Identifying Nonregular Languages

Raju Hazari

Department of Computer Science and Engineering
National Institute of Technology Calicut

October 22, 2020

Nonregular Languages

- The class of languages known as the regular languages has at least four different descriptions.
  - They are the languages accepted by DFA's, by NFA's, and by ϵ-NFA's; they are also the languages defined by regular expressions.
- Not every language is a regular language.
- A powerful technique, known as the "**pumping lemma**" is very useful for showing certain languages not to be regular.

Nonregular Languages

- Is the language $L = \{a^n b^n : n \geq 0\}$ regular ?

Nonregular Languages

- Is the language $L = \{a^n b^n : n \geq 0\}$ regular ?
- No.

Nonregular Languages

- Is the language $L = \{a^n b^n : n \geq 0\}$ regular ?
- No. **Why?**

Nonregular Languages

- Is the language $L = \{a^n b^n : n \geq 0\}$ regular ?
- No. **Why?**
  - Intuitively, the argument is, finite state automaton can not remember everything. If it has to accept $a^n b^n$, it has to remember every n. It is not possible with finite amount memory.
  - So, if it has accept $a^n b^n$, it has to remember that after reading the entire sequence of a how many $a's$ it has read. This is not possible by finite state automata.
  - But formally, using the **pumping lemma**, we can prove this.

Pumping Lemma

- **Theorem:** (*The pumping lemma for regular languages*)

  Let L be a regular language. Then there exists a constant n such that for every string w in L with $|w| \geq n$ can be decomposed as $w = xyz$, such that:

  - $|xy| \leq n$,
  - $|y| \geq 1$,
  - For all $k \geq 0$, the string xy^kz is also in L.

  That is, we can always find a nonempty string y not too far from the beginning of w that can be "pumped"; that is, repeating y any number of times, or deleting it (the case $k = 0$), keeps the resulting string in the language L.

Pumping Lemma

- **Theorem:** (*The pumping lemma for regular languages*)

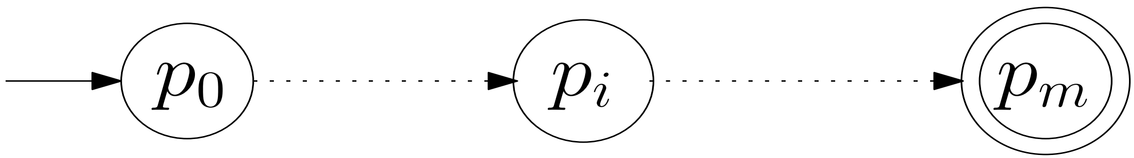
  Let L be a regular language. Then there exists a constant n such that for every string w in L with $|w| \geq n$ can be decomposed as $w = xyz$, such that:
  - $|xy| \leq n$,
  - $|y| \geq 1$,
  - For all $k \geq 0$, the string xy^kz is also in L.

  That is, we can always find a nonempty string y not too far from the beginning of w that can be "pumped"; that is, repeating y any number of times, or deleting it (the case $k = 0$), keeps the resulting string in the language L.

- **Proof:** Suppose L is regular. Then $L = L(A)$ for some DFA A. Suppose A has n states.
  Now, consider any string w of length n or more, say
  $w = a_1 a_2 \cdots a_m$, where $m \geq n$ and each a_i is an input symbol.

Pumping Lemma

For $i = 0, 1, \cdots, n$ define state p_i to be $\hat{\delta}(q_0, a_1 a_2 \cdots a_i)$, where δ is the transition function of A, and q_0 is the start state of A. That is, p_i is the state A is in after reading the first i symbols of w. Note that, $p_0 = q_0$.

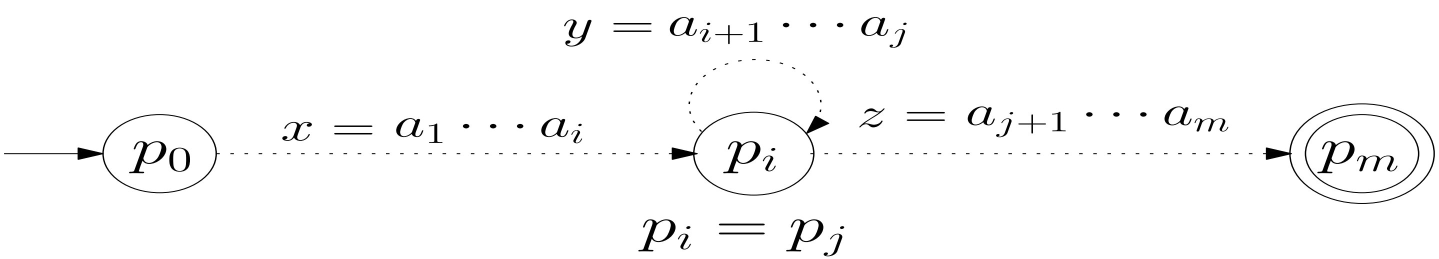


By the **pigeonhole principle** (If we put n objects into m boxes and if $n > m$, then at least one box must have more than one item in it), it is not possible for $n + 1$ different p_i's for $i = 0, 1, \cdots, n$ to be distinct, since there are only n different states. Thus, we can find two different integers i and j, with $0 \leq i < j \leq n$, such that $p_i = p_j$. Now, we can break $w = xyz$ as follows:

$x = a_1 a_2 \cdots a_i, \qquad y = a_{i+1} a_{i+2} \cdots a_j \qquad$ and $z = a_{j+1} a_{j+2} \cdots a_m$

Pumping Lemma

That is, x takes us to p_i once; y takes us from p_i back to p_i (since $p_i = p_j$), and z is the balance of w. Note that x may be empty, in the case that $i = 0$. Also, z may be empty if $j = n = m$. However, y can not be empty, since i is strictly less than j.



