The Pumping Lemma for Regular Languages

Nonregular languages

Consider the language $B = \{0^n1^n | n \geq 0\}$.

Nonregular languages

Consider the language $B = \{0^n1^n | n \geq 0\}$.

- If we attempt to find a DFA that recognizes B we discover that such a machine needs to remember how many 0s have been seen so far as it reads the input

Nonregular languages

Consider the language $B = \{0^n 1^n | n \geq 0\}$.

- If we attempt to find a DFA that recognizes B we discover that such a machine needs to remember how many 0s have been seen so far as it reads the input
- Because the number of 0s isn't limited, the machine needs to keep track of an unlimited number of possibilities

Nonregular languages

Consider the language $B = \{0^n 1^n | n \geq 0\}$.

- If we attempt to find a DFA that recognizes B we discover that such a machine needs to remember how many 0s have been seen so far as it reads the input
- Because the number of 0s isn't limited, the machine needs to keep track of an unlimited number of possibilities
- This cannot be done with any finite number of states

Intuition may fail us

- Just because a language appears to require unbounded memory to be recognized, it doesn't mean that it is necessarily so

Intuition may fail us

- Just because a language appears to require unbounded memory to be recognized, it doesn't mean that it is necessarily so
- Example:

Intuition may fail us

- Just because a language appears to require unbounded memory to be recognized, it doesn't mean that it is necessarily so
- Example:
  - $C = \{w \mid w$ has an equal number of 0s and 1s$\}$

Intuition may fail us

- Just because a language appears to require unbounded memory to be recognized, it doesn't mean that it is necessarily so
- Example:
  - $C = \{w \mid w$ has an equal number of 0s and 1s$\}$ not regular

Intuition may fail us

- Just because a language appears to require unbounded memory to be recognized, it doesn't mean that it is necessarily so
- Example:
  - $C = \{w \mid w$ has an equal number of 0s and 1s$\}$ not regular
  - $D = \{w \mid w$ has equal no of 01 and 10 substrings$\}$

Intuition may fail us

- Just because a language appears to require unbounded memory to be recognized, it doesn't mean that it is necessarily so
- Example:
  - $C = \{w \mid w$ has an equal number of 0s and 1s$\}$ not regular
  - $D = \{w \mid w$ has equal no of 01 and 10 substrings$\}$ regular

Language nonregularity

- The technique for proving nonregularity of some language is provided by a theorem about regular languages called pumping lemma

Language nonregularity

- The technique for proving nonregularity of some language is provided by a theorem about regular languages called pumping lemma
- Pumping lemma states that all regular languages have a special property

Language nonregularity

- The technique for proving nonregularity of some language is provided by a theorem about regular languages called pumping lemma
- Pumping lemma states that all regular languages have a special property
- If we can show that a language L does not have this property we are guaranteed that L is not regular.

Observation

Pumping lemma states that all regular languages have a special property.

Observation

Pumping lemma states that all regular languages have a special property.

Pumping lemma does not state that only regular languages have this property. Hence, the property used to prove that a language L is not regular does not ensure that language is L regular.

Observation

Pumping lemma states that all regular languages have a special property.

Pumping lemma does not state that only regular languages have this property. Hence, the property used to prove that a language L is not regular does not ensure that language is L regular.

Consequence: A language may not be regular and still have strings that have all the properties of regular languages.

Pumping property

All strings in the language can be “pumped" if they are at least as long as a certain value, called the pumping length

Pumping property

All strings in the language can be “pumped" if they are at least as long as a certain value, called the pumping length

Meaning: each such string in the language contains a section that can be repeated any number of times with the resulting string remaining in the language.

