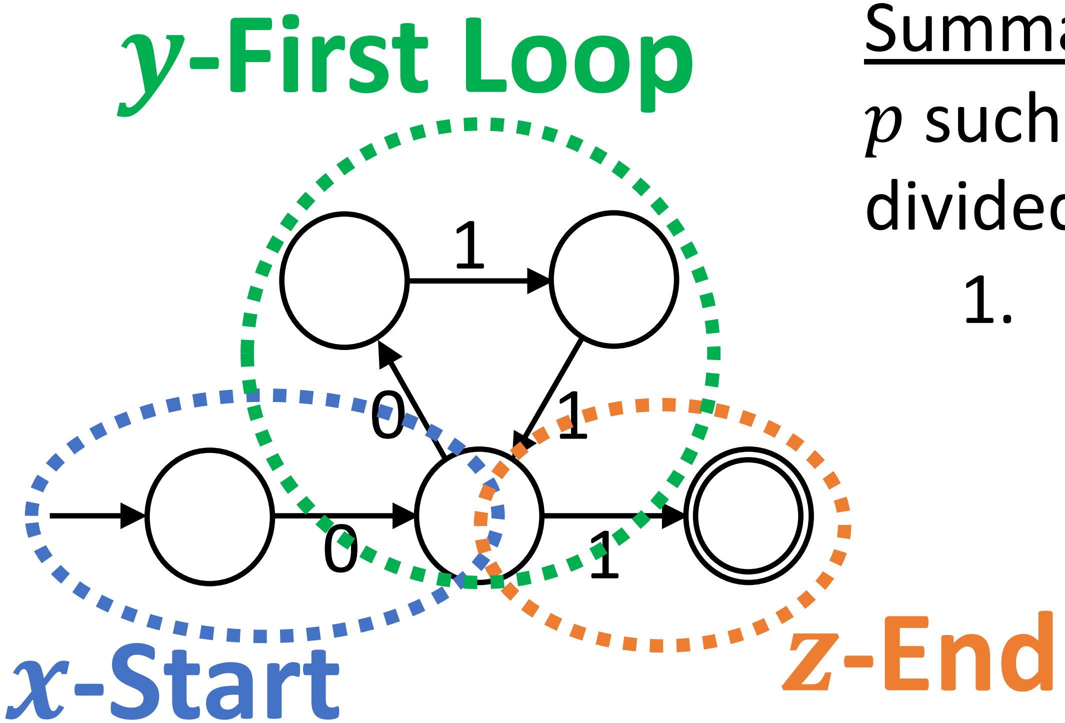



Summary: Given a regular language $L$, ∃ a number $p$ such that any string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.




Summary: Given a regular language $L$, $\exists$ a number $p$ such that any string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

