PUMPING LEMMA FOR REGULAR LANGUAGES

Chapter 1.4: Proving Languages Not to Be Regular

Materials used from Dr. Lisong Xu 's "Automata, Computation, and Formal Languages" class taught at University of Nebraska - Lincoln

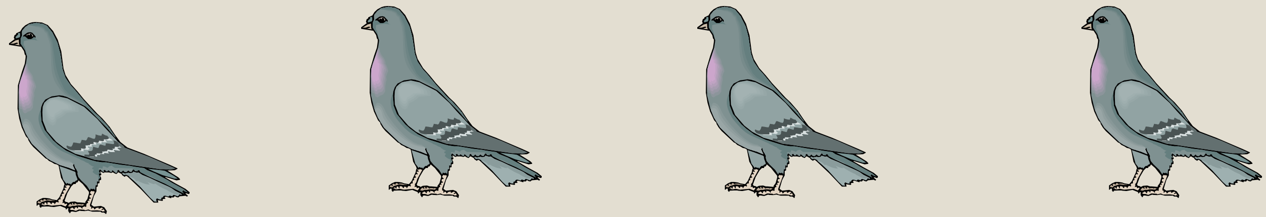
Pumping Lemma

- The pumping lemma describes a property of all regular languages.
- How a regular language “generates” infinitely many strings?
- Why study the pumping lemma?
  - The pumping lemma can be used to prove that some language is not regular.

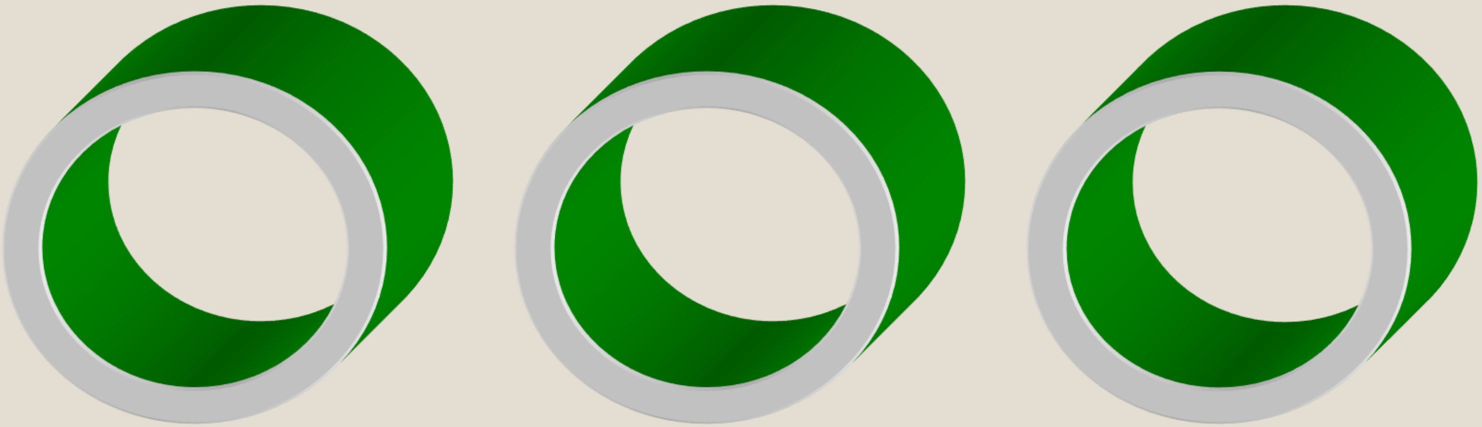
Pigeonhole Principle

- Pigeonhole principle
  - if more than *p* pigeons are placed into *p* holes, then some hole must have *more than one* pigeon in it

4 pigeons

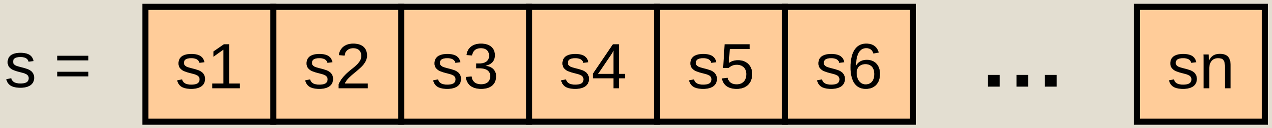


3 holes



Basic Idea of Pumping Lemma (1)

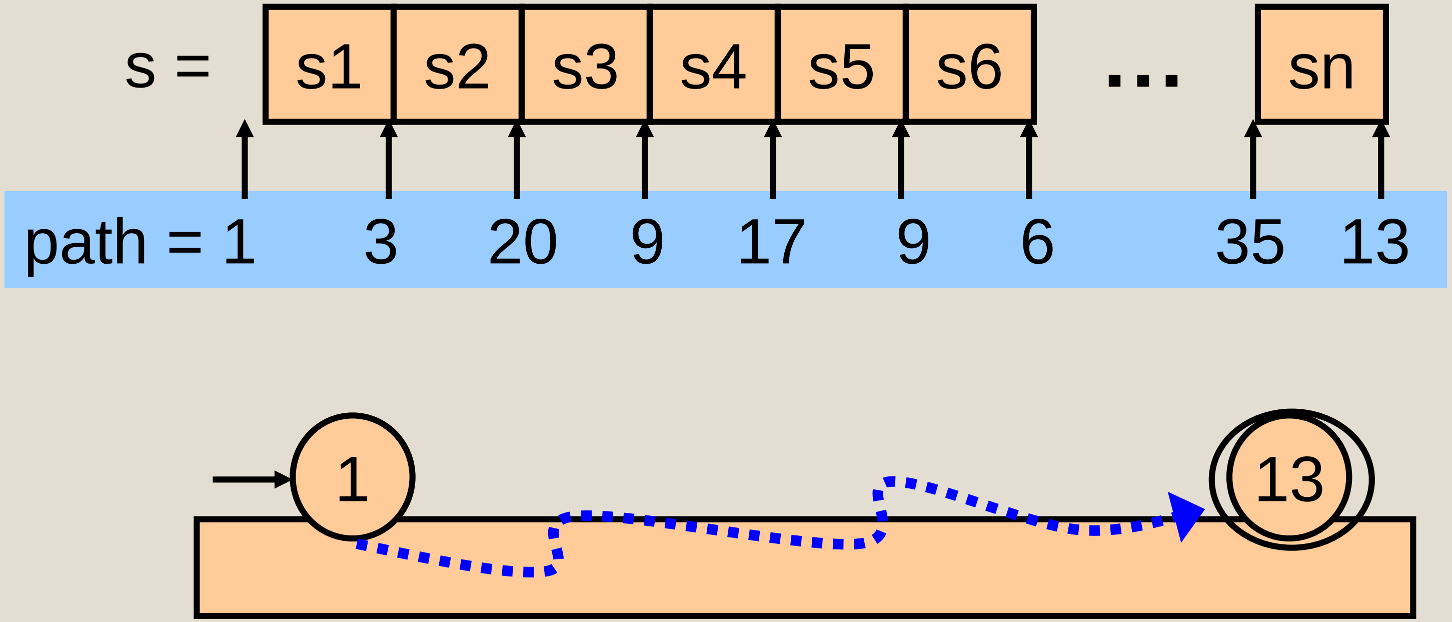
- Consider
  - a regular language A
  - a DFA for A
  - A string s ∈ A with n symbols



