1.3 NONREGULAR LANGUAGES

Pumping Lemma

NONREGULAR LANGUAGES

- **How to prove that certain languages cannot be recognized by any finite automaton.**

NONREGULAR LANGUAGES

- **How to prove that certain languages cannot be recognized by any finite automaton.**

- $B=\{0^n1^n | n\geq 0\}$ is not regular.
  - the machine seems to need to remember how many 0s have been seen so far as it reads the input.

NONREGULAR LANGUAGES

- **How to prove that certain languages cannot be recognized by any finite automaton.**

- $B=\{0^n1^n \mid n\geq 0\}$ is not regular.
  - the machine seems to need to remember how many 0s have been seen so far as it reads the input.

- $C=\{w \mid w$ has an equal number of 0s and 1s$\}$
  - is not regular.
  - Same reason as language B.

Chapter 1

NONREGULAR LANGUAGES

- **How to prove that certain languages cannot be recognized by any finite automaton.**

- $B=\{0^n1^n \mid n\geq 0\}$ is not regular.
  - the machine seems to need to remember how many 0s have been seen so far as it reads the input.

- C={w|w has an equal number of 0s and 1s}
  - is not regular.
  - Same reason as language B.

- D={w|w has an equal number of occurrences of 01 and 10 as substrings}.
  - is regular
  - Try to construct the machine.

NONREGULAR LANGUAGES

How to prove that certain languages ?

Chapter 1

NONREGULAR LANGUAGES

How to prove that certain languages ?

How can we generate an acceptable long string using a FSM ?

- longer than the number of the states ???

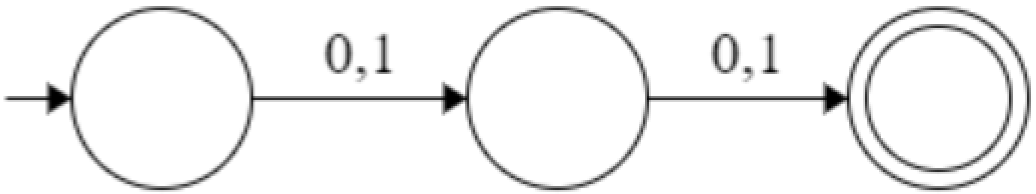
NONREGULAR LANGUAGES

How to prove that certain languages ?

How can we generate an acceptable long string **using a FSM ?**

- **longer than the number of the states ???**

- Assume that we want to design a **3-state FSM** ($|Q|=3$) that accepts strings s of **length 2 or more** ($|s|>3$).



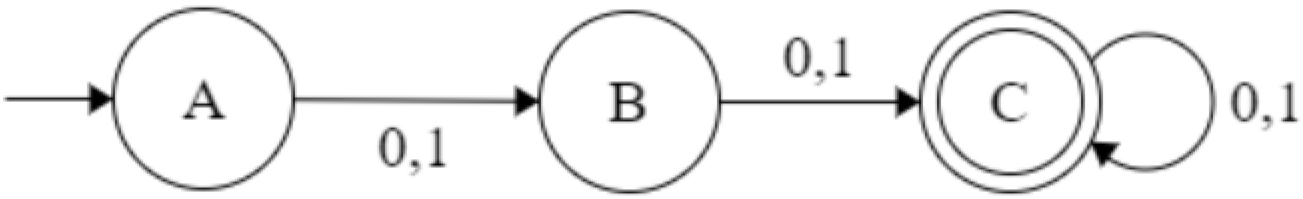
Chapter 1

NONREGULAR LANGUAGES

- **How we can generate a long string using a FSM ?**
  - **longer than the number of the states ???**

- Assume that we want to design a 3-state FSM (|Q|=3) that accepts strings s of length 4 or more (|s|>3).
- For example the number of states are 3, but the length of a binary string is 5.

- We need to use a **cycle**.
- Some **states** should be **visited more than once**.



