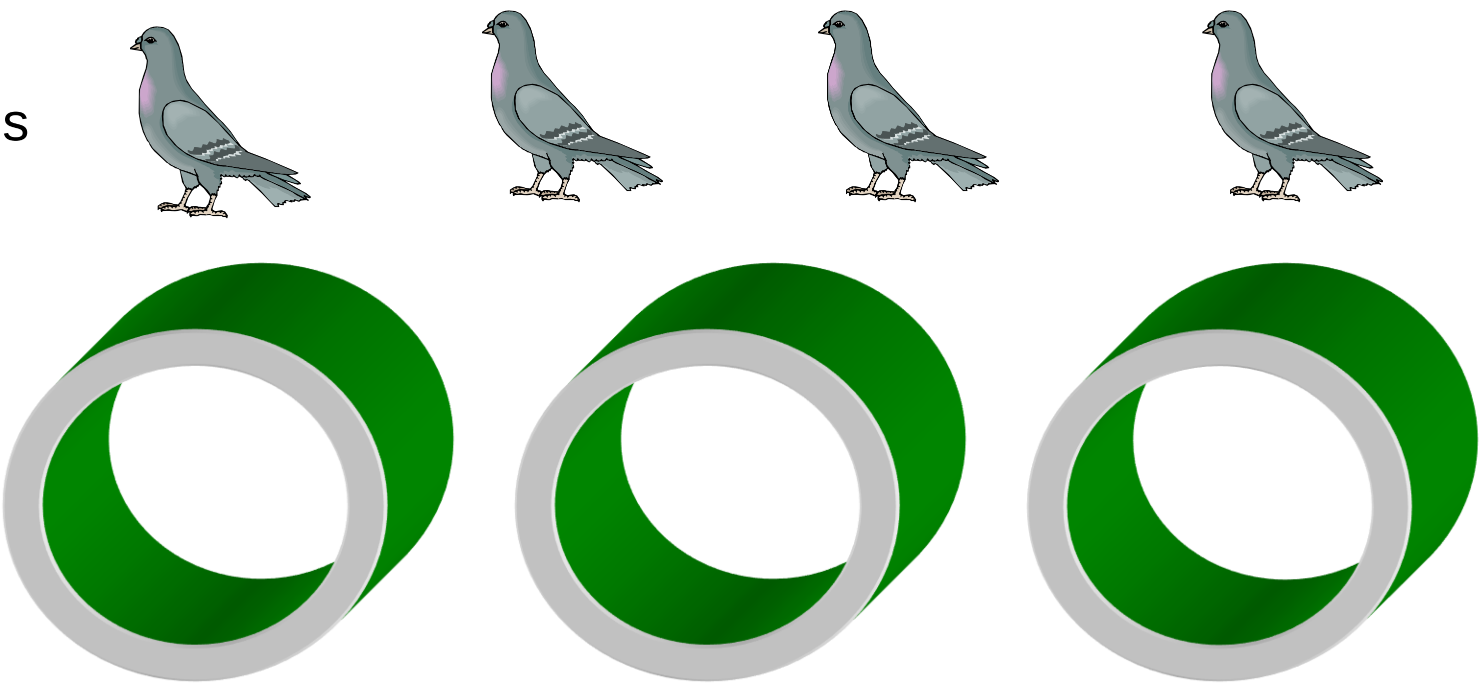
Pigeonhole Principle

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

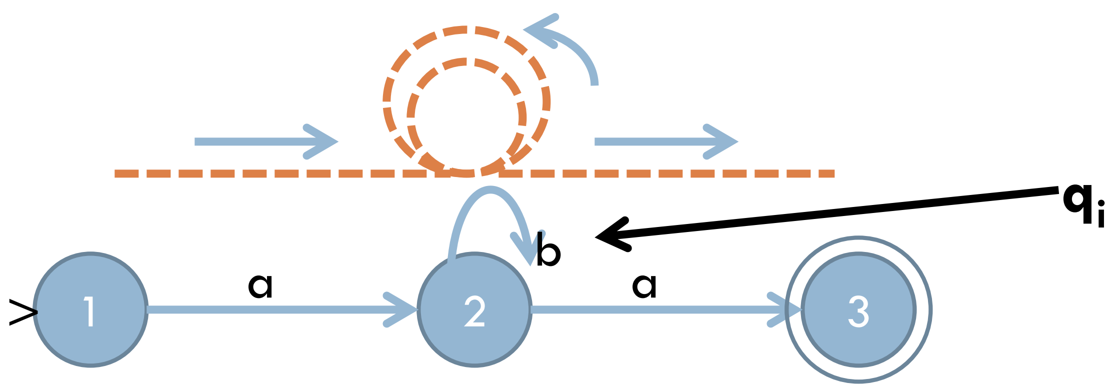
4 pigeons

3 holes



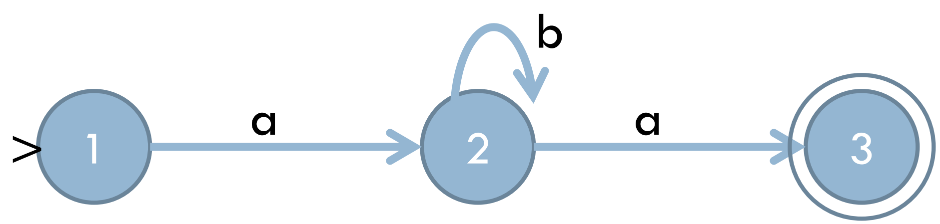
Pumping Lemma

- Let M be a DFA with k states. Any path of length k in M contains a cycle
  - A path of length k contains $k+1$ nodes (states). Since there are only k nodes in M, there must be a node, q_i, that occurs in at least two positions in the path. The subpath from the first occurrence of q_i to the second produces a cycle



Pumping Lemma

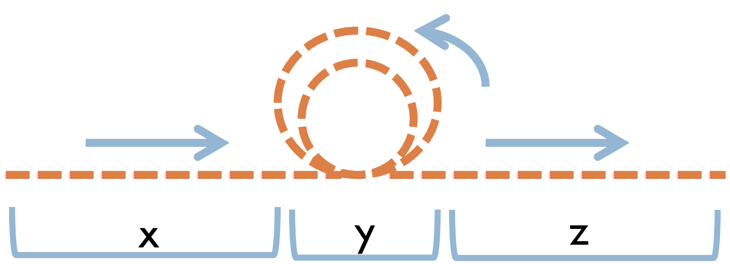
- The pumping lemma for regular languages requires strings in regular language to admit decompositions satisfying certain repetition properties
  - Consider a string $s=aba$ in L(M)
    - s can be decomposed into substrings x, y, z where $x=a$, $y=b$, and $z=a$, and $s=xyz$



    - The strings xy^iz can be obtained by **pumping** b in aba
      - The strings are accepted by DFA since the repetition of y simply adds additional trips around the loop

Pumping Lemma

- The pumping lemma can be used to show a language is non-regular by
  - Finding one string that does not satisfy the conditions of the pumping lemma
- To show a language is not regular using pumping lemma
  - Choose a string *s* in L and show that there is not decomposition *xyz* for s for which xy^iz is in L for all $i \geq 0$



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"

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

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

The pumping lemma describes a property of all regular languages

P can be the number of states in a DFA

Intuitive Explanation

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. $|xy| \leq p$

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

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

Applying the Pumping Lemma