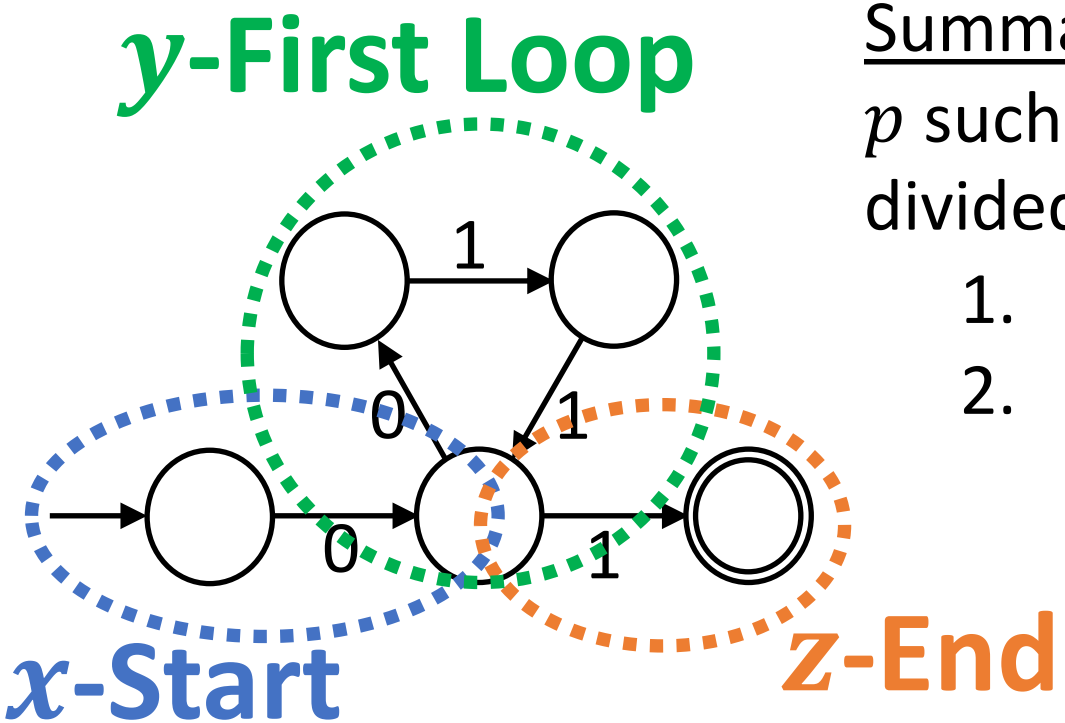
1. $xy^iz \in L, \forall i \geq 0$.

From our previous argument.

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.




Summary: Given a regular language $L$, $\exists$ a number $p$ such that any string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

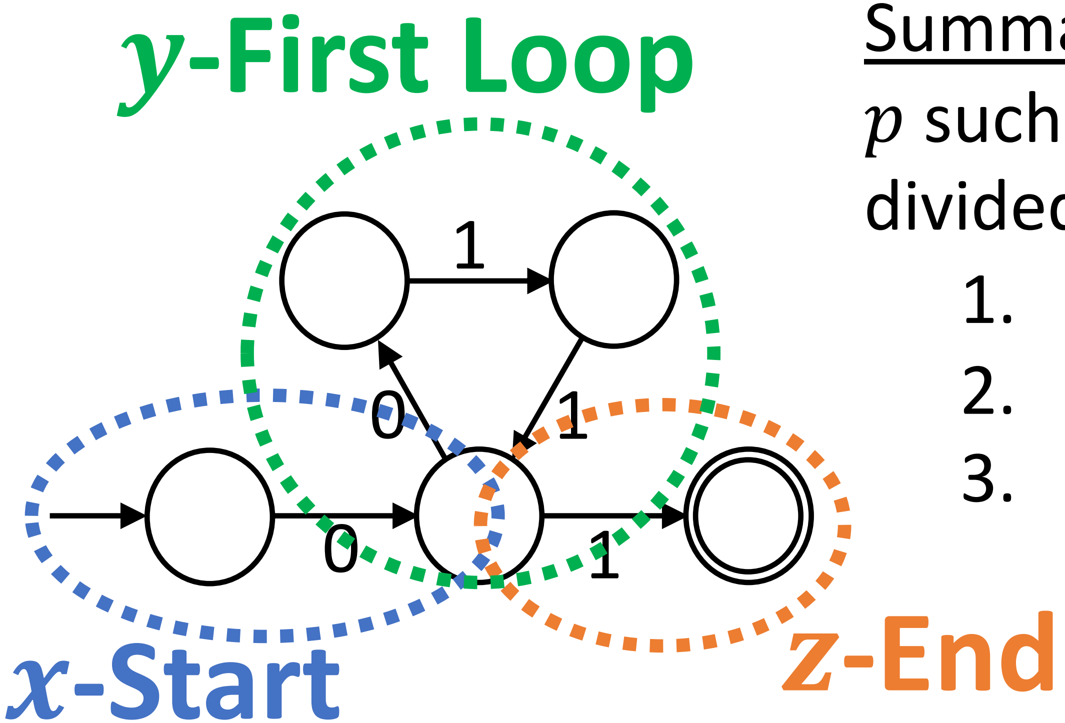
1. $xy^iz \in L, \forall i \geq 0.$
2. $|y| > 0.$

Since $|s| \geq p$, we must have repeated states.