DFA for A

Basic Idea of Pumping Lemma (2)

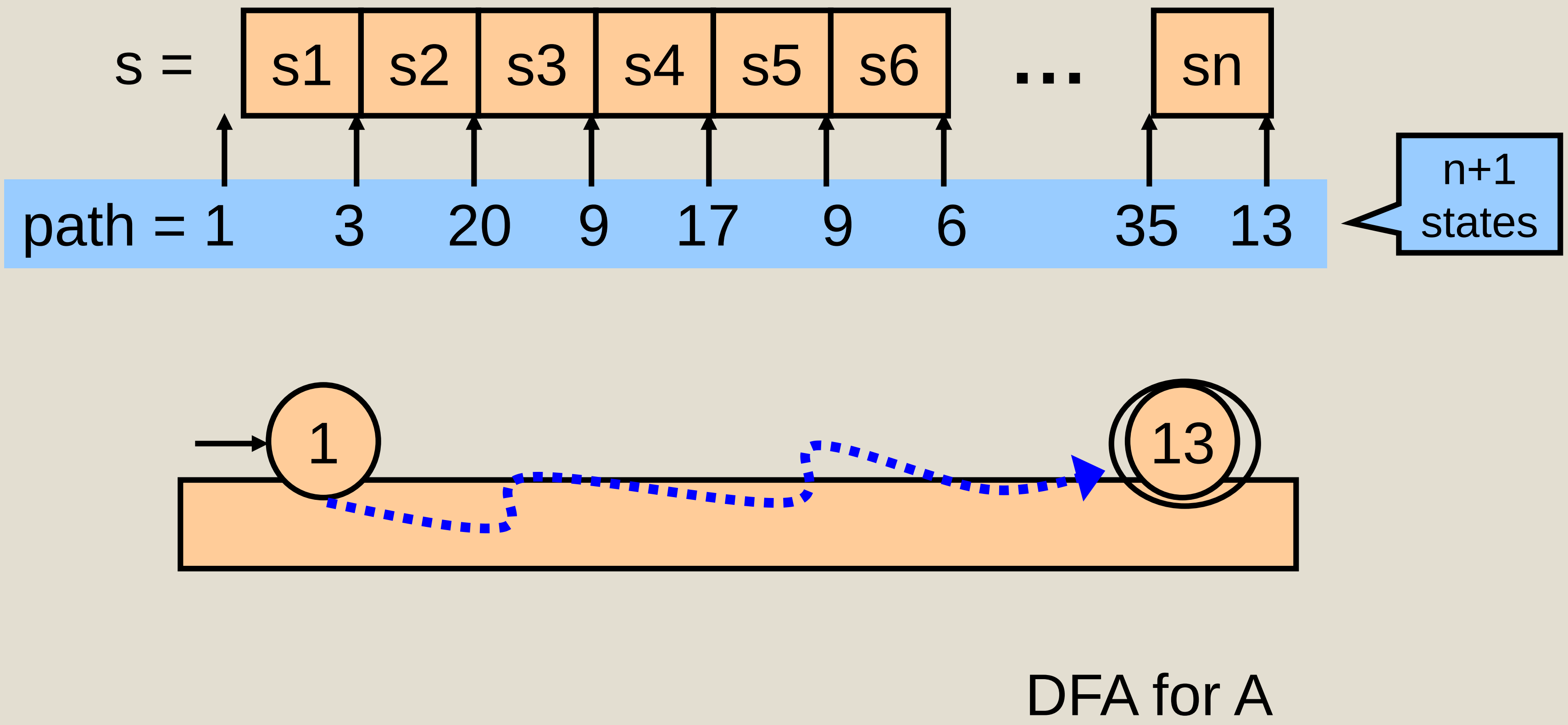
- Consider the path that the machine goes through when reading input string s.
- The path starts with state 1 (the start state), goes through some intermediate states (say 3, 20, 9, …), and finally ends at state 13 (which is a final state)



DFA for A

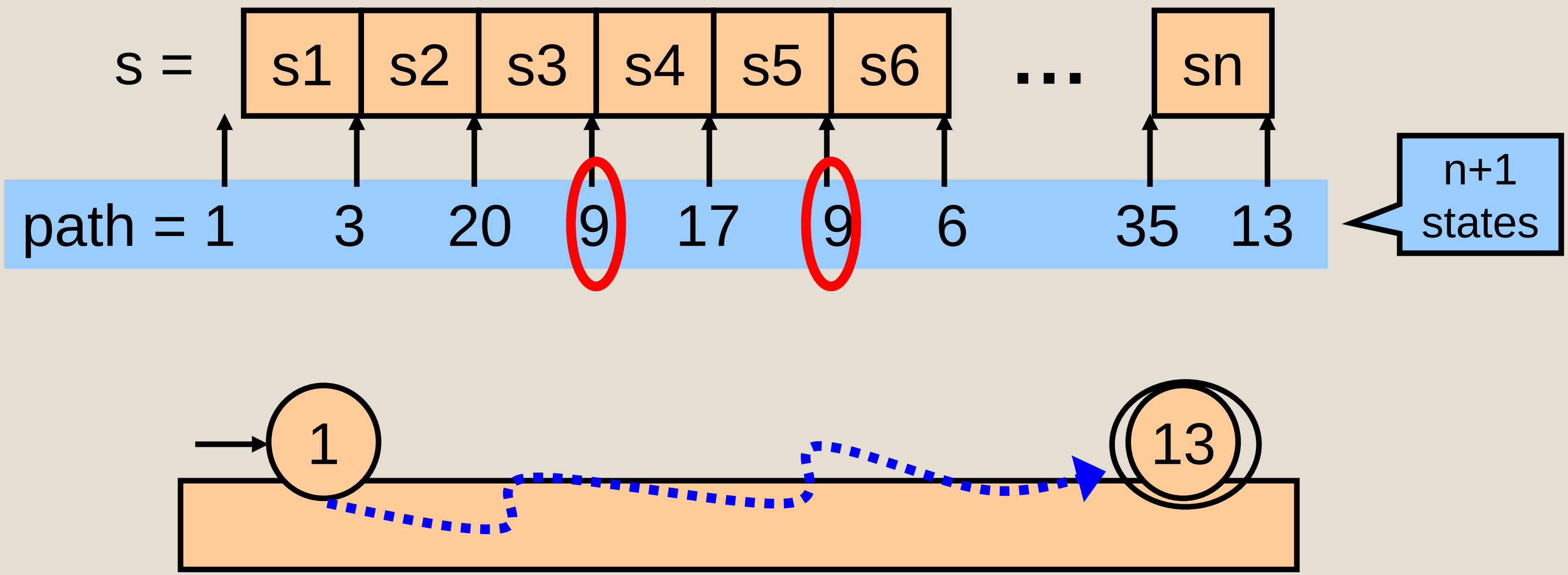
Basic Idea of Pumping Lemma (3)

