Computer Science 313

Nonregular Languages, Part One

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #10

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!

Reference:

- *Introduction to the Theory of Computation,* Section 1.4

Pumping Lemma for Regular Languages

Pumping Lemma: If $A \subseteq \Sigma^\star$ 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 $i \geq 0$.
2. $|y| > 0$ (so $y \neq \lambda$).
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. $|xy| \leq p$.
- 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

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

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 *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.
- However, we ***will*** use the next result, which is a consequence (or "corollary") of it: It follows because if $P \Rightarrow Q$ for two logical statements P and Q, then $\neg Q \Rightarrow \neg P$ as well.
- In particular, the corollary will be used to prove that a given language A is ***not*** a regular language.

Applying the Pumping Lemma

Corollary of Pumping Lemma: Suppose that $A \subseteq \Sigma^\star$ such that the following holds:

> For ***every*** number $p \geq 1$, ***there exists*** a string $s \in A$ whose length is at least p such that ***it is impossible*** to write s as $s = xyz$, for ***any*** strings $x, y, z \in \Sigma^\star$, so that all three of the following properties are satisfied.
>
> 1. $xy^i z \in A$ for every integer $i \geq 0$.
> 2. $|y| > 0$ (so that $y \neq \lambda$).
> 3. $|xy| \leq p$.

Then A ***is not*** a regular language.

First Example

Suppose $\Sigma_1 = \{\mathtt{a}, \mathtt{b}\}$ and consider the language

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

so that L_1 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: The Pumping Lemma for Regular Languages will be used to prove this.

- With that noted, let p be any arbitrarily chosen integer such that $p \geq 1$.

First Example

- Consider the string $s = \mathrm{a}^p\mathrm{b}^p$.
  - $s \in L_1$, since $s = \mathrm{a}^n\mathrm{b}^n$ when $n = p$.
  - $|s| = 2p \geq p$.
- Let us ***assume*** that there exist strings $x, y, z \in \Sigma_1^\star$ such that $s = xyz$ and all three properties (included in the statement of the Pumping Lemma for Regular Languages) are satisfied for this choice of x, y and z.
- Since xy is a prefix of s with length at most p (by Property #3), and the first p symbols in s are all a's, $xy = \mathrm{a}^k$ for some integer k such that $0 \leq k \leq p$. Since $s = \mathrm{a}^p\mathrm{b}^p = xyz$ it follows that $z = a^{p-k}b^p$.

First 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 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.

Second Example

Suppose that $\Sigma_2 = \{0, 1\}$. The ***reversal*** of a string

$$\omega = \sigma_1 \sigma_2 \ldots \sigma_n$$

is the string

$$\omega^R = \sigma_n \ldots \sigma_2 \sigma_1$$

obtained by listing the symbols in ω in reverse order. Thus $\lambda_R = \lambda$, $0^R = 0$, and $001^R = 100$. Note that if $\omega \in \Sigma_2^\star$ then $\omega^R \in \Sigma_2^\star$ as well. Furthermore ω and ω^R always have the same length. They also contain the number of 0's and the same number of 1's.

Suppose that $L_2 = \{\omega \cdot \omega^R \mid \omega \in \Sigma_2^\star\}$. Then L_2 includes $\lambda, 00, 001100$ and lots of other strings, but not 10, 0010, or any string with odd length.

Second Example

Claim: L_2 is not a regular language.

Proof: The Pumping Lemma for Regular Languages will be used to prove this.

- With that noted, let p be any arbitrarily chosen integer such that $p \geq 1$.
- Consider the string $s = 0^p 1 1 0^p$.
  - $s \in L_2$, since $s = \omega \cdot \omega^R$ for the string $\omega = 0^p 1 \in \Sigma_2^*$.
  - $|s| = 2p + 2 \geq p$.

Second Example

- Let us ***assume*** that there exist strings $x, y, z \in \Sigma_2^\star$ such that $s = xyz$ and all three properties (included in the statement of the Pumping Lemma for Regular Languages) are satisfied for this choice of x, y and z.
- Since xy is a prefix of s with length at most p (by Property #3), and the first p symbols in s are all 0's, $xy = 0^k$ for some integer k such that $0 \leq k \leq p$. Since $s = 0^p 1 1 0^p = xyz$ it follows that $z = 0^{p-k} 1 1 0^p$.
- 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 $xy = 0^k$ it follows that $x = 0^h$ and $y = 0^\ell$.

Second Example

- Let $i = 2$. Then

$$xy^i z = xyyz = 0^h 0^\ell 0^\ell 0^{p-k} 110^p = 0^{p+\ell} 110^p$$

(since $h + \ell = k$), where $\ell = |y| \geq 1$.

Now if ℓ is odd then it is impossible for $xy^i z$ to be in L_2 because the length of $xy^i z$ is odd, and L_2 only includes strings with even length.

On the other hand, if ℓ is even then (since $\ell \geq 1$) $\ell \geq 2$. Furthermore, s has length $2(p + \ell/2 + 1) \leq 2(p + \ell)$.

Thus $s = 0^{p+\ell} 110^p$ does not have the form $\omega \cdot \omega^R$ for any string $\omega \in \Sigma^\star$ — because the number (0) of 1's in the first half of s is different from the number (2) of 1's in the second half.

Second Example

- Thus Property #1 is ***not*** satisfied (since $xy^i z \notin L_2$ when $i = 2$) and a ***contradiction*** has not been obtained.
- It follows that L_2 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. After mentioning that you are *going to* use the Pumping Lemma, you should write something like "Let p be an arbitrarily chosen integer such that $p \geq 1$."

   You ***cannot*** pick and use a particular value for p, because the corollary says that something must be true ***all*** such p.

Summary of Process

2. 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

3. In order to obtain a contradiction, ***assume*** that $s = xyz$ for strings $x, y, z \in \Sigma^\star$ such that properties 1, 2, and 3 (listed in the claim) 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

4. 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.