Computer Science 351

Nonregular Languages, Part One

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #8

Goal for Today

Goals for Today:

- Introduction of a theorem called the ***Pumping Lemma for Regular Languages***
- Describe how to use this to prove that a language $L \subseteq \Sigma^\star$ is ***not*** regular!

Pumping Lemma for Regular Languages

Pumping Lemma: Let Σ be an alphabet and let $A \subseteq \Sigma^\star$.

If A is a regular language, then there is a number $p \geq 1$ (called the ***pumping length*** for A) — which only depends on A — such that if s is any string in A with length at least p, then s can be divided into three pieces $s = xyz$ (for $x, y, z \in \Sigma^\star$), satisfying the following three conditions.

1. $xy^i z \in A$ for every integer i such that $i \geq 0$.
2. $|y| > 0$ (so that $y \neq \lambda$).
3. $|xy| \leq p$.

Note: y^i is the concatenation of i copies of the string y.

Pumping Lemma for Regular Languages

Proof: Let $A \subseteq \Sigma^\star$ be a regular language.

- Then there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ with language A.
- Let $p = |Q|$, the number of states in M — so that $p \geq 1$ is a number that depends on A (but nothing else).
- Either A does not include any strings $s \in \Sigma^\star$ with length at least p, or A includes at least one such string.
- These cases are considered separately, next.

Pumping Lemma for Regular Languages

Case: A does not include any strings $s \in \Sigma^\star$ with length at least p.

- In this case there is nothing more we need to prove — because the claim only said something about strings $s \in A$ with length at least p.

Pumping Lemma for Regular Languages

Case: A includes at least one string $s \in \Sigma^\star$ with length at least p.

- Let s be some string in $\Sigma^\star$ such that $s \in A$ and s has length at least p. It is necessary (and sufficient) to show that it is possible to write s as xyz (for $x, y, z \in \Sigma^\star$) such that
  1. $xy^i z \in A$ for every integer $i \geq 0$.
  2. $|y| > 0$ (so $y \neq \lambda$).
  3. $|xy| \leq p$.

Pumping Lemma for Regular Languages

- Let $m = |s|$, so that $m \geq p$, and suppose that

$$s = \sigma_1 \sigma_2 \ldots, \sigma_m \in \Sigma^*.$$

- Let $r_0, r_1, \ldots, r_m$ be the sequence of states visited as s is processed — so that $r_0 = q_0 = \delta^\star(q_0, \lambda)$, and

$$r_i = \delta^\star(q_0, \sigma_1 \sigma_2 \ldots \sigma_i) \qquad \text{for } 1 \leq i \leq m.$$

Pumping Lemma for Regular Languages

- In other words, r_1 is reached after processing σ_1, r_2 is reached after processing $\sigma_1\sigma_2$, and so on.
- Consider the *first* $p+1$ states in this sequence,

$$r_0, r_1, \ldots, r_p$$

  which are visited as the "prefix" $\sigma_1\sigma_2\ldots\sigma_p$ of s with length p is processed.
- ***Key Observation:*** Since $|Q| = p$ and the above sequence of states has length $p+1$, *at least* one state $\widehat{q} \in Q$ must appear ***at least twice*** in the above sequence!

Pumping Lemma for Regular Languages

- Now let $\widehat{q} \in Q$ be a state that *does* appear at least twice in the sequence $r_0, r_1, \ldots, r_p$. Suppose it appears first as "r_i" and the second time as "r_j", so that
  - i and j are integers such that $0 \leq i < j \leq p$.
  - If $x = \sigma_1 \sigma_2 \ldots \sigma_i$, the prefix of s with length i, then

    $$\delta^\star(q_0, x) = r_i = \widehat{q},$$

    because $\widehat{q}$ is the state that is reached after processing the first i symbols of s.
  - $y = \sigma_{i+1} \sigma_{i+2} \ldots \sigma_j$, the string consisting of the next $j - i$ symbols in s, then

    $$\delta^\star(\widehat{q}, y) = r_j = \widehat{q},$$

    because processing the next $j - i$ symbols takes you back to the state $r_j = \widehat{q}$ again.

