Pumping Lemma Handout

Contents

1 Regular Languages

1.1 Overview and Statement of the Lemma

The pumping lemma states the "pumping" property of regular languages, which is that, if L is a regular language, every $w \in L$ which is suffiently long has a "pumpable" substring, that is, a substring which can be either deleted ("pumped down") or repeated any number of times ("pumped up") without leaving L. Since this property holds for every language, it can be used to prove a language is not regular by proving (usually by contradiction) that the language does not satify the pumping property.

The formal statement of the pumping lemma follows.

Lemma 1 (Pumping Lemma) *For any regular language L, there exists a positive integer p, (which is called a pumping length of L) such that for any $w \in L$ of length at least p, there exist strings x, y, z such that the following four conditions hold:*

1. $w = xyz$,
2. the length of xy is no greater than p, 3. the string y is not empty, 4. for any integer $i \geq 0$, $xy^iz \in L$.

1.2 Using the Pumping Lemma to Prove that a Language is not Regular

Let $L = \{a^n b^n : n \geq 0\}$.

Theorem 1 *L is not regular.*

Proof: By contradiction. Assume L is regular. Let p be a pumping length of L. Let $w = a^p b^p$, which is a member of L. There exist strings x, y, z such that the four conditions given in the pumping lemma hold. Let k be the length of xy, and let ℓ be the length of y. Since xy is a prefix of w of length no more than p and y is not empty, $xy = a^k$ and hence $y = a^\ell$, where $\ell \geq 1$. Let $i = 0$. Then $xy^iz = xz = a^{p-\ell}b^p$ which is not in L since $p - \ell \neq p$, contradiction. We conclude that L is not regular. ∎

2 Context-Free Languages

2.1 Overview and Statement of the Lemma

The pumping lemma for context-free languages states a "pumping" property of context-free languages The lemma is very similar in form to the pumping lemma for regular languages, and can similarly be used to prove that certain languages are not context-free.

Lemma 2 (Context-Free Pumping Lemma) *For any context-free language L, there exists a positive integer p, (which is called a pumping length of L) such that for any $w \in L$ of length at least p, there exist strings u, v, x, y, z such that the following four conditions hold:*

1. $w = uvxyz$,
2. the length of vxy is no greater than p, 3. the strings v and y are not both empty, 4. for any integer $i \geq 0$, $uv^ixy^iz \in L$.

2.2 Using the Lemma to Prove that a Language is not Context-Free

Let $L = \{a^nb^nc^n : n \geq 0\}$.

Theorem 2 *L is not context-free.*

Proof: By contradiction. Assume L is context-free. Let p be a pumping length of L. Let $w = a^pb^pc^p$, which is a member of L. By the context-free pumping lemma, there exist strings u, v, x, y, z such that the four conditions given in the pumping hold. Then $vxy \in L$. Let k be the length of vxy, let ℓ be the length of v, and let m be the length of y. vxy is a substring of w of length less than p.

Claim: vxy either contains no a or no c.

Proof of Claim: By contradiction. Suppose vxy contains at least one a and at least one c. The smallest substring of w which contains both an a and a c is ab^pc, which has length $p + 2$, Thus the length of vxy is at least $p + 2$, contradiction.

Case 1: vxy contains no a. Then neither v nor y contains an a. Let $i = 0$. We have $uv^0xy^0z = uxz \in L$. The length of uxz cannot be more than $3p - \ell - m < 3p$, and uxz contains the string a^p, contradiction.

Case 2: vxy contains no c. This case is similar to Case 1, and we also reach a contradiction.

We conclude that L is not context-free. ∎

3 Proofs of the Pumping Lemmas

3.1 Proof of the Pumping Lemma for Regular Languages

Let L be a regular language, and let M be a DFA which accepts L. Let p be the number of states of L. We will prove that p is a pumping length for L.

Let $w \in L$ have length $n \geq p$, and let a_i be the i^{th} symbol of w. Let $q_0, q_1, \ldots q_n$ be the sequence of states of the computation of M which accepts w. That is, $\delta(q_{i-1}, a_i) = q_i$ for all i, where δ is the transition function of M, and q_n is a final state.

The sequence $\sigma = q_0, q_1, \ldots q_p$ has length $p + 1$. Since M has p states, σ contain a duplication. Pick k, k such that $0 \leq i < k \leq p$ and $q_i = q_k$. Let $x = a_1 \ldots a_j$, $y = a_{j+1} \ldots a_k$, and $z = a_{k+1} \ldots q_n$. Note that:

$xyz = w$
$|xy| = k \leq p$
y cannot be empty since $|y| = k - j > 0$
$\delta(q_0, x) = q_j$
$\delta(q_j, y) = q_k$
$\delta(q_k, z) = q_n$
For any $i \geq 0$, $q_0 \ldots q_j (q_{j+1} \ldots q_k)^i q^{k+1} \ldots q_n$ is an accepting computation of M with input $xy^i z$, since $\delta(q_j, y) = q_k = q_j$. Thus, $xy^i z \in L$.