Theorem 1.70

Pumping Lemma: If A is a regular language, then there is a pumping length p such that:

Theorem 1.70

Pumping Lemma: If A is a regular language, then there is a pumping length p such that:

- If s is any string in A of length at least p,

Theorem 1.70

Pumping Lemma: If A is a regular language, then there is a pumping length p such that:

- 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:

Theorem 1.70

Pumping Lemma: If A is a regular language, then there is a pumping length p such that:

- 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 $i \geq 0$, $xy^i z \in A$

Theorem 1.70

Pumping Lemma: If A is a regular language, then there is a pumping length p such that:

- 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. $|y| > 0$

Theorem 1.70

Pumping Lemma: If A is a regular language, then there is a pumping length p such that:

- 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. $|y| > 0$
  3. $|xy| \leq p$

Interpretation

- Recall that $|s|$ represents the length of string s and y^i means that y may be concatenated i times, and $y^0 = \epsilon$

Interpretation

- Recall that $|s|$ represents the length of string s and y^i means that y may be concatenated i times, and $y^0 = \epsilon$
- When $s = xyz$, either x or z may be ϵ, but $y \neq \epsilon$

Interpretation

- Recall that $|s|$ represents the length of string s and y^i means that y may be concatenated i times, and $y^0 = \epsilon$
- When $s = xyz$, either x or z may be ϵ, but $y \neq \epsilon$
- Without condition $y \neq \epsilon$ theorem would be trivially true

Proof idea

Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA that recognizes A

Proof idea

Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA that recognizes A

- Assign a pumping length p to be the number of states of M

Proof idea

Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA that recognizes A

- Assign a pumping length p to be the number of states of M
- Show that any string $s \in A$, $|s| \geq p$ may be broken into three pieces xyz satisfying the pumping lemma's conditions

More ideas

- If $s \in A$ and $|s| \geq p$, consider a sequence of states that M goes through to accept s, example: $q_1, q_3, q_{20}, \ldots, q_{13}$

More ideas

- If $s \in A$ and $|s| \geq p$, consider a sequence of states that M goes through to accept s, example: $q_1, q_3, q_{20}, \ldots, q_{13}$
- Since M accepts s, q_{13} must be final;if $|s| = n$ then the length of $q_1, q_3, q_{20}, \ldots, q_{13}$ is $n + 1$

More ideas

- If $s \in A$ and $|s| \geq p$, consider a sequence of states that M goes through to accept s, example: $q_1, q_3, q_{20}, \ldots, q_{13}$
- Since M accepts s, q_{13} must be final;if $|s| = n$ then the length of $q_1, q_3, q_{20}, \ldots, q_{13}$ is $n + 1$
- Because $|s| = n$ and $|s| \geq p$ it result that $n + 1 > p$.

More ideas

- If $s \in A$ and $|s| \geq p$, consider a sequence of states that M goes through to accept s, example: $q_1, q_3, q_{20}, \ldots, q_{13}$
- Since M accepts s, q_{13} must be fi nal;if $|s| = n$ then the length of $q_1, q_3, q_{20}, \ldots, q_{13}$ is $n + 1$
- Because $|s| = n$ and $|s| \geq p$ it result that $n + 1 > p$.
- By pigeonhole principle:

More ideas

- If $s \in A$ and $|s| \geq p$, consider a sequence of states that M goes through to accept s, example: $q_1, q_3, q_{20}, \ldots, q_{13}$
- Since M accepts s, q_{13} must be final;if $|s| = n$ then the length of $q_1, q_3, q_{20}, \ldots, q_{13}$ is $n + 1$
- Because $|s| = n$ and $|s| \geq p$ it result that $n + 1 > p$.
- By pigeonhole principle:

  - If p pigeons are placed into fewer than p holes, some holes must hold more than one pigeon

More ideas

- If $s \in A$ and $|s| \geq p$, consider a sequence of states that M goes through to accept s, example: $q_1, q_3, q_{20}, \ldots, q_{13}$
- Since M accepts s, q_{13} must be fi nal;if $|s| = n$ then the length of $q_1, q_3, q_{20}, \ldots, q_{13}$ is $n + 1$
- Because $|s| = n$ and $|s| \geq p$ it result that $n + 1 > p$.
- By pigeonhole principle:

  - If p pigeons are placed into fewer than p holes, some holes must hold more than one pigeon

  the sequence $q_1, q_3, q_{20}, \ldots, q_{13}$ must contain a repeated state, see Figure 1

Recognition sequence

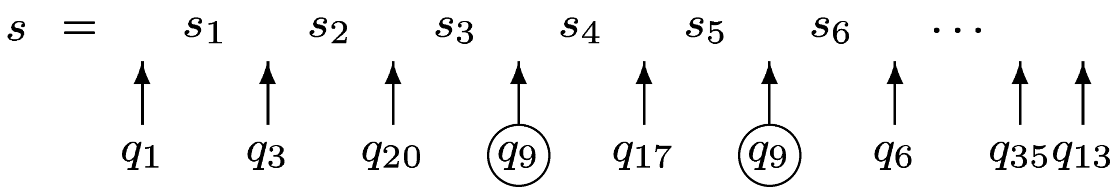

Figure 1: State q_9 repeats when M reads s

More ideas, continuation

Divide s in to the three pieces: x, y, and z

More ideas, continuation

Divide s in to the three pieces: x, y, and z

- Piece x is the part of s appearing before q_9

More ideas, continuation

Divide s in to the three pieces: x, y, and z

- Piece x is the part of s appearing before q_9
- Piece y is the part of s between two appearances of q_9

More ideas, continuation

Divide s in to the three pieces: x, y, and z

- Piece x is the part of s appearing before q_9
- Piece y is the part of s between two appearances of q_9
- Piece z is the part of s after the 2nd appearance of q_9

More ideas, continuation

Divide s in to the three pieces: x, y, and z

- Piece x is the part of s appearing before q_9
- Piece y is the part of s between two appearances of q_9
- Piece z is the part of s after the 2nd appearance of q_9

In other words:

More ideas, continuation

Divide s in to the three pieces: x, y, and z

- Piece x is the part of s appearing before q_9
- Piece y is the part of s between two appearances of q_9
- Piece z is the part of s after the 2nd appearance of q_9

In other words:

- x takes M from q_1 to q_9,

More ideas, continuation

Divide s in to the three pieces: x, y, and z

- Piece x is the part of s appearing before q_9
- Piece y is the part of s between two appearances of q_9
- Piece z is the part of s after the 2nd appearance of q_9

In other words:

- x takes M from q_1 to q_9,
- y takes M from q_9 to q_9,

More ideas, continuation

Divide s in to the three pieces: x, y, and z

- Piece x is the part of s appearing before q_9
- Piece y is the part of s between two appearances of q_9
- Piece z is the part of s after the 2nd appearance of q_9

In other words:

- x takes M from q_1 to q_9,
- y takes M from q_9 to q_9,
- z takes M from q_9 to q_{13}

Note

The division specified above satisfies the 3 conditions

Observations

Suppose that we run M on $xyyz$

Observations

Suppose that we run M on $xyyz$

- Condition 1: it is obvious that M accepts xyz, $xyyz$, and in general xy^iz for all $i > 0$. For $i = 0$, $xy^iz = xz$ which is also accepted because z takes M to q_{13}

Observations

Suppose that we run M on $xyyz$

- Condition 1: it is obvious that M accepts xyz, $xyyz$, and in general xy^iz for all $i > 0$. For $i = 0$, $xy^iz = xz$ which is also accepted because z takes M to q_{13}
- Condition 2: Since $|s| \geq p$, state q_9 is repeated. Then because y is the part between two successive occurrences of q_9, $|y| > 0$.