Input : 1 1 0 1 1

State: A → B → C → C → C → C

Accept

NONREGULAR LANGUAGES

- **What is the maximum length of a string that a FSM can accept without containing a cycle?**

$$|S| < |Q|$$

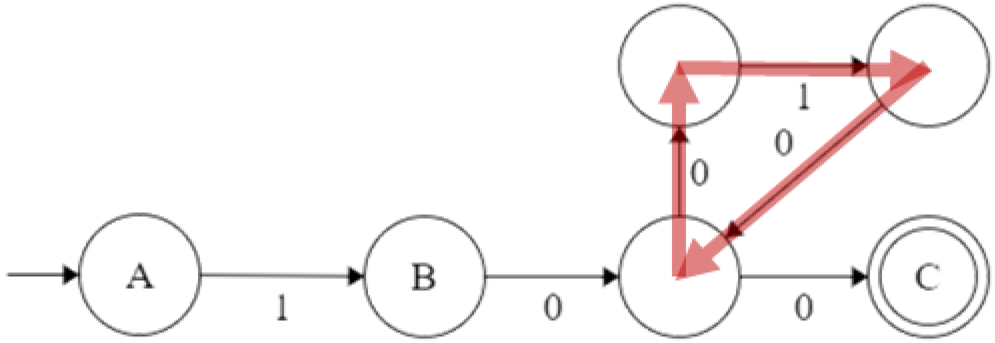
1. IF A FSM accepts **s** and $|S| \geq |Q|$, then in the sequence of the states that FSM takes after reading the input symbols, **there is a cycle**.

NONREGULAR LANGUAGES

- **What is the maximum length of a string that a FSM can accept without containing a cycle?**

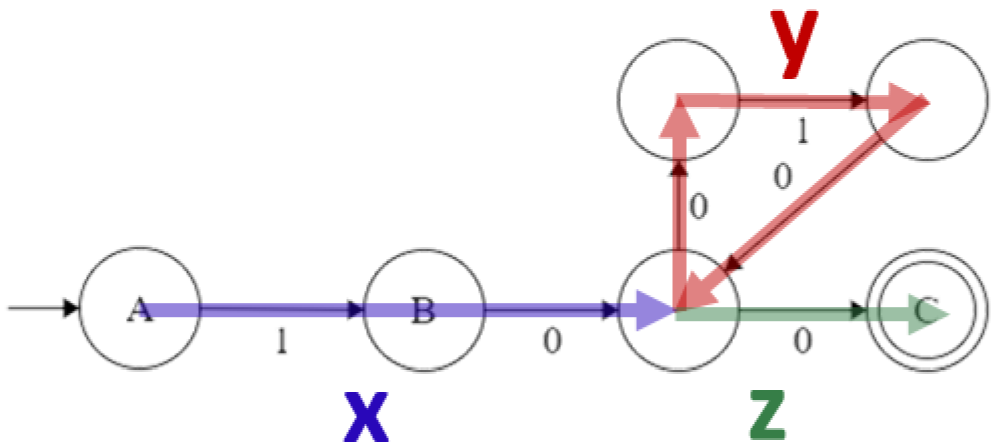
$$|S| < |Q|$$

1. IF A FSM accepts **s** and $|S| \geq |Q|$, then in the sequence of the states that FSM takes after reading the input symbols, **there is a cycle**.
2. The following machine accepts **100100100.**
   - $|s|=9 > |Q|=6$