Now, consider what happens if the automaton A receives the input xy^kz for any $k \geq 0$.
If $k = 0$, then the automaton goes from the start state p_0 to p_i on input x. Since $p_i = p_j$, it must be that A goes from p_i to the accepting state on input z. Thus, A accepts xz.
If $k > 0$, then A goes from p_0 to p_i on the input x, circles from p_i to p_i k times on the input y^k, and then goes to the accepting state on input z.
Thus, for any $k \geq 0$, xy^kz is also accepted by A; that is, xy^kz is in L.

Pumping Lemma

- **Example 1:** Using the pumping lemma, show that $L = \{a^n b^n : n \geq 0\}$ is not regular.

- Assume that L is regular, so that the pumping lemma must hold. Now using the pumping lemma, this $a^n b^n$ can be written in the form xyz, where $|xy| \leq n$ and $|y| \geq 1$. Suppose something like this
$$\underbrace{a \cdots a}_{x}\ \underbrace{a \cdots a}_{y}\ \underbrace{a \cdots ab \cdots b}_{z}$$
$|xy| \leq n$ means the substring y must consist entirely of a's. Suppose $|y| = p$. Then the string obtained by using $k = 0$ is $a^{n-p} b^n$ and is clearly not in L. This contradicts the pumping lemma and thereby indicates that the assumption that L is regular must be false.

Pumping Lemma

- **Example 2:** Using the pumping lemma, show that $L = \{ww^R : w \in \Sigma^*\}$ is not regular.

- Assume that L is regular, so that the pumping lemma must hold. Now using the pumping lemma, this ww^R can be written in the form xyz, where $|xy| \le n$ and $|y| \ge 1$. Suppose something like this

$$\overbrace{a\cdots a\cdots a\cdots a}^{a^n}\ \overbrace{b\cdots\cdots\cdots b}^{b^n}\ \overbrace{b\cdots\cdots\cdots b}^{b^n}\ \overbrace{a\cdots\cdots\cdots a}^{a^n}$$

$$x \qquad y \qquad\qquad\qquad\qquad z$$

$|xy| \le n$ means choosing a y that consists entirely of a's. Now consider $k = 0$. The string obtained in this fashion has fewer a's on the left than on the right and so cannot be of the form ww^R. Therefore, L is not regular.

Pumping Lemma

- Pumping lemma says that L is regular, then pumping lemma holds.
  - L is regular $\Rightarrow$ pumping lemma holds
- The argument the way we have done is: if pumping lemma does not holds, L is not regular.

Pumping Lemma

- Pumping lemma says that L is regular, then pumping lemma holds.
  - L is regular $\Rightarrow$ pumping lemma holds
- The argument the way we have done is: if pumping lemma does not holds, L is not regular.
- What is a converse to pumping lemma ?

Pumping Lemma

- Pumping lemma says that L is regular, then pumping lemma holds.
  - L is regular $\Rightarrow$ pumping lemma holds
- The argument the way we have done is: if pumping lemma does not holds, L is not regular.
- What is a converse to pumping lemma ?
  - Let L is some language. Then if there is a constant n such that for any string $|w| \geq n$. We can write w in the form xyz such that xy^kz belongs to L for all k, then L is a regular language.

Pumping Lemma

- Pumping lemma says that L is regular, then pumping lemma holds.
  - L is regular $\Rightarrow$ pumping lemma holds
- The argument the way we have done is: if pumping lemma does not holds, L is not regular.
- What is a converse to pumping lemma ?
  - Let L is some language. Then if there is a constant n such that for any string $|w| \geq n$. We can write w in the form xyz such that $xy^k z$ belongs to L for all k, then L is a regular language.
- Is is true ?

Pumping Lemma

- Pumping lemma says that L is regular, then pumping lemma holds.
  - L is regular $\Rightarrow$ pumping lemma holds
- The argument the way we have done is: if pumping lemma does not holds, L is not regular.
- What is a converse to pumping lemma ?
  - Let L is some language. Then if there is a constant n such that for any string $|w| \geq n$. We can write w in the form xyz such that xy^kz belongs to L for all k, then L is a regular language.
- Is is true ?
  - **Converse is not true.** That is, the pumping lemma may hold, L may not be regular.
  - If the pumping lemma does not hold L is not regular, but if the pumping lemma holds you cannot conclude L is regular, L may be regular or not regular.

Pumping Lemma

- We must realize that what pumping lemma says:
  - If we have a string whose length is sufficiently large, then the string w can be written in the form $w = xyz$ such that $xy^k z \in L$ where $k = 0, 1, \cdots$
  - So, we will get an infinite number of strings which belong to L. It does not means that if we have sufficiently large string belonging to L, we can write it in the form $xy^k z$ for large k. **That is not true**.
  - What pumping lemma says is if we have a fairly large string, then we can write it in the form $w = xyz$ and we can get infinite number of string, it does not means that if we have a very large string we can write it in the form $xy^k z$ for large k.
  - For example, if we consider Σ^* where $\Sigma = \{a, b\}$. We know that there are large string which are cube free. We have seen that there are large strings which are cube free, that is no sub string will occur three times consecutively.