Pumping Lemma for Regular Languages

- Finally, if $z = \sigma_{j+1}\sigma_{j+2}\ldots,\sigma_m$, so that $s = xyz$, then

$$\delta^\star(\widehat{q}, z) = r_F \qquad \text{for some state } r_F \in F.$$

  This is true because processing the rest of the symbols in s must take you to an accepting state, since $s \in A = L(M)$.

- Now all that is left is to check that properties 1, 2, and 3 in the claim are all satisfied when s is written (as xyz) in this way.

Pumping Lemma for Regular Languages

1. Since $\widehat{\delta^\star}(\widehat{q}, y) = \widehat{q}$, $\widehat{\delta^\star}(\widehat{q}, y^i) = \widehat{q}$ for every integer $i \geq 0$ as well: If you start from $\widehat{q}$ and process y^i, then you do so by following a loop that takes you back to $\widehat{q}$, i times.

   Now, if you start from q_0 and process xy^iz, then

   - You will reach $\widehat{q}$ after processing x, because $\widehat{\delta^\star}(q_0, x) = \widehat{q}$.
   - You end up back in state $\widehat{q}$ after processing y^i, because $\widehat{\delta^\star}(\widehat{q}, y^i) = \widehat{q}$.
   - You will end up in state r_F after processing the remaining substring z, because $\widehat{\delta^\star}(\widehat{q}, z) = r_F$.

   So $\widehat{\delta^\star}(q_0, xy^iz) = r_F \in F$. It follows that $xy^iz \in L(M) = A$. Since this is true for every integer $i \geq 0$, this establishes the first property mentioned in the claim.

Pumping Lemma for Regular Languages

2. The string y was chosen to have length $j - i > 0$, so that $y \neq \lambda$. Thus the second property, mentioned in the claim, is also satisfied.

3. The strings x and y were chosen so that xy was the prefix of s with length $j \leq p$. Thus $|xy| \leq p$, so that the third property, mentioned in the claim, is satisfied too.

Since s was an "arbitrarily chosen" string in A with length at least p, it follows that *every* such string can be written in this way — and this completes the proof of the claim. □

Note: This is essentially the same as the proof of Theorem 1.70 on pages 78–79 of the third edition of Michael Sipser's text, *Introduction to the Theory of Computation.*

Applying the Pumping Lemma

- We ***will not*** be using this theorem to try to prove that a given language A is a regular language.
- Instead, this will be used in proofs — by contradiction — to prove that various languages are *not* regular.

An Example

Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and consider the language

$$L_1 = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}$$

so that L includes strings $\lambda, \mathtt{ab}, \mathtt{aabb}, \mathtt{aaabbb}, \ldots$, but L_1 does not include $\mathtt{ba}$ or $\mathtt{abb}$.

Claim: L_1 is not a regular language.

Proof: Suppose, to obtain a contradiction, that L_1 *is* a regular language.

An Example

Then it follows by the Pumping Lemma for Regular Languages, that there is a number $p \geq 1$ such that if s is any string in L_1 with length at least p, then s can be divided in to three pieces $s = xyz$ (for $x, y, z \in \Sigma^\star$), satisfying the following three conditions.

1. $xy^i z \in L_1$ for every integer i such that $i \geq 0$.
2. $|y| > 0$ (so that $y \neq \lambda$).
3. $|xy| \leq p$.

An Example

Consider the string $s = \mathtt{a}^p\mathtt{b}^p$.

- $s \in L_1$, since $s = \mathtt{a}^n\mathtt{b}^n$ when $n = p$.
- $|s| = 2p \geq p$.

It follows that $s = xyz$ for strings $x, y, z \in \Sigma^\star$ that satisfy properties #1 – #3, as above.

- Since xy is a prefix of s with length at most p (by property #3), and the first p symbols in s are all $\mathtt{a}$'s, $xy = \mathtt{a}^k$ for some integer k such that $0 \leq k \leq p$.