NONREGULAR LANGUAGES

1. The following machine accepts the following strings.

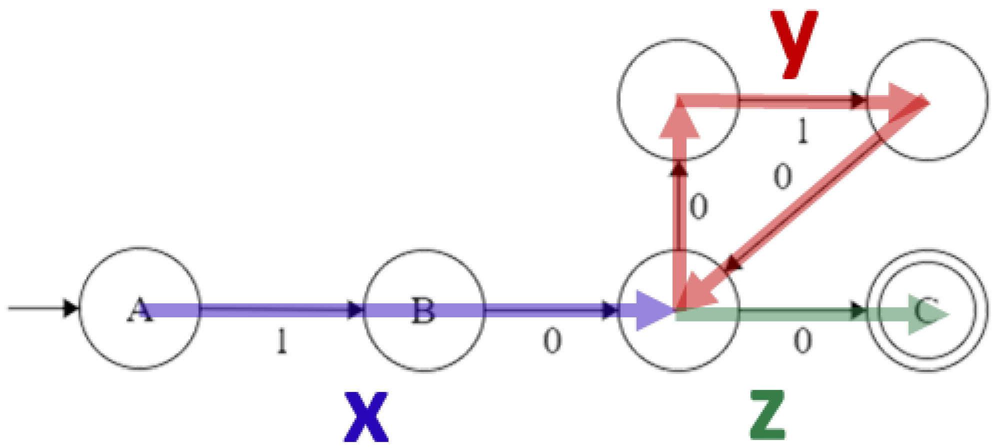


- 100
- 100100
- 100100100
- $10(010)^n0$

$s = x(y)^iz$

NONREGULAR LANGUAGES

1. The following machine accepts the following strings.



- 100
- 100100
- 100100100
- $10(010)^i0$

$s = x(y)^i z$

We will observe that:

If $|s| \geq |Q| \rightarrow$

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

Pumping Lemma

- **Theorem (Pumping Lemma):** If **A is a regular language**, then there is a number **p** (the pumping length) where if **s** is **any string in A of length at least p**, then **s** may be divided into three pieces, **s=xyz**, satisfying the following conditions:

Chapter 1

Pumping Lemma

- **Theorem (Pumping Lemma):** If **A is a regular language**, then there is a number **p** (the pumping length) where if **s** is **any string in A of length at least p**, then **s** may be divided into three pieces, $\boldsymbol{s = xyz}$, satisfying the following conditions:

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

Pumping Lemma

- **Theorem (Pumping Lemma):** If **A is a regular language**, then there is a number **p (the pumping length)** where if **s** is **any string in A of length at least 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. $|xy| \leq p$.

- Where
  - |s| represents the **length of string s,**
  - y^i means that i copies of y are concatenated together, and y^0 equals ε.

  - **Note:** the **pumping length** is the property of a regular language.

Pumping Lemma

Proof Idea:

- Let $M = (Q, \Sigma, \delta, q1, F)$ be a DFA that **recognizes A**.
- We assign the **pumping length** p to be the **number of states of M**.
- We show that **any strings s** in A of length at least p may be broken into the three pieces xyz, satisfying our three conditions.

Chapter 1

Pumping Lemma

Proof Idea:

- Let M=(Q,Σ,δ,q1,F) be a DFA that recognizes A.
- We assign the **pumping length** p to be the number of states of M.
- We show that **any strings s** in A of length at least p may be broken into the three pieces xyz, satisfying our three conditions.
- Assume |s|=n and the sequence of state after scanning s is as below.
- $\mathbf{q_{13}}$ is the **accept state**.

$$
\begin{array}{ccccccccccccccccc}
s = & & s_1 & & s_2 & & s_3 & & s_4 & & s_5 & & s_6 & & \cdots & & s_n & \\
& \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
& q_1 & & q_3 & & q_{20} & & q_9 & & q_{17} & & q_9 & & q_6 & & q_{35} & & q_{13}
\end{array}
$$

Pumping Lemma

Proof Idea:

- Let M=(Q,Σ,δ,q1,F) be a DFA that recognizes A.
- We assign the **pumping length** p to be the number of states of M.
- We show that **any strings s** in A of length at least p may be broken into the three pieces xyz, satisfying our three conditions.
- Assume |s|=n and the sequence of state after scanning s is as below.
- $\mathbf{q_{13}}$ is the **accept state**.

$$
\begin{array}{ccccccccccccccccccc}
s = & & s_1 & & s_2 & & s_3 & & s_4 & & s_5 & & s_6 & & \cdots & & s_n & \\
& \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
& q_1 & & q_3 & & q_{20} & & q_9 & & q_{17} & & q_9 & & q_6 & & q_{35} & & q_{13}
\end{array}
$$