- Since the length of string s is n, so the path consists of n+1 states.



Basic Idea of Pumping Lemma (4)

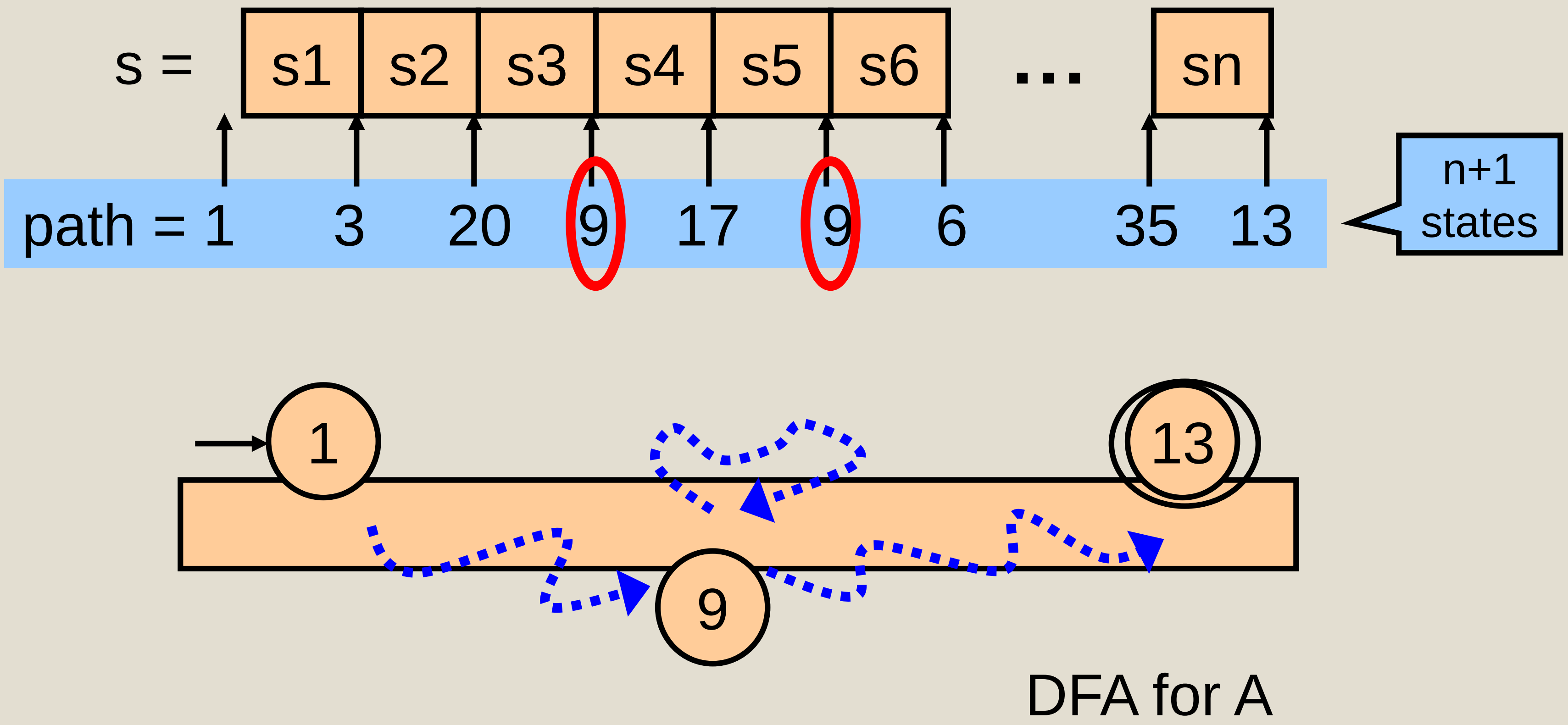
- If n+1 is greater than the number of DFA states,
- then the path goes through some state at least twice. (here is state 9)