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.




Summary: Given a regular language $L$, $\exists$ a number $p$ such that any string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

1. $xy^iz \in L, \forall i \geq 0$.
2. $|y| > 0$.
3. $|xy| \leq p$.

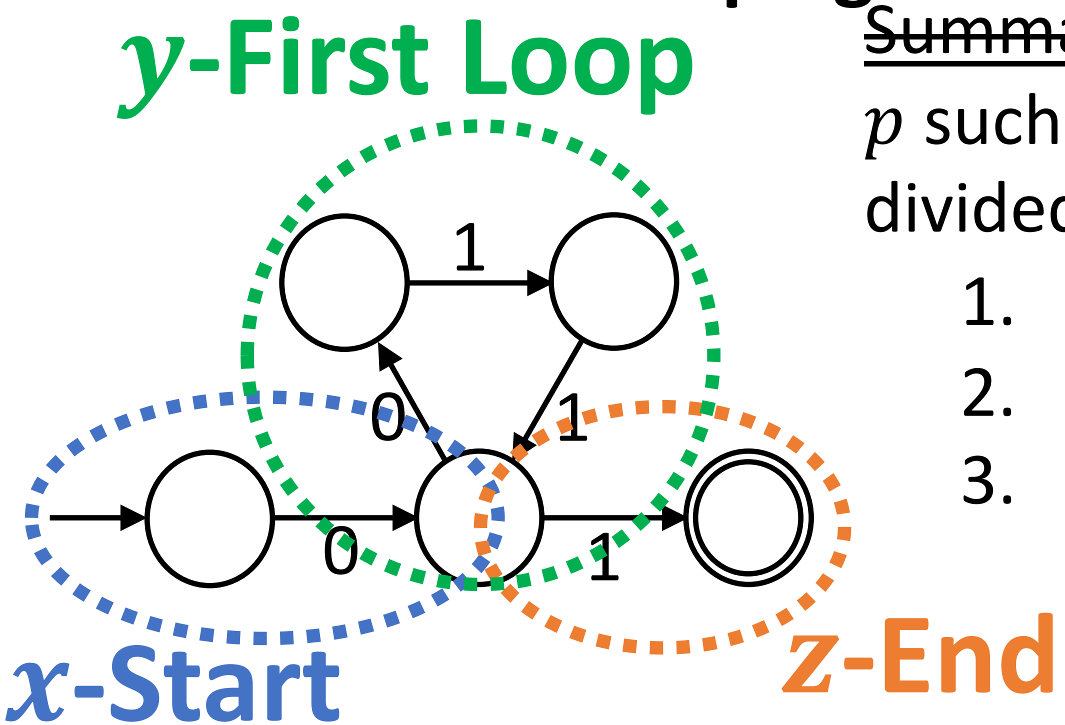
Since there have to be repeated states within the first $p$ transitions.

# Quest for Regular Language Properties

Suppose some regular language $L$ contains strings of arbitrary length (i.e. $\forall n \geq 0, \exists s \in L$ such that $|s| \geq n$). Then:

1. We know the DFA for that language has a finite number of states.
2. $\exists$ strings longer than the number of states.

**Pumping Lemma**




~~Summary~~: Given a regular language $L$, $\exists$ a number $p$ such that any string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