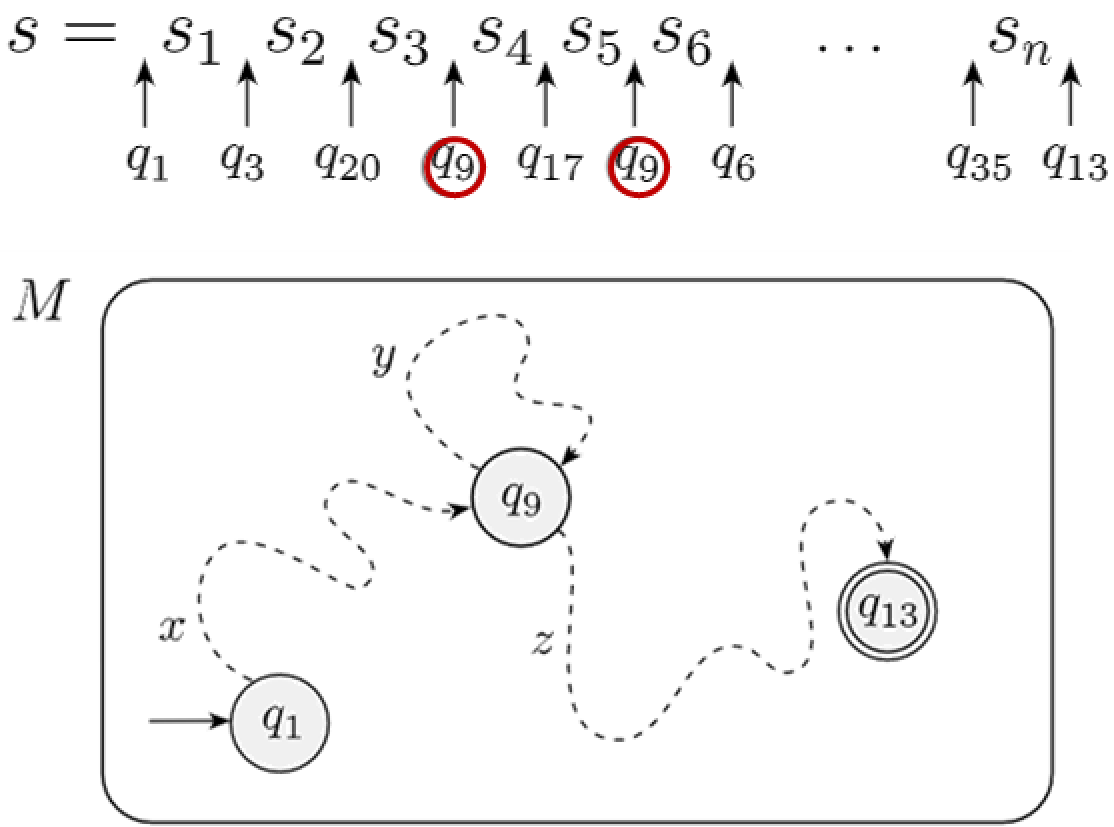
- If |s|= n → $|q_1q_3 \ldots q_{35}q_{13}|$ = n + 1.
- Since |n|≥|p| → n + 1 > p
- It means: the sequence must **contain a repeated state**. (pigeonhole principle)
- $\mathbf{q_9}$ has been **repeated two times** in the example above.

Pumping Lemma

Proof Idea: (cont.)

- We now divide **s** into the three pieces x, y, and z.
- this division of **s satisfies** the three conditions:

- Condition1: xy^iz is in A.
  - Substring x: $q_1 \rightarrow \ldots \rightarrow q_9$
  - Substring y: $q_9 \rightarrow \ldots \rightarrow q_9$
    - can be repeated.
  - Substring z: $q_9 \rightarrow \ldots \rightarrow q_{13}$