DFA for A

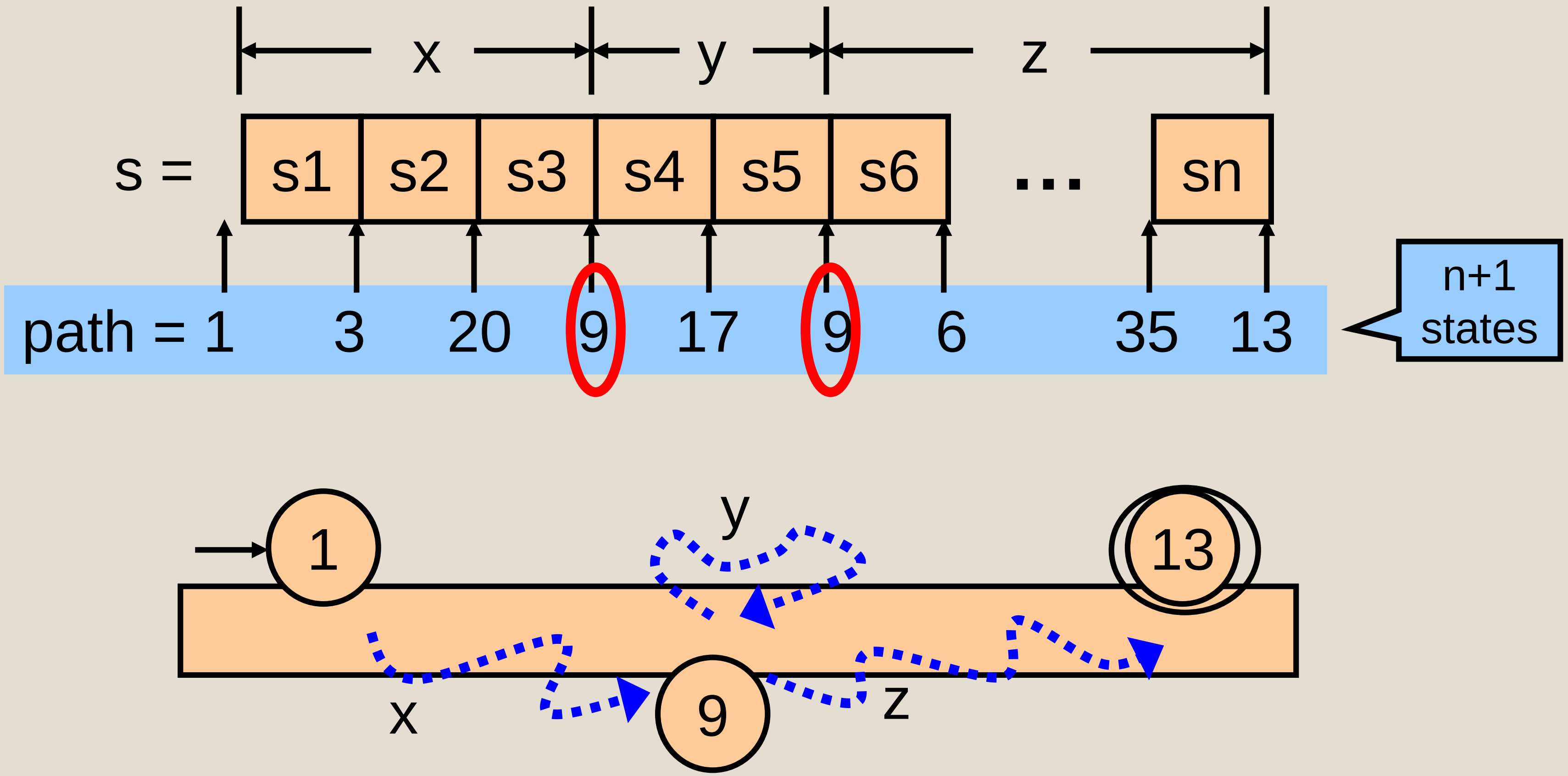
Basic Idea of Pumping Lemma (5)

- Since the path goes through state 9 twice, there is a loop in the path. The loop starts from state 9 and ends at state 9



Basic Idea of Pumping Lemma (6)

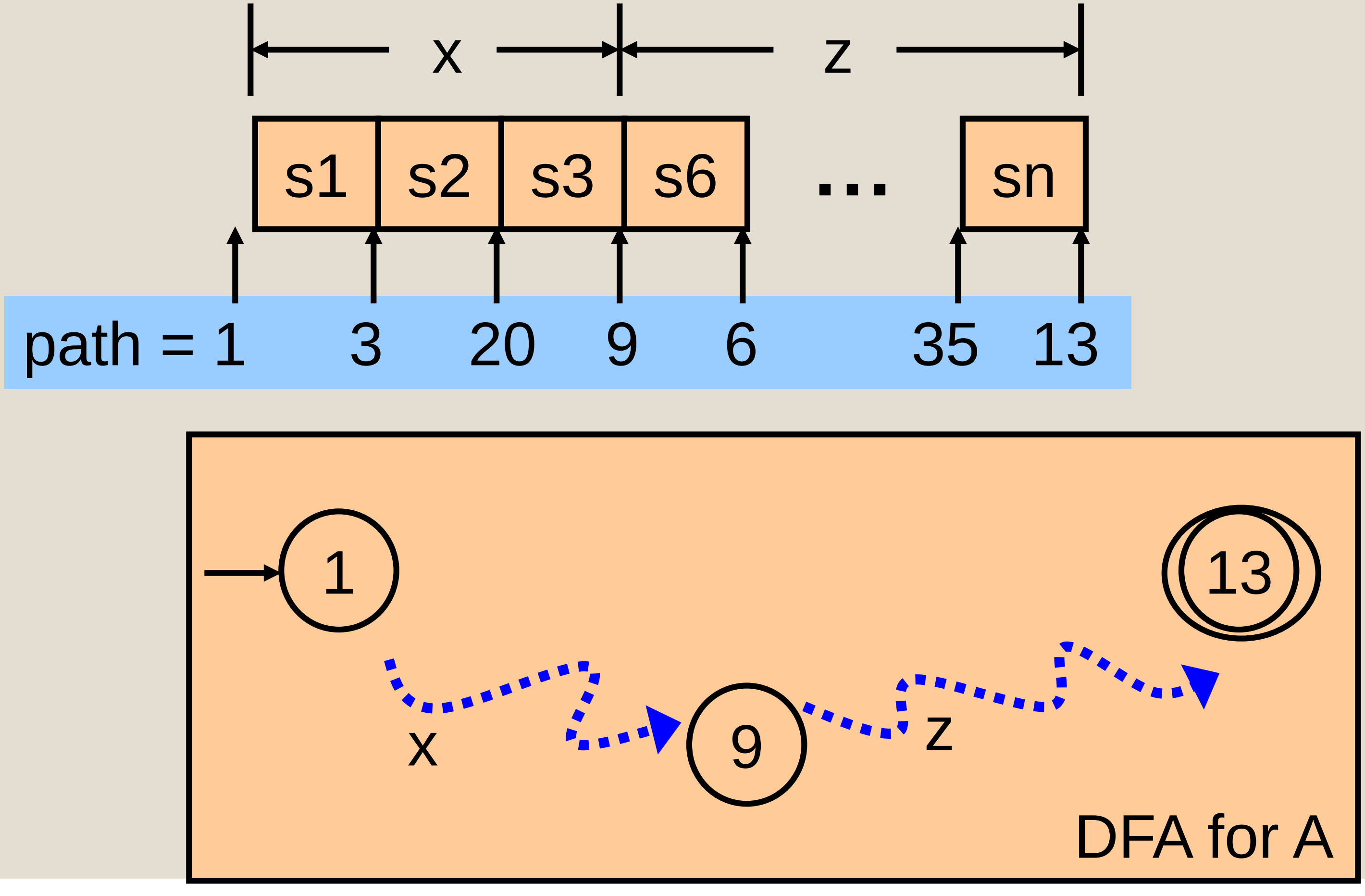
- Now we divide the string s into three pieces x, y, and z.