Observations

Suppose that we run M on $xyyz$

- Condition 1: it is obvious that M accepts xyz, $xyyz$, and in general xy^iz for all $i > 0$. For $i = 0$, $xy^iz = xz$ which is also accepted because z takes M to q_{13}
- Condition 2: Since $|s| \geq p$, state q_9 is repeated. Then because y is the part between two successive occurrences of q_9, $|y| > 0$.
- Condition 3: makes sure that q_9 is the first repetition in the sequence. Then by pigeonhole principle, the first $p + 1$ states in the sequence must contain a repetition. Therefore, $|xy| \leq p$

Pumping lemma's proof

Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA that has p states and recognizes A. Let $s = s_1 s_2 \ldots s_n$ be a string over Σ of length $n \geq p$. Let $r_1, r_2, \ldots, r_{n+1}$ be the sequence of states while processing s, i.e., $r_{i+1} = \delta(r_i, s_i)$, $1 \leq i \leq n$

Pumping lemma's proof

Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA that has p states and recognizes A. Let $s = s_1 s_2 \ldots s_n$ be a string over Σ of length $n \geq p$. Let $r_1, r_2, \ldots, r_{n+1}$ be the sequence of states while processing s, i.e., $r_{i+1} = \delta(r_i, s_i)$, $1 \leq i \leq n$

- $n + 1 \geq p + 1$ and among the first $p + 1$ elements in $r_1, r_2, \ldots, r_{n+1}$ two must be the same state, say $r_j = r_k$.

Pumping lemma's proof

Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA that has p states and recognizes A. Let $s = s_1 s_2 \ldots s_n$ be a string over Σ of length $n \geq p$. Let $r_1, r_2, \ldots, r_{n+1}$ be the sequence of states while processing s, i.e., $r_{i+1} = \delta(r_i, s_i)$, $1 \leq i \leq n$

- $n + 1 \geq p + 1$ and among the first $p + 1$ elements in $r_1, r_2, \ldots, r_{n+1}$ two must be the same state, say $r_j = r_k$.
- Because r_k occurs among the first $p + 1$ places in the sequence starting at r_1, we have $k \leq p + 1$

Pumping lemma's proof

Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA that has p states and recognizes A. Let $s = s_1 s_2 \ldots s_n$ be a string over Σ of length $n \geq p$. Let $r_1, r_2, \ldots, r_{n+1}$ be the sequence of states while processing s, i.e., $r_{i+1} = \delta(r_i, s_i)$, $1 \leq i \leq n$

- $n + 1 \geq p + 1$ and among the first $p + 1$ elements in $r_1, r_2, \ldots, r_{n+1}$ two must be the same state, say $r_j = r_k$.
- Because r_k occurs among the first $p + 1$ places in the sequence starting at r_1, we have $k \leq p + 1$
- Now let $x = s_1 \ldots s_{j-1}$, $y = s_j \ldots s_{k-1}$, $z = s_k \ldots s_n$.

Note

- As x takes M from r_1 to r_j, y takes M from r_j to r_j, and z takes M from r_j to r_{n+1}, which is an accept state, M must accept xy^iz, for $i \geq 0$

Note

- As x takes M from r_1 to r_j, y takes M from r_j to r_j, and z takes M from r_j to r_{n+1}, which is an accept state, M must accept xy^iz, for $i \geq 0$
- We know that $j \neq k$, so $|y| > 0$;

Note

- As x takes M from r_1 to r_j, y takes M from r_j to r_j, and z takes M from r_j to r_{n+1}, which is an accept state, M must accept xy^iz, for $i \geq 0$
- We know that $j \neq k$, so $|y| > 0$;
- We also know that $k \leq p + 1$, so $|xy| \leq p$

Note

- As x takes M from r_1 to r_j, y takes M from r_j to r_j, and z takes M from r_j to r_{n+1}, which is an accept state, M must accept xy^iz, for $i \geq 0$
- We know that $j \neq k$, so $|y| > 0$;
- We also know that $k \leq p+1$, so $|xy| \leq p$

Thus, all conditions are satisfi ed and lemma is proven

Before using lemma

Note: To use this lemma we must also ensure that if the property stated by the pumping lemma is true then the language is regular.

Before using lemma

Note: To use this lemma we must also ensure that if the property stated by the pumping lemma is true then the language is regular.

Proof: assuming that each element of language L satisfi es the three conditions stated in pumping lemma we can easily construct a FA that recognizes L, that is, L is regular.