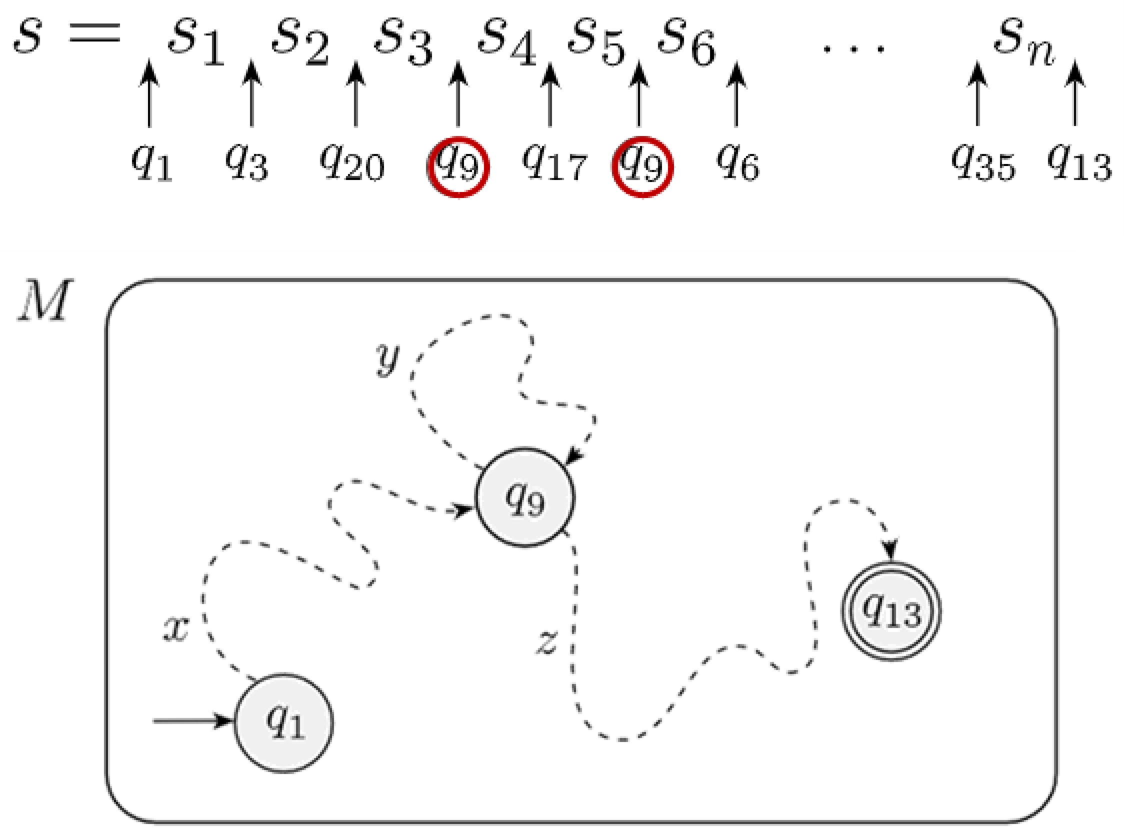


Pumping Lemma

Proof Idea: (cont.)

- We now divide **s** into the three pieces x, y, and z.
- this division of **s satisfies** the three conditions:

- Condition2: |y|> 0.
  - The part of s that occurred between two different occurrences of state q_9

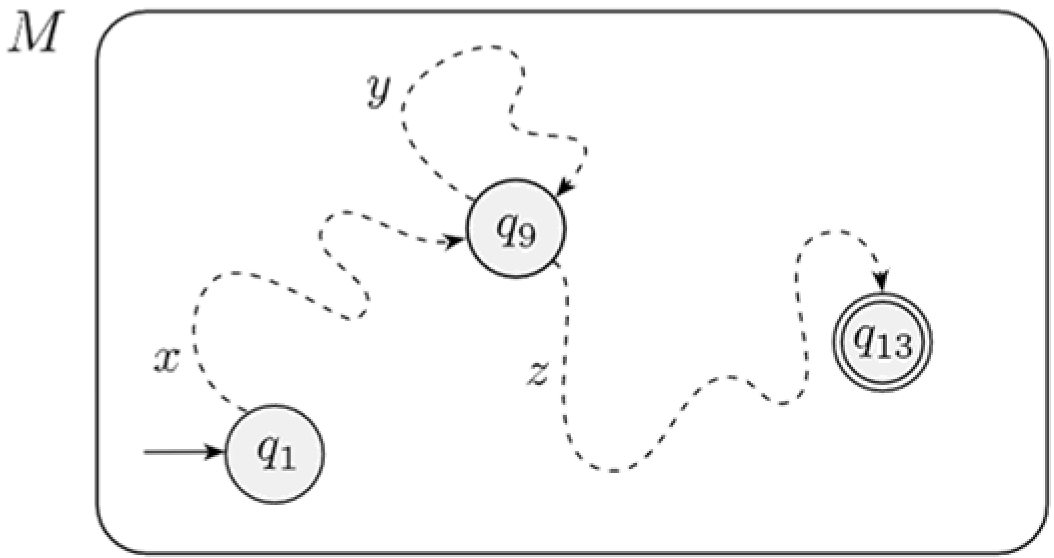


Pumping Lemma

Proof Idea: (cont.)