  Since $s = \mathtt{a}^p\mathtt{b}^p = xyz = a^k z$, it follows that $z = a^{p-k}b^p$.

An Example

- By property #2, $|y| > 0$ so that $y \neq \lambda$. Thus (since $|xy| = |x| + |y| = k$), $|x| = h$ and $|y| = \ell$ for integers h and ℓ such that $h \geq 0$, $\ell \geq 1$, and $h + \ell = k$.

  Since $xy = \mathrm{a}^k$ it now follows that $x = \mathrm{a}^h$ and $y = \mathrm{a}^\ell$.

- Let $i = 0$. Then

$$xy^i z = x\lambda z = xz = \mathrm{a}^h \mathrm{a}^{p-k} \mathrm{b}^p = \mathrm{a}^{p-\ell} \mathrm{b}^p$$

  since $h + \ell = p$. Thus $xy^i z \notin L_1$, since $\ell > 0$ (so that $p - \ell \neq p$).

- Thus property #1 is ***not*** satisfied (since $xy^i z \notin L_1$ when $i = 0$), and a ***contradiction*** has been obtained.

- It follows that L_1 is not a regular language. □

Summary of Process

In order to prove that a given language $L \subseteq \Sigma^\star$ is ***not*** a regular language, using the Pumping Lemma for Regular Languages, you should do the following things.

1. Assume — to obtain a contradiction — that L *is* a regular language.

   Note: It is a "courtesy to the reader" to *say* that you are giving a proof by contradiction, at the beginning of your proof.

Summary of Process

2. Note that it follows, by the Pumping Lemma for Regular Languages, that there exists a number $p \geq 1$ such that if s is any string in L with length at least p, then s can be divided in to three pieces $s = xyz$ (for $x, y, z \in \Sigma^\star$), satisfying the following three conditions.
    1. $xy^i z \in L$ for every integer i such that $i \geq 0$.
    2. $|y| > 0$ (so that $y \neq \lambda$).
    3. $|xy| \leq p$.

Summary of Process

Things to Note:

- You ***cannot*** pick and use a particular value for p, or assume anything more about its value: You would be introducing *another* contradiction if you did that, and then you could not conclude that the *original* assumption was incorrect, once a contradiction was obtained.
- It is advisable to *state* the properties that follows, by the Pumping Lemma, in your written proof — because you will need to refer to all of them, later on in the proof. Students who leave them out generally write proofs that are hard to read and they often make mistakes, because they get the properties (that follow from the Pumping Lemma) mixed up when they need to use them.

Summary of Process

3. Describe a string s — that will generally depend on the value p that has just been chosen. Give a (generally short) proof that $s \in L$ and $|s| \geq p$.

   Note: You ***do*** get to pick the string s that will be used for the rest of this proof!

   However, this is generally the trickiest part of this process! Some choices of s will make the rest of this easy. Other choices will make it impossible to correctly complete the proof.

Summary of Process

4. Observe that it now follows that there exist strings $x, y, z \in \Sigma^\star$ such that $s = xyz$ and properties #1, #2, and #3 (listed in the "Pumping Lemma for Regular Languages") hold.

   Note: Once again, you ***do not*** get to choose the strings x, y and z to work with — you need to prove that there is a problem with ***all*** possible choices of these strings.

   However, if the string s has been chosen well (and the language L is really ***not*** regular) then it should be possible to show that if x and y are ***any*** initial strings of s such that properties #2 and #3 hold, then $xy^i z \notin L$ for ***some*** nonnegative integer i.

Summary of Process

5. If a contradiction really *is* obtained, without assuming anything more than what has been noted above, then you can (and should) ***conclude*** that the language L is not regular, as claimed.

Expectations

- The *next* lecture will provide another technique that can be used to prove that various languages are not regular — which is probably easier to use, and probably more important, than this one.
- There will be at least one problem on an upcoming tutorial exercise asking students to use the Pumping Lemma for Regular Languages to prove that a given language is not regular. Given sufficient time and hints, students may also be asked to use the Pumping Lemma for Regular Languages to prove that a language is not a regular on a test.