Before using lemma

Note: To use this lemma we must also ensure that if the property stated by the pumping lemma is true then the language is regular.

Proof: assuming that each element of language L satisfi es the three conditions stated in pumping lemma we can easily construct a FA that recognizes L, that is, L is regular.

Note: if only some elements of L satisfy the three conditions it does not mean that L is regular.

Using pumping lemma (PL)

Proving that a language A is not regular using PL:

Using pumping lemma (PL)

Proving that a language A is not regular using PL:

1. Assume that A is regular in order to obtain a contradiction

Using pumping lemma (PL)

Proving that a language A is not regular using PL:

1. Assume that A is regular in order to obtain a contradiction
2. The pumping lemma guarantees the existence of a pumping length p s.t. all strings of length p or greater in A can be pumped

Using pumping lemma (PL)

Proving that a language A is not regular using PL:

1. Assume that A is regular in order to obtain a contradiction
2. The pumping lemma guarantees the existence of a pumping length p s.t. all strings of length p or greater in A can be pumped
3. Find $s \in A$, $|s| \geq p$, that cannot be pumped: demonstrate that s cannot be pumped by considering all ways of dividing s into x,y,z, showing that for each division one of the pumping lemma conditions, (1) $xy^iz \in A$, (2) $|y| > 0$, (3) $|xy| \leq p$, fails.

Observations

- The existence of s contradicts pumping lemma, hence A cannot be regular

Observations

- The existence of s contradicts pumping lemma, hence A cannot be regular
- Finding s sometimes takes a bit of creative thinking. Experimentation is suggested

Applications

Example 1: prove that $B = \{0^n1^n | n \geq 0\}$ is not regular

Applications

Example 1: prove that $B = \{0^n 1^n | n \geq 0\}$ is not regular

Assume that B is regular and let p be the pumping length of B. Choose $s = 0^p 1^p \in B$; obviously $|0^p 1^p| > p$. By pumping lemma $s = xyz$ such that for any $i \geq 0$, $xy^i z \in B$

Example, continuation

Consider the cases:

Example, continuation

Consider the cases:

1. y consists of 0s only. In this case $xyyz$ has more 0s than 1s and so it is not in B, violating condition 1

Example, continuation

Consider the cases:

1. y consists of 0s only. In this case $xyyz$ has more 0s than 1s and so it is not in B, violating condition 1
2. y consists of 1s only. This leads to the same contradiction

Example, continuation

Consider the cases:

1. y consists of 0s only. In this case $xyyz$ has more 0s than 1s and so it is not in B, violating condition 1
2. y consists of 1s only. This leads to the same contradiction
3. y consists of 0s and 1s. In this case $xyyz$ may have the same number of 0s and 1s but they are out of order with some 1s before some 0s hence it cannot be in B either

Example, continuation

Consider the cases:

1. y consists of 0s only. In this case $xyyz$ has more 0s than 1s and so it is not in B, violating condition 1
2. y consists of 1s only. This leads to the same contradiction
3. y consists of 0s and 1s. In this case $xyyz$ may have the same number of 0s and 1s but they are out of order with some 1s before some 0s hence it cannot be in B either

The contradiction is unavoidable if we make the assumption that B is regular so B is not regular

Example 2

Prove that $C = \{w \mid w \text{ has an equal number of 0s and 1s}\}$ is not regular

Example 2

Prove that $C = \{w \mid w \text{ has an equal number of 0s and 1s}\}$ is not regular

Proof: assume that C is regular and p is its pumping length.

Let $s = 0^p 1^p$ with $s \in C$. Then pumping lemma guarantees that $s = xyz$, where $xy^i z \in C$ for any $i \geq 0$.