- We now divide **s** into the three pieces x, y, and z.
- this division of **s satisfies** the three conditions:

- Condition3: $|xy| \leq p$.
  - Let $\mathbf{q_9}$ is the **first repetition** in the sequence.
  - By the **pigeonhole principle**, the **first p+1 states** in the sequence must contain a repetition.
  - Therefore, $|xy| \leq p$.



Pumping Lemma

Proof :

- Let $M = (Q, \Sigma, \delta, q1, F)$ be a DFA recognizing A and p be the number of states of M.
- Let $s = s_1 s_2 \cdots s_n$ be a string in **A** of length **n**, where $n \geq p$.
- Let $r_1, \ldots, r_{n+1}$ be the sequence of states that M enters while processing s,
  - so $r_{i+1} = \delta(r_i, s_i)$ for $1 \leq i \leq n$.
  - This sequence has length $n + 1$, which is at least $p + 1$.

- **Among the first p+1 elements** in the sequence, two states (r_j, r_l) must be the same state,
  - by the pigeonhole principle.
- Because r_l occurs among the first p+1 places in a sequence starting at r_1, we have $l \leq p + 1$.

- **Now let** $x = s_1 \cdots s_{j-1}$, $y = s_j \cdots s_{l-1}$, and $z = s_l \cdots s_n$.

Pumping Lemma

Proof : (cont.)

- Now let $x = s_1 \cdots s_{j-1}$, $y = s_j \cdots s_{l-1}$, and $z = s_l \cdots s_n$.

- M must accept xy^iz for $i \geq 0$. → condition 1
- We know that $j \neq l$, so $|y|>0$ → condition 2
- and $l \leq p+1$, so $|xy| \leq p$. → condition 3

How to use Pumping Lemma to prove that a language A is not regular.

By contradiction:

- Assume A is regular.
  - It has a pumping length, called "p".
  - All strings longer than p can be pumped. $|s| \geq p$.

How to use Pumping Lemma to prove that a language A is not regular.

By contradiction:

- Assume A is regular.
  - It has a pumping length, called “p”.
  - All strings longer than p can be pumped. $|s| \geq p$.
- Find a string “s” in A such that $|s| \geq p$.
  - Divide s into xyz.
  - Show that $xy^iz \notin A$ for some.
  - Then consider **all ways** that s can be divided into xyz.

How to use Pumping Lemma to prove that a language A is not regular.

By contradiction:

- Assume A is regular.
  - It has a pumping length, called "p".
  - All strings longer than p can be pumped. $|s| \geq p$.
- Find a string "s" in A such that $|s| \geq p$. (and cannot be found)
  - Divide s into xyz.
  - Show that $xy^iz \notin A$ for some.
  - Then consider all ways that s can be divided into xyz.
- Show that none of these can **satisfy all the 3 pumping conditions** at the same time.

- s cannot be pumped.
- The **existence of s contradicts** the pumping lemma if A were regular.

Pumping Lemma

Example:

- Let **B** be the language $\{0^n1^n | n \geq 0\}$. Use the pumping lemma to prove that B is not regular.

Pumping Lemma

Example:

- Let **B** be the language $\{0^n1^n | n \geq 0\}$. Use the pumping lemma to prove that B is not regular.
- Assume B is regular.
- Let p be the pumping length.
- Choose $s \in B$ to be the string 0^p1^p.

Pumping Lemma

Example:

- Let **B** be the language $\{0^n1^n | n \geq 0\}$. Use the pumping lemma to prove that B is not regular.
- Assume B is regular.
- Let p be the pumping length.
- Choose $s \in B$ to be the string 0^p1^p.
- **Pumping lemma:** s can be split into three pieces, $s = xyz$, where for any i≥0 the string $s = xy^iz$ is in B.

- We consider **three cases** to show that this result is impossible.

Pumping Lemma

Example: (cont.)

- We consider **three cases** to show that this result is impossible.
  - **Case 1:** the “y” is in the zeros part.

  - If $s = 0^6 1^6 = 00\underline{000}0111111$ → $xy^2z = 00\underline{0000000}0111111 = 0^9 1^6$ is not in B.

    x y z

Pumping Lemma

Example: (cont.)