DFA for A

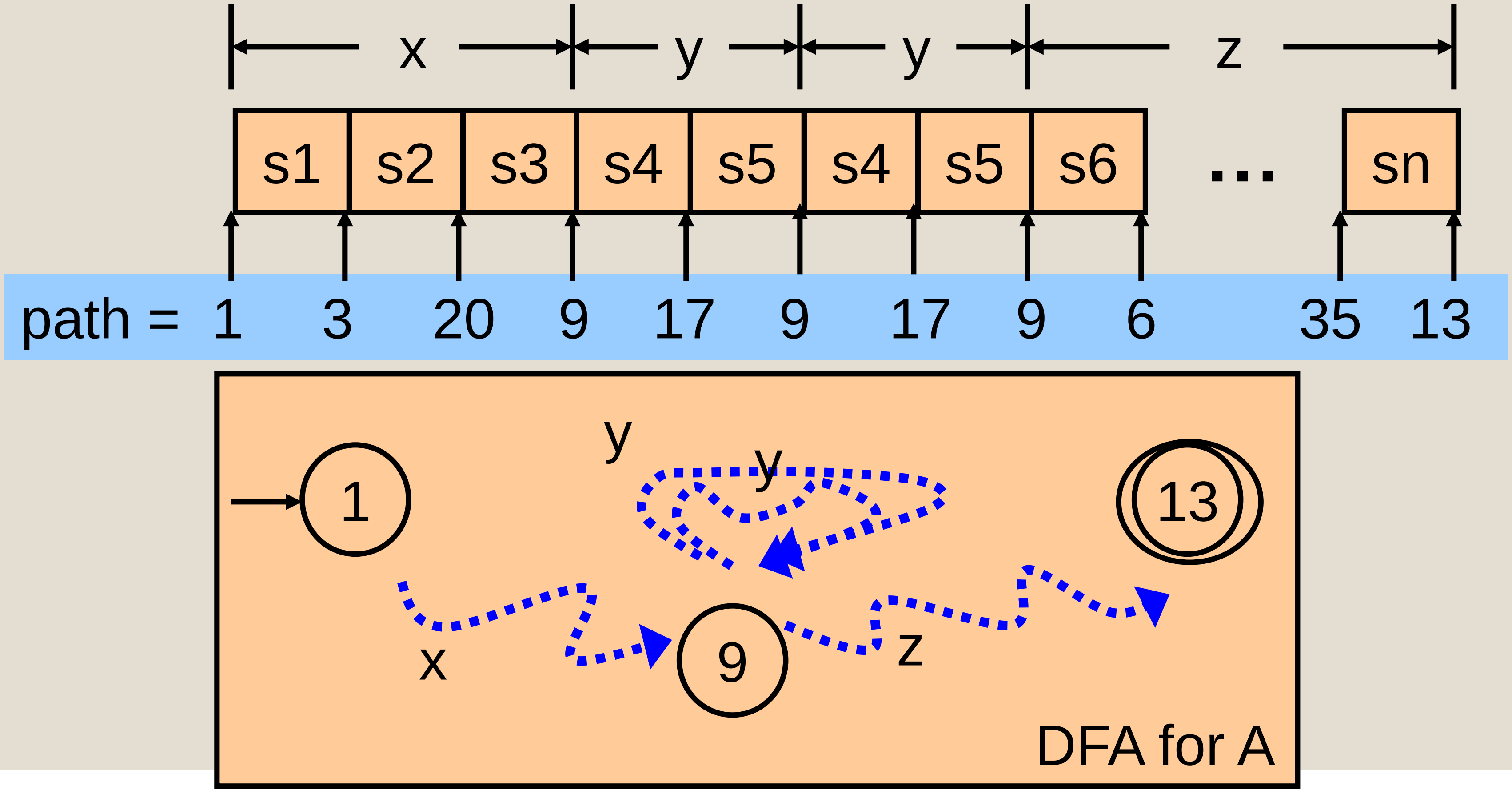
Basic Idea of Pumping Lemma (7)

- Consider another string xz. x takes the machine from state 1 to state 9, and then z takes the machine from state 9 to state 13 (a final state). So, xz is also accepted by the machine



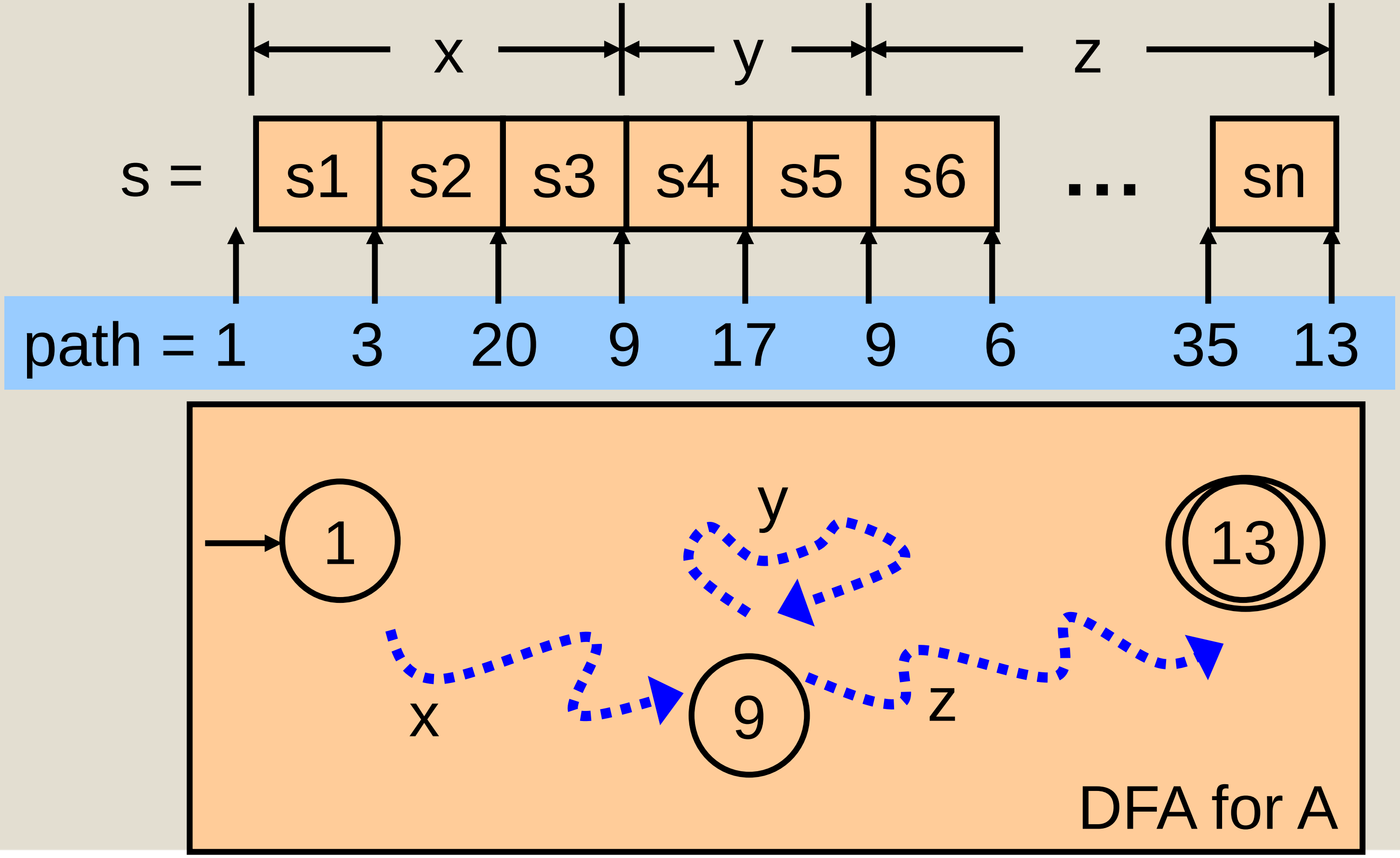
Basic Idea of Pumping Lemma (8)