Note

If we take the division $x = z = \epsilon$, $y = 0^p1^p$ it seems that indeed, no contradiction occurs. However:

Note

If we take the division $x = z = \epsilon$, $y = 0^p1^p$ it seems that indeed, no contradiction occurs. However:

- Condition 3 states that $|xy| \leq p$, and in our case $xy = 0^p1^p$ and $|xy| > p$. Hence, 0^p1^p cannot be pumped

Note

If we take the division $x = z = \epsilon$, $y = 0^p 1^p$ it seems that indeed, no contradiction occurs. However:

- Condition 3 states that $|xy| \leq p$, and in our case $xy = 0^p 1^p$ and $|xy| > p$. Hence, $0^p 1^p$ cannot be pumped
- If $|xy| \leq p$ then y must consists of only 1s, so $xyyz \notin C$ because there are more 1-s than 0-s.

Note

If we take the division $x = z = \epsilon$, $y = 0^p 1^p$ it seems that indeed, no contradiction occurs. However:

- Condition 3 states that $|xy| \leq p$, and in our case $xy = 0^p 1^p$ and $|xy| > p$. Hence, $0^p 1^p$ cannot be pumped
- If $|xy| \leq p$ then y must consists of only 1s, so $xyyz \notin C$ because there are more 1-s than 0-s.

This gives us the desired contradiction

Other selections

Selecting $s = (01)^p$ leads us to trouble because this string can be pumped by the division: $x = \epsilon$, $y = 01$, $z = (01)^{p-1}$.

Then $xy^iz \in C$ for any $i \geq 0$

An alternative method

Use the fact that B is nonregular.

An alternative method

Use the fact that B is nonregular.

- If C were regular then $C \cap 0^*1^*$ would also be regular because 0^*1^* is regular and $\cap$ of regular languages is a regular language.

An alternative method

Use the fact that B is nonregular.

- If C were regular then $C \cap 0^*1^*$ would also be regular because 0^*1^* is regular and $\cap$ of regular languages is a regular language.
- But $C \cap 0^*1^* = \{0^n 1^n | n \geq 0\}$ which is not regular.

An alternative method

Use the fact that B is nonregular.

- If C were regular then $C \cap 0^*1^*$ would also be regular because 0^*1^* is regular and $\cap$ of regular languages is a regular language.
- But $C \cap 0^*1^* = \{0^n1^n | n \geq 0\}$ which is not regular.
- Hence, C is not regular either.

Example 3

Show that $F = \{ww | w \in \{0,1\}^*\}$ is nonregular using pumping lemma

Example 3

Show that $F = \{ww | w \in \{0,1\}^*\}$ is nonregular using pumping lemma

Proof: Assume that F is regular and p is its pumping length.

Consider $s = 0^p10^p1 \in F$. Since $|s| > p$, $s = xyz$ and satisfi esthe conditions of the pumping lemma.

Note

- Condition 3 is again crucial because without it we could pump s if we let $x = z = \epsilon$, so $xyyz \in F$

Note

- Condition 3 is again crucial because without it we could pump s if we let $x = z = \epsilon$, so $xyyz \in F$
- The string $s = 0^p10^p1$ exhibits the essence of the nonregularity of F.

Note

- Condition 3 is again crucial because without it we could pump s if we let $x = z = \epsilon$, so $xyyz \in F$
- The string $s = 0^p10^p1$ exhibits the essence of the nonregularity of F.
- If we chose, say $0^p0^p \in F$ we fail because this string can be pumped

Example 4

Show that $D = \{1^{n^2} | n \geq 0\}$ is nonregular.

Example 4

Show that $D = \{1^{n^2} | n \geq 0\}$ is nonregular.

Proof by contradiction: Assume that D is regular and let p be its pumping length. Consider $s = 1^{p^2} \in D$, $|s| \geq p$. Pumping lemma guarantees that s can be split, $s = xyz$, where for all $i \geq 0$, $xy^iz \in D$

Searching for a contradiction

The elements of D are strings whose lengths are perfect squares. Looking at first perfect squares we observe that they are: 0, 1, 4, 9, 25, 36, 49, 64, 81, ...

Searching for a contradiction

The elements of D are strings whose lengths are perfect squares. Looking at first perfect squares we observe that they are: 0, 1, 4, 9, 25, 36, 49, 64, 81, ...