1. $xy^iz \in L, \forall i \geq 0.$
2. $|y| > 0.$
3. $|xy| \leq p.$

# Quest for Regular Language Properties

Suppos … gth
(i.e. $\forall n$ …

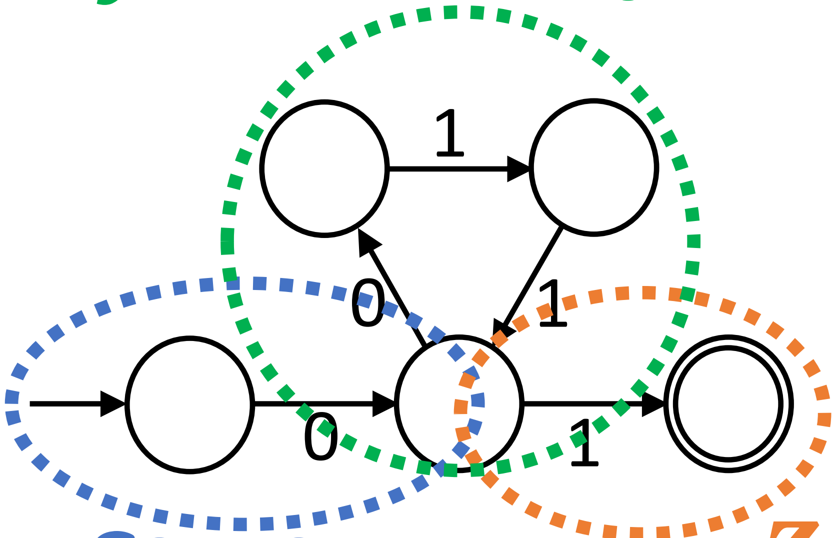
1. W … .
2. $\exists$ strings longer than the number of states.

**The Pumping Lemma is our property that all regular languages must have.**

**Pumping Lemma**

~~Summary~~: Given a regular language $L$, $\exists$ a number $p$ such that <u>any</u> string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

1. $xy^iz \in L, \forall i \geq 0.$
2. $|y| > 0.$
3. $|xy| \leq p.$

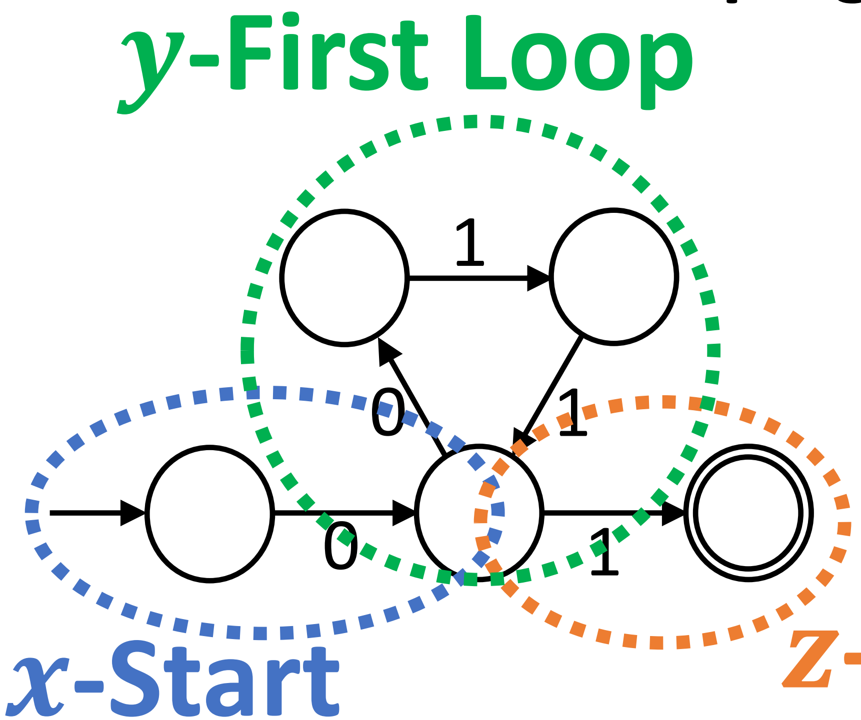


# Quest for Regular Language Properties

Suppos… …gth
(i.e. $\forall n$ …
1. W… .
2. $\exists$ strings longer than the number of states.

**The Pumping Lemma is our property that all regular languages must have. So, if some language does not have that property, it cannot be a regular language.**

**Pumping Lemma**




~~Summary~~: Given a regular language $L$, $\exists$ a number $p$ such that any string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

1. $xy^iz \in L, \forall i \geq 0$.
2. $|y| > 0$.
3. $|xy| \leq p$.