- String xyyz: x takes the machine from 1 to 9; first y takes the machine from 9 back to 9, as does the second y; and z takes the machine from 9 to 13 (final state). So, xyyz is also accepted by the machine



Basic Idea of Pumping Lemma (9)

- So, for any integer $i \geq 0$, we have
  xy^iz is accepted by the machine



Pumping Lemma

- If A is a regular language, then there is a number p (called the pumping length) where, if s is any string in A with $|s| \geq p$, then s may be divided into three pieces, $s = xyz$, satisfying the following conditions:

1. for each $i \geq 0$, $xy^iz \in A$
2. $|y| > 0$, and
3. $|xy| \leq p$

- The pumping lemma describes a property of all regular languages.

Pumping Lemma

- If A is a regular language, then there is a number p (called the pumping length) where, if s is any string in A with $|s| \geq p$, then s may be divided into three pieces, $s=xyz$, satisfying the following conditions:

1. for each $i \geq 0$, $xy^iz \in A$
2. $|y| > 0$, and
3. $|xy| \leq p$

- One possible value of p is the number of DFA states.

Pumping Lemma

- If A is a regular language, then there is a number p (called the pumping length) where, if s is any string in A with $|s| \geq p$, then s may be divided into three pieces, $s=xyz$, satisfying the following conditions:

1. for each $i \geq 0$, $xy^iz \in A$
2. $|y| > 0$, and
3. $|xy| \leq p$

- i could be any integer at least 0
- If i=1, then xy^iz is just string s itself
- If i>1, then it is "pumping up string s"
- If i=0, then it is "pumping down string s"

Pumping Lemma

- If A is a regular language, then there is a number p (called the pumping length) where, if s is any string in A with $|s| \geq p$, then s may be divided into three pieces, $s = xyz$, satisfying the following conditions:

1. for each $i \geq 0$, $xy^iz \in A$
2. $|y| > 0$, and
3. $|xy| \leq p$

- Substring y (the loop) cannot not be empty, but x and z could be empty

Pumping Lemma

- If A is a regular language, then there is a number p (called the pumping length) where, if s is any string in A with $|s| \geq p$, then s may be divided into three pieces, $s=xyz$, satisfying the following conditions:

1. for each $i \geq 0$, $xy^iz \in A$
2. $|y| > 0$, and
3. $|xy| \leq p$

- In general, there exists a loop for any p symbols.
- The pumping lemma is interested only in the loop in the first p symbols.

Pumping Lemma

- If A is a regular language, then there is a number p (called the pumping length) where, if s is any string in A with $|s| \geq p$, then s may be divided into three pieces, $s=xyz$, satisfying the following conditions:
  1. for each $i \geq 0$, $xy^iz \in A$
  2. $|y| > 0$, and
  3. $|xy| \leq p$

Intuitive explanation:

- If A is regular, then for any long enough string s in A, some part of its first p symbols can be pumped.

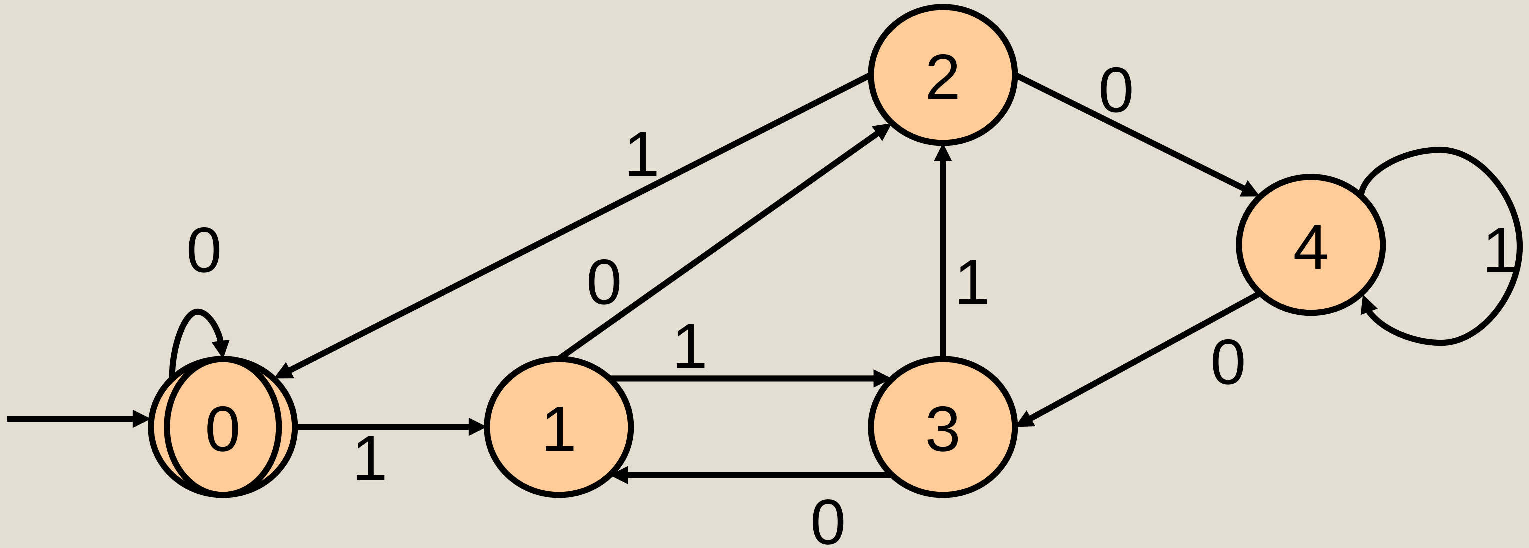
Outline

- ✓Pumping Lemma
- ➢Examples of Pumping Lemma
  - Contrapositive of Pumping Lemma
  - Proof of Nonregular Languages