- Note the growing gap between these numbers: large members cannot be near each other

Searching for a contradiction

The elements of D are strings whose lengths are perfect squares. Looking at fi rst perfect squareswe observe that they are: 0, 1, 4, 9, 25, 36, 49, 64, 81, $\ldots$

- Note the growing gap between these numbers: large members cannot be near each other
- Consider two strings $xy^i z$ and $xy^{i+1} z$ which differ from each other by a single repetition of y.

Searching for a contradiction

The elements of D are strings whose lengths are perfect squares. Looking at fi rst perfect squareswe observe that they are: 0, 1, 4, 9, 25, 36, 49, 64, 81, ...

- Note the growing gap between these numbers: large members cannot be near each other
- Consider two strings xy^iz and $xy^{i+1}z$ which differ from each other by a single repetition of y.
- If we chose i very large the lengths of xy^iz and $xy^{i+1}z$ cannot be both perfect square because they are too close to each other.

Turning this idea into a proof

Calculate the value of i that gives us the contradiction.

Turning this idea into a proof

Calculate the value of i that gives us the contradiction.

- If $m = n^2$, calculating the difference we obtain $(n+1)^2 - n^2 = 2n + 1 = 2\sqrt{m} + 1$

Turning this idea into a proof

Calculate the value of i that gives us the contradiction.

- If $m = n^2$, calculating the difference we obtain $(n+1)^2 - n^2 = 2n + 1 = 2\sqrt{m} + 1$
- By pumping lemma $|xy^iz|$ and $|xy^{i+1}z|$ are both perfect squares. But letting $|xy^iz| = m$ we can see that they cannot be both perfect square if $|y| < 2\sqrt{|xy^iz|} + 1$, because they would be too close together.

Value of i for contradiction

To calculate the value for i that leads to contradiction we observe that:

Value of i for contradiction

To calculate the value for i that leads to contradiction we observe that:

- $|y| \leq |s| = p^2$

Value of i for contradiction

To calculate the value for i that leads to contradiction we observe that:

- $|y| \leq |s| = p^2$
- Let $i = p^4$. Then
  $|y| \leq p^2 = \sqrt{p^4} < 2\sqrt{p^4} + 1 \leq 2\sqrt{|xy^iz|} + 1$

Example 5

Sometimes “pumping down" is useful when we apply pumping lemma.

Example 5

Sometimes “pumping down" is useful when we apply pumping lemma.

- We illustrate this using pumping lemma to prove that $E = \{0^i 1^j | i > j\}$ is not regular

Example 5

Sometimes “pumping down" is useful when we apply pumping lemma.

- We illustrate this using pumping lemma to prove that $E = \{0^i 1^j | i > j\}$ is not regular
- Proof: by contradiction using pumping lemma. Assume that E is regular and its pumping length is p.

Searching for a contradiction

- Let $s = 0^{p+1}1^p$; From decomposition $s = xyz$, from condition 3, $|xy| \leq p$ it results that y consists only of 0s.

Searching for a contradiction

- Let $s = 0^{p+1}1^p$; From decomposition $s = xyz$, from condition 3, $|xy| \leq p$ it results that y consists only of 0s.
- Let us examine $xyyz$ to see if it is in E. Adding an extra-copy of y increases the number of zeros. Since E contains all strings 0^*1^* that have more 0s than 1s, it will still give a string in E

Searching for a contradiction

- Let $s = 0^{p+1}1^p$; From decomposition $s = xyz$, from condition 3, $|xy| \leq p$ it results that y consists only of 0s.
- Let us examine $xyyz$ to see if it is in E. Adding an extra-copy of y increases the number of zeros. Since E contains all strings 0^*1^* that have more 0s than 1s, it will still give a string in E

Try something else

- Since $xy^iz \in E$ even when $i = 0$, consider $i = 0$ and $xy^0z = xz \in E$.

