Chapter 10: Nonregular Languages *

Peter Cappello
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106
cappello@cs.ucsb.edu

- The corresponding textbook chapter should be read before attending this lecture.

- These notes are not intended to be complete. They are supplemented with figures, and other material that arises during the lecture period in response to questions.

*Based on Theory of Computing, 2nd Ed., D. Cohen, John Wiley & Sons, Inc.
THE PUMPING LEMMA

**Definition:** A language that cannot be defined by a regular expression is a **nonregular language** or an **irregular language**.
**Theorem:** For all regular languages, \( L \), with infinitely many words, there exists a constant \( n \) (which depends on \( L \)) such that for all strings \( w \in L \), where \(|w| \geq n\), there exists a factoring of \( w = xyz \), such that:

- \( y \neq \Lambda \).
- \( |xy| \leq n \).
- For all \( k \geq 0 \), \( xy^kz \in L \).

**Proof:**

1. Since \( L \) is regular, there is an FA \( A \) that accepts \( L \).
2. Let \( |Q_A| = n \).
3. Since \(|L| = \infty\), there exists a word \( w = a_0a_1\cdots a_m \in L \), for \( m \geq n \).
4. Let \( p_0, p_1, \ldots, p_m \) be the sequence of states visited by \( w \) as it is accepted by \( A \).

Since \( m \geq n \), at least 1 of these states appears previously in the sequence: There exists \( i < j \) such that \( p_i = p_j \).

Draw a picture of this situation.

5. Factor \( w \) into 3 strings as follows:
   - \( x = a_0a_1 \cdots a_i \).
   - \( y = a_{i+1}a_{i+2} \cdots a_j \).
   - \( z = a_{j+1}a_{j+2} \cdots a_m \).

6. Although either \( x \) or \( z \) may be \( \Lambda \), \( |y| \geq 1 \); the smallest loop in \( A \) is a self-loop, which consumes 1 symbol.

7. For any \( k \geq 0 \), \( xy^kz \in L \).
The Pumping Lemma as a 2-Person Game

1. You pick the language $L$ to be proved nonregular.

2. Your adversary picks $n$, but does not reveal to you what $n$ is. You must devise a move for all possible $n$’s.

3. You pick $w$, which may depend on $n$. $|w| \geq n$.

4. Your adversary picks a factoring of $w = xyz$. Your adversary does not reveal what the factors are, only that they satisfy the constraints of the theorem: $|y| > 0$ and $|xy| \leq n$.

5. You “win” by picking $k$, which may be a function of $n$, $x$, $y$, and $z$, such that $xy^kz \notin L$. 

\{a^n b^n \mid n = 0, 1, 2, \ldots\} \text{ is nonregular}

\textbf{Proof}

1. Assume that the adversary has chosen a particular \( n \).
2. Pick \( w = a^n b^n \).
3. Since \( |xy| \leq n \), \( y = a^i \), for some \( i > 0 \).
4. Then, \( xy^2 z \notin L \), since it has at least 1 more \( a \) than \( b \).
\{w \mid w \text{ has an equal number of } a’s \& b’s \} \text{ is Nonregular}

Proof

1. We refer to the language under consideration as \textit{EQUALS}.
   \[
   \{a^n b^n \mid n \geq 0\} = a^* b^* \cap \textit{EQUAL}.
   \]
2. If \textit{EQUALS} is regular, then \{a^n b^n \mid n \geq 0\} is regular.
3. \{a^n b^n \mid n \geq 0\} is nonregular.
4. \textit{EQUALS} is nonregular.

Study the applications of the pumping lemma given in the textbook.
The Myhill-Nerode Theorem

Given a language $L$, define a binary relation, $E$, on strings in $\Sigma^*$, where $xEy$ when for all $z \in \Sigma^*$, $xz \in L \iff yz \in L$.

1. $E$ is an equivalence relation.
2. If $L$ is regular, $E$ partitions $L$ into finitely many equivalence classes.
3. If $E$ partitions $L$ into finitely many equivalence classes, $L$ is regular.

Proof

1. For part 1:
   - $E$ is reflexive: $xEx$, for all $x \in \Sigma^*$.
   - $E$ is symmetric: If $xEy$ then $yEx$. 
• E is transitive: If $xEy$ and $yEz$ then $xEz$.
  
  (a) Let $xEy$ and $yEz$, and $w \in \Sigma^*$.
  (b) Since $xEy$, $xw \in L \iff yw \in L$.
  (c) Since $yEz$, $yw \in L \iff zw \in L$.
  (d) Therefore, $xw \in L \iff zw \in L$: $xEz$.

2. Since $L$ is regular, there is an FA $A$ that accepts it.

   Associate with each string, $w$, the state, $q$ of $A$ that $w$ ends in.

   If $x$ and $y$ are associated with the same state, they are in the same equivalence class.

   Since $A$ has a finite number of states, there is only a finite number of distinct equivalence classes.

   (It may be fewer than $|Q_A|$.)

3. Let $C_0, C_1, \ldots, C_n$ be the finite equivalence classes. Let $\Lambda \in C_0$.  

9
Claim: For all $C_i$, $C_i \subseteq L$ or $C_i \cap L = \emptyset$.

(a) Let $x, y \in C_i$ and $x \in L$.
(b) Then, $x\Lambda \in L \iff y\Lambda \in L$.
(c) Thus, $y \in L$.
(d) By analogous reasoning, if $x \notin L$, then $y \notin L$.

We build an FA $E$ that accepts $L$.

$Q_E$: The $C_i$ are $E$’s states.

$C_0$ is $E$’s start state.

If $C_i \subseteq L$, then $C_i \in F_E$.

For the $\delta$ function, consider the following.

(a) Let $a \in \Sigma$ and $z \in \Sigma^*$.

If $x, y \in C_i$, then $x(az) \in L \iff y(az) \in L$.
(b) Then, $(xa)z \in L \iff (ya)z \in L$. Thus, $xa, ya \in C_j$ for some $j$.

(c) Define $\delta(C_i, a) = C_j$.

4. Clearly, the language accepted by $E$ is $L$.

5. Therefore, $L$ is regular.
Applications of Myhill-Nerode

$a^n b^n$ is nonregular

Proof

Each $a^i$ is not equivalent to $a^j$, when $i \neq j$;
$a^i b^j \in L$ but $a^j b^i \notin L$.
There thus are infinitely many equivalence classes.

Please see other applications in the textbook.
Quotient Languages

Definition: \( \text{Pref}(Q \text{ in } R) = \{ p \mid \text{there exists } q \in Q \text{ such that } pq \in R \} \).

Example:
Let \( Q = \{aa, abaaabb, bbaaaaa, bbbbbbbbb\} \)
\( R = \{b, bbbb, bbbaaa, bbaaaaa\} \).
\( \text{Pref}(Q \text{ in } R) = \{b, bba, bbbaa\} \).

Theorem: If \( R \) is regular and \( L \) is a language, then \( \text{Pref}(L \text{ in } R) \) is regular.

Proof

Since \( R \) is regular, there is an FA that accepts it.
Let \( A \) be such an FA.

Construct an FA \( P \) that accepts \( \text{Pref}(L \text{ in } R) \) as follows:

1. \( Q_P = Q_A \).
2. The start state of \( P \) is \( q_0 \), the start state of \( A \).
3. \( q \in F_P \) if there exists a \( w \in L \) such that starting \( w \) in \( q \) leads to an accepting state in \( A \).
4. \( \delta_P = \delta_A \)

\( P \) accepts all words \( p \) such that \( pw \in R \) for some \( w \in L \).