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

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 panic, just try another one

Pumping Lemma

- Example: $L=\{a^n b^n \mid n \geq 0\}$
  - Assume L is regular, and let *p* be the number specified by the pumping lemma
  - Let *s* be the string $a^p b^p$ and $|s| \geq p$. There exist substrings *x*, *y*, *z*, such that $s=xyz$, $|xy| \leq p$, $|y|>0$, and xy^iz is in L for all $i \geq 0$

$$\begin{array}{ccc} x & y & z \\ a^r & a^t & a^{p-r-t}b^p \end{array} \qquad \left\{ \begin{array}{l} r+t \leq p \\ t > 0 \end{array} \right.$$

  - Pumping *y* twice generates $xy^2z= a^r a^t a^t a^{p-r-t} b^p = a^p a^t b^p$
  - Since t>0, then $p+t \neq p$, which is a **contradiction**, thus $a^{p+t}b^p$ is *not* in L. Since *s* in L cannot be decomposed to satisfy the conditions of the pumping lemma, *L is not regular*

Pumping Lemma

- A simpler way to think of the Pumping Lemma
  - **For every** regular language *L*
  - **There exists** a constant *n*
  - **For every** string *w* in *L* such that $|w| \geq n$,
  - **There exists** a way to break up *w* into three strings $w=xyz$ such that $|y| > 0$, $|xy| \leq n$ and
  - **For every** $i>=0$, the string xy^iz is also in L

Pumping Lemma

- An alternative view: Game between you and an opponent

Assume L is regular

Choose some value p

Choose cleverly a string s in L of length $\geq p$

Break s into some xyz, where $|xy| \leq p$ and y is not null

Need to choose an $i \geq 0$ such that xy^iz is not in L to win the prize of non-regularity!

Pumping Lemma

More Examples

- Is $L1 = \{a^m b^n \mid m, n > 0\}$, regular? If yes, construct the DFA, if not, use pumping lemma to prove it
- Is $L2 = \{a^m b^n \mid m < n\}$, regular? If yes, construct the DFA, if not, use pumping lemma to prove it
- Is $L3 = \{ ww \mid w \in \{a, b\}^*\}$ regular? If yes, construct the DFA, if not, use pumping lemma to prove it