Try something else

- Since $xy^iz \in E$ even when $i = 0$, consider $i = 0$ and $xy^0z = xz \in E$.
- This decreases the number of 0s in s.

Try something else

- Since $xy^i z \in E$ even when $i = 0$, consider $i = 0$ and $xy^0 z = xz \in E$.
- This decreases the number of 0s in s.
- Since s has just one more 0 than 1 and xz cannot have more 0s than 1s,
  ($xyz = 0^{p+1}1^p$ and $|y| \neq 0$)
  xz cannot be in E.

Try something else

- Since $xy^iz \in E$ even when $i = 0$, consider $i = 0$ and $xy^0z = xz \in E$.
- This decreases the number of 0s in s.
- Since s has just one more 0 than 1 and xz cannot have more 0s than 1s,

  ($xyz = 0^{p+1}1^p$ and $|y| \neq 0$)

  xz cannot be in E.

This is the required contradiction

Minimum pumping length

- The pumping lemma says that every regular language has a pumping length p, such that every string in the language of length at least p can be pumped.

Minimum pumping length

- The pumping lemma says that every regular language has a pumping length p, such that every string in the language of length at least p can be pumped.
- Hence, if p is a pumping length for a regular language A so is any length $p' \geq p$.

Minimum pumping length

- The pumping lemma says that every regular language has a pumping length p, such that every string in the language of length at least p can be pumped.
- Hence, if p is a pumping length for a regular language A so is any length $p' \geq p$.
- The minimum pumping length for A is the smallest p that is a pumping length for A.

Example

Consider $A = 01^*$. The minimum pumping length for A is 2.

Example

Consider $A = 01^*$. The minimum pumping length for A is 2.

Reason: the string $s = 0 \in A$, $|s| = 1$ and s cannot be pumped. But any string $s \in A$, $|s| \geq 2$ can be pumped because for $s = xyz$ where $x = 0$, $y = 1$, $z = rest$ and $xy^iz \in A$. Hence, the minimum pumping length for A is 2.

Problem 1

Find the minimum pumping length for the language 0001^*.

Problem 1

Find the minimum pumping length for the language 0001^*.

Solution: The minimum pumping length for 0001^* is 4.

Reason: $000 \in 0001^*$ but 000 cannot be pumped. Hence, 3 is not a pumping length for 0001^*. If $s \in 0001^*$ and $|s| \geq 4$ s can be pumped by the division $s = xyz$, $x = 000$, $y = 1$, $z = rest$.

Problem 2

Find the minimum pumping length for the language 0^*1^*.

Problem 2

Find the minimum pumping length for the language 0^*1^*.

Solution: The minimum pumping length of 0^*1^* is 1.

Problem 2

Find the minimum pumping length for the language 0^*1^*.

Solution: The minimum pumping length of 0^*1^* is 1.

Reason: the minimum pumping length for 0^*1^* cannot be 0 because ϵ is in the language but cannot be pumped. Every nonempty string $s \in 0^*1^*$, $|s| \geq 1$ can be pumped by the division: $s = xyz$, $x = \epsilon$, y first character of s and z the rest of s.

Problem 3

Find the minimum pumping length for the language $0^*1^+0^+1^* \cup 10^*1$.

Problem 3

Find the minimum pumping length for the language $0^*1^+0^+1^* \cup 10^*1$.

Solution: The minimum pumping length for $0^*1^+0^+1^* \cup 10^*1$ is 3.

Problem 3

Find the minimum pumping length for the language $0^*1^+0^+1^* \cup 10^*1$.

Solution: The minimum pumping length for $0^*1^+0^+1^* \cup 10^*1$ is 3.

Reason: The pumping length cannot be 2 because the string 11 is in the language and it cannot be pumped. Let s be a string in the language of length at least 3. If s is generated by $0^*1^+0^+1^*$ we can write is as $s = xyz$, $x = \epsilon$, y is the first symbol of s, and z is the rest of the string. If s is generated by 10^*1 we can write it as $s = xyz$, $x = 1$, $y = 0$ and z is the remainder of s.