# Pumping Lemma

Pumping Lemma: Given a regular language $L$, $\exists$ a number $p$ such that any string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

1. $xy^iz \in L, \forall i \geq 0.$
2. $|y| > 0.$
3. $|xy| \leq p.$

# Pumping Lemma

<u>Pumping Lemma</u>: Given a regular language $L$, $\exists$ a number $p$ such that <u>any</u> string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

1. $xy^iz \in L, \forall i \geq 0.$
2. $|y| > 0.$
3. $|xy| \leq p.$

Pumping Lemma User Manual:

1. The pumping lemma says all regular languages have property $\boldsymbol{P}$.
2. If we can show a language does not have property $\boldsymbol{P}$, then it cannot be regular.

# Non-Regularity Proofs

Pumping Lemma: **Given a regular language $L$**, $\exists$ a number $p$ such that any string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

1. $xy^i z \in L, \forall i \geq 0.$
2. $|y| > 0.$
3. $|xy| \leq p.$

1. Suppose language is regular.

# Non-Regularity Proofs

Pumping Lemma: Given a regular language $L$, **∃ a number $\boldsymbol{p}$** such that any string $s \in L$, with $|s| \geq p$, can be divided into three pieces, $s = xyz$ satisfying:

1. $xy^iz \in L, \forall i \geq 0$.
2. $|y| > 0$.
3. $|xy| \leq p$.

1. Suppose language is regular.
2. Select $p$ from pumping lemma.

# Non-Regularity Proofs

Pumping Lemma: Given a regular language $L$, $\exists$ a number $p$ **such that any string $\boldsymbol{s} \in \boldsymbol{L}$, with $|\boldsymbol{s}| \geq \boldsymbol{p}$**, can be divided into three pieces, $s = xyz$ satisfying:

1. $xy^i z \in L, \forall i \geq 0$.
2. $|y| > 0$.
3. $|xy| \leq p$.

1. Suppose language is regular.
2. Select $p$ from pumping lemma.
3. Carefully select $s \in L$ and $|s| \geq p$.

# Non-Regularity Proofs

Pumping Lemma: Given a regular language $L$, $\exists$ a number $p$ such that any string $s \in L$, with $|s| \geq p$, **can be divided into three pieces, $s = xyz$ satisfying:**

1. $xy^iz \in L, \forall i \geq 0$.
2. **$|y| > 0$.**
3. **$|xy| \leq p$.**

1. Suppose language is regular.
2. Select $p$ from pumping lemma.
3. Carefully select $s \in L$ and $|s| \geq p$.
4. Determine what $y$ must consist of.

# Non-Regularity Proofs

<u>Pumping Lemma</u>: Given a regular language $L$, $\exists$ a number $p$ such that <u>any</u> string $s \in L$, with $|s| \geq p$, **can be divided into three pieces, $s = xyz$ satisfying:**

1. $\boldsymbol{xy^iz \in L, \forall i \geq 0.}$
2. $|y| > 0.$
3. $|xy| \leq p.$

1. Suppose language is regular.
2. Select $p$ from pumping lemma.
3. Carefully select $s \in L$ and $|s| \geq p$.
4. Determine what $y$ must consist of.
5. Make new string by selecting $i$.

# Non-Regularity Proofs

Pumping Lemma: Given a regular language $L$, $\exists$ a number $p$ such that any string $s \in L$, with $|s| \geq p$, **can be divided into three pieces, $s = xyz$ satisfying:**

1. $xy^iz \in L, \forall i \geq 0$.
2. $|y| > 0$.
3. $|xy| \leq p$.

1. Suppose language is regular.
2. Select $p$ from pumping lemma.
3. Carefully select $s \in L$ and $|s| \geq p$.
4. Determine what $y$ must consist of.
5. Make new string by selecting $i$.
6. Show new string is not in language.