Example of Pumping Lemma

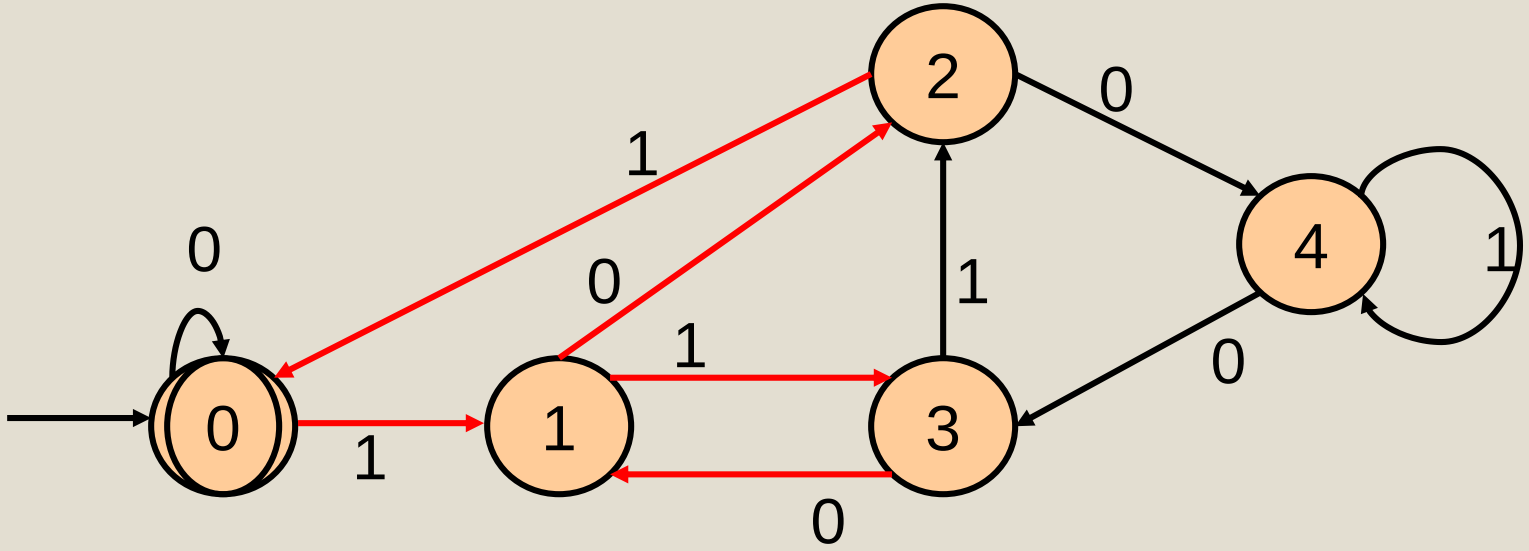
- Consider the regular language

  L = { x over {0,1} | x is the binary representation of an integer divisible by 5}



Example of Pumping Lemma

- The number of states of the finite automaton is 5. So we set p = 5
- Consider string 11001
  - 11001 $\in$ L
  - $|11001| = 5 \geq p$



Example of Pumping Lemma

- string 11001 could be divided into three pieces as follows
  - x = 1
  - y = 10
  - z = 01
- Condition 1:
  - i=0, $xy^iz = 101 \in L$ $(101_2=5)$
  - i=2, $xy^iz = 1101001 \in L$ $(1101001_2=105)$
  - i=3, $xy^iz = 110101001 \in L$ $(110101001_2=425)$
- Condition 2:
  - $|y| = |10| = 2 > 0$
- Condition 3:
  - $|xy| = |110| = 3 \le p$

Example of Pumping Lemma

- Alternatively, string 11001 could be divided into three pieces as follows
  - $x = \varepsilon$
  - $y = 11001$
  - $z = \varepsilon$
- Condition 1:
  - $i=0$, $xy^iz = \varepsilon \in L$ $(\varepsilon = 0)$
  - $i=2$, $xy^iz = 1100111001 \in L$ $(1100111001_2=825)$
- Condition 2:
  - $|y| = |11001| = 5 > 0$
- Condition 3:
  - $|xy| = |11001| = 5 \leq p$

Outline

- ✓Pumping Lemma
- ✓Examples of Pumping Lemma
- ➢Contrapositive of Pumping Lemma
- ○ Proof of Nonregular Languages

Intuitive Explanation

- Pumping Lemma: If *A* is regular, then for *any* long enough string *s* in *A*, *some part* of its first *p* symbols *can* be pumped.

- Contrapositive: If there *exists* a long enough string *s* in *A*, and *any part* of its first *p* symbols *cannot* be pumped, then *A* is not regular.

Contrapositive of Pumping Lemma

- Language A is not regular if, for every number p, there exists a string s in A with $|s| \geq p$ having the following property:

  For any decomposition of $s=xyz$, in which $|y|>0$ and $|xy| \leq p$, there is an $i \geq 0$ for which $xy^iz \notin A$

Outline

- ✓Pumping Lemma
- ✓Examples of Pumping Lemma
- ✓Contrapositive of Pumping Lemma
- ➢Proof of Nonregular Languages

Proof of Nonregular Languages

- Prove that a language A is not regular
  - step 1: For the purpose of contradiction, assume that A is regular
  - step 2: Let p be the pumping length
  - step 3: Pick a string s in A with $|s| \geq p$
  - step 4: Identify all possible decompositions of s into xyz, with $|xy| \leq p$ and $|y| > 0$
  - step 5: Show that for each decomposition, there exists an $i \geq 0$ such that $xy^iz \notin A$
  - step 6: Conclude that the assumption is wrong

- The purpose of step 3 is to find a string such that any part of its first p symbols cannot be pumped
- Choose a string whose first p symbols are as simple as possible. Otherwise, it may be difficult to show that ***any part*** of its first p symbols cannot be pumped.