- We consider **three cases** to show that this result is impossible.
  - **Case 2:** the "y" is in the ones part.
  - If $s = 0^6 1^6 = 000000\underline{111}11 \rightarrow xy^2z = 000000\underline{111111}11 = 0^6 1^9$ is not in B.

x y z

Pumping Lemma

Example: (cont.)

- We consider **three cases** to show that this result is impossible.
  - **Case 3:** the "y" has zeros and ones part.

  - If s = $0^6 1^6$=000000111111 → xy^2z = 00000010111111 = $0^7 101^7$ is not in B.

    x y z

Pumping Lemma

Example: (cont.)

- We consider **three cases** to show that this result is impossible.
  - **Note:** Cases 2 and 3 contradict with condition 3 of pumping lemma.
    - Condition 3 : $|xy| \leq p$

- **Case1 :** $s = 0^6 1^6 = 000000111111$

  x y z → $|xy| = 6 \leq p = 6$

- **Case2 : s =** $0^6 1^6 = 000000111111$

  x y z → $|xy| = 10 > p = 6$

- **Case1 :** $s = 0^6 1^6 = 000000111111$

  x y z → $|xy| = 7 > p = 6$

These cases contradict condition 3

Pumping Lemma

Example:

- Let **C = {w|w has an equal number of 0s and 1s}**. Use the pumping lemma to prove that C is not regular.
- Assume C is regular.
- Let p be the pumping length.
- Choose $s \in C$ to be the string 0^p10^p1.
- **Pumping lemma:** s can be split into three pieces, $s = xyz$, where for any i ≥ 0 the string xy^iz is in C.
- **To satisfy : Condition 2:** |y| > 0 and **Condition 3:** |xy|≤ p
  - y should be a substring of the first zeros. 000…0001000…0001
  - then xy^2z is a substring in form of 0^p10^p1. Then xy^2z is not in C.
  - Contradiction.

  - **C is not regular.**

Pumping Lemma

Example:

- Let F = {ww|w∈{0,1}*}. Use the pumping lemma to prove that F is not regular.
- Assume F is regular.
- Let p be the pumping length.
- Choose $s \in F$ to be the string 0^p10^p1.
- **Pumping lemma:** s can be split into three pieces, $s = xyz$, where for any i ≥ 0 the string xy^iz is in F.
- **To satisfy : Condition 2:** |y| > 0 and **Condition 3:** |xy|≤ p
  - y should be a substring of the first zeros. 000…0001000…0001
  - then xy^2z is a substring in form of 0^p10^p1. Then xy^2z is not in F.
  - **Contradiction.**

  - **F is not regular.**

Pumping Lemma

Example:

- Let D = $\{1^{n^2} | n \geq 0\}$. Use the pumping lemma to prove that D is not regular.
- Assume D is regular.
- Let p be the pumping length.
- Choose $s \in D$ to be the string $\mathbf{1^{p^2}}$.
- **Pumping lemma:** s can be split into three pieces, $s = xyz$, where for any i ≥ 0 the string xy^iz is in D.
- **Condition 3:** $|xy| \leq p \rightarrow |y| \leq p$
- $|s| = |xyz| = p^2$ and $= |xy^2z| \leq p^2 + p < p^2 + 2p + 1 = (p+1)^2$
- Then xy^2z is not in D.
  - Its length should be a perfect square, but its length is smaller than the next member of s in D.
- **Contradiction.**
- **D is not regular.**

Pumping Lemma

Example: (pumping down)

- Let E = $\{0^i1^j \mid i > j\}$. Use the pumping lemma to prove that E is not regular.
- Assume E is regular.
- Let p be the pumping length.
- Choose $s \in E$ to be the string $\mathbf{0^{p+1}1^p}$.
- **Pumping lemma:** s can be split into three pieces, $s = xyz$, where for any i ≥ 0 the string xy^iz is in E
- **Condition 3:** y consists only of zeros.
- Pumping up doesn't work.
- the string $s = xy^iz$ is in E even when I = 0. (pumping down)
- If **x= 0^p , y=0, and z = 1^p** → $xy^0z = xy = 0^p1^p \notin E$
- Then xy^0z is not in E.
- **Contradiction.**
- **E is not regular.**