Proof of Nonregular Languages

- Prove that a language A is not regular
  - step 1: For the purpose of contradiction, assume that A is regular
  - step 2: Let p be the pumping length
  - step 3: Pick a string s in A with $|s| \geq p$
  - step 4: Identify all possible decompositions of s into xyz, with $|xy| \leq p$ and $|y| > 0$
  - step 5: Show that for each decomposition, there exists an $i \geq 0$ such that $xy^iz \notin A$
  - step 6: Conclude that the assumption is wrong

- For different decompositions, you may choose different i's.
- If at step 5, you find that for some decomposition, $xy^iz \in A$ for any $i \geq 0$, then this means the string you picked is not a good choice. Do not despair, just try another one

Example 1

- L1 = { w | w has the same number of a's and b's}
  - For the purpose of contradiction, assume that L1 is regular
  - Let p be the pumping length
  - Let s = $(ab)^p$, and we have

    $(ab)^p \in$ L1, and $|(ab)^p| = 2p \geq p$
  - We decompose s into xyz, with $x = \varepsilon$, $y = aba$, $z = b(ab)^{p-2}$
  - Let $i = 0$, then $xy^iz = xz = b(ab)^{p-2} \notin$ L1
  - Therefore, L1 is not regular

we must consider all possible decompositions!

Example 1

- Prove that the following language is not regular

L1 = { w | w has the same number of a's and b's}

- step 1: For the purpose of contradiction, assume that L1 is regular
- step 2: Let p be the pumping length
- step 3: Pick a string s in L1 with $|s| \geq p$
  - Let $s = (ab)^p$, and we have

    $(ab)^p \in L1$, and $|(ab)^p| = 2p \geq p$

Example 1

- step 4: Identify all possible decompositions of s into xyz, with $|xy| \leq p$ and $|y| > 0$
- step 5: Show that for each decomposition, there exists an $i \geq 0$ such that $xy^iz \notin L1$
  - However, we can see that if $x=\varepsilon$, $y=ab$, and $z= (ab)^{p-1}$, then $xy^iz =(ab)^i(ab)^{p-1} = (ab)^{i+p-1} \in L1$ for any i.
  - So $(ab)^p$ is not a good choice for this language, and we have to find another one.

Example 1

- Prove that the following language is not regular

L1 = { w | w has the same number of a’s and b’s}

- step 1: For the purpose of contradiction, assume that L1 is regular
- step 2: Let p be the pumping length
- step 3: Pick a string s in L1 with $|s| \geq p$
  - Let $s = a^p b^p$, and we have

    $a^p b^p \in L1$, and $|a^p b^p| = 2p \geq p$

Example 1

- step 4: Identify all possible decompositions of s into xyz, with $|xy| \leq p$ and $|y| > 0$
  - Since $s = a^p b^p$, for any possible decomposition, y must consist of one or more a's but no b's
- step 5: Show that for each decomposition, there exists an $i \geq 0$ such that $xy^iz \notin L1$
  - Let $y = a^k$ ($p \geq k > 0$), and let $i = 2$, we see that

    $$xy^iz = xy^2z = a^{p+k}b^p \notin L1$$
- step 6: Conclude that the assumption is wrong. That is L1 is not regular

Example 2

- Prove that the following language is not regular

  $L2 = \{ ww \mid w \in \{a, b\}^*\}$

  - step 1: For the purpose of contradiction, assume that L2 is regular
  - step 2: Let p be the pumping length
  - step 3: Pick a string s in L2 with $|s| \geq p$
    - Let $s = a^p b a^p b$, and we have

      $a^p b a^p b \in L2$, and $|a^p b a^p b| = 2p+2 \geq p$

Example 2

- step 4: Identify all possible decompositions of s into xyz, with $|xy| \leq p$ and $|y| > 0$
  - Since $s = a^p b a^p b$, for any possible decomposition, y must consist of one or more a's but no b's
- step 5: Show that for each decomposition, there exists an $i \geq 0$ such that $xy^iz \notin L2$
  - Let $y = a^k$ $(p \geq k > 0)$, and let $i = 2$, we see that

$$xy^iz = xy^2z = a^{p+k}ba^pb \notin L2$$

- step 6: Conclude that the assumption is wrong. That is L2 is not regular

Example 3

- L3 = $\{ a^m b^n \mid m > n \}$

  - step 1: For the purpose of contradiction, assume that L3 is regular
  - step 2: Let p be the pumping length
  - step 3: Pick a string s in L3 with $|s| \geq p$
    - Let $s = a^{p+1}b^p$, and we have

      $a^{p+1}b^p \in$ L3, and $|a^{p+1}b^p| = 2p+1 \geq p$

Example 3

- step 4: Identify all possible decompositions of s into xyz, with $|xy| \leq p$ and $|y| > 0$
  - Since $s = a^{p+1}b^p$, for any possible decomposition, y must consist of one or more a's but no b's
- step 5: Show that for each decomposition, there exists an $i \geq 0$ such that $xy^iz \notin L3$
  - Let $y = a^k$ $(p \geq k > 0)$, and let $i = 0$, we see that

    $$xy^iz = xz = a^{p+1-k}b^p$$

  Because $p+1-k \leq p$, $a^{p+1-k}b^p \notin L3$
- step 6: Conclude that the assumption is wrong. That is L3 is not regular

Example 4

- L4 = { a^{n^2}| n is an integer $\geq$ 0}
  - step 1: For the purpose of contradiction, assume that L4 is regular
  - step 2: Let p be the pumping length
  - step 3: Pick a string s in L4 with |s| $\geq$ p
    - Let s = a^{p^2}, and we have

      $a^{p^2} \in$ L4, and $|a^{p^2}| = p^2 \geq p$

Example 4

- step 4: Find all possible decompositions of s into xyz, with $|xy| \leq p$ and $|y| > 0$
  - Since $s = a^{p^2}$, for any possible decomposition, y must consist of one or more a's
- step 5: Show that for any decomposition, there exists an $i \geq 0$ such that $xy^iz \notin L4$
  - Let $y = a^k$ $(p \geq k > 0)$, and let $i = 2$, we see that

$$xy^iz = xy^2z = a^{p^2+k}$$

since $p^2 < p^2+k \leq p^2+p < p^2+2p+1 = (p+1)^2$

so $a^{p^2+k} \notin L4$

- step 6: Conclude that the assumption is wrong. That is L4 is not regular