CONTENTS

♦ Class Notes:
- Quiz #1 is taken place on Thur, March 6
- TA Review homework on Wed, March 5 from 2-3 P.M. Olsen 407 or Olsen 413

♦ Closure Properties Revisit

♦ Non-Regular Languages and Pumping Lemma

♦ Some Algorithms on Regular Languages
1. Let $L_1$ and $L_2$ be regular languages. We know that $L_1 \cup L_2$, $L_1 L_2$, and $L_1^*$ are regular.

We will show that $L_1 \cap L_2$, $L_1 - L_2$, and $L_1'$ are also regular. Let’s first look at $L_1'$, the complement of $L_1$. Since $L_1$ is regular, there exists a DFA $M_1$ such that $L_1 = L(M_1)$. We change the final states in $M$ to non-final, and the non-final states in $M_1$ to final, to obtain a new DFA $M'$. Then $x \notin L(M_1) \iff x \in L(M')$. Hence $L_1' = L(M')$; i.e., $L_1'$ is regular.
Now let’s look at $L_1 \cap L_2$.
We know that $L_1 \cap L_2 = (L_1' \cap L_2')'$ (by DeMorgan’s Law). Since $L_1'$ and $L_2'$ are regular, and regular languages are closed under union, we know that $L_1' \cup L_2'$ is regular. Hence, $(L_1' \cup L_2')'$ is also regular. Thus $L_1 \cap L_2$ is regular.

For $L_1 - L_2$; we know that $L_1 - L_2 = L_1 \cap L_2'$. Since regular languages are closed under complement and intersection, $L_2'$ is regular and so is $L_1 \cap L_2'$. Hence, $L_1 - L_2$ is regular.

**Non-regular Languages and Pumping Lemma**

Consider $L = \{a^n b^n | n \geq 0\}$. This language is not regular. Intuition: without extra memory, there is no way to memorize the exact number of a’s; namely, we cannot count.

But we need to provide convincing arguments why it is not regular. For doing so, let’s seek some properties that regular languages must possess and use such as property as a regularity test.

Let $L$ be regular. Then $\exists$ a DFA $M = (Q, \sum, \delta, S, F)$ such that $L = L(M)$. Assume $w \in L$, then $\exists$ states $q_1, q_2, \ldots, q_{|x|}$, where $q_0 = s, q_{|x|} \in F$, such that

$w = w_1 w_2 \ldots w_{|x|}, w_i \in \sum$

- If $|w| = |Q|$, then on this path there must be at least one state that occurs more than once.
- Assume that $q_i$ and $q_j$, $i \neq j$, are such states on the path.

We know that $w = xyz$ since $q_i = q_j$, that means we have

$|xy| \leq |w| = |Q|$

$xz \in L, xy^2z \in L, \ldots, xy^kz \in L$.

This property is possessed by any regular language, and we call it **pumping lemma**.
Pumping Lemma
Let \( L \) be a regular language, then there is a positive integer \( n \) such that \( \forall w \in L, \text{ if } |w| \geq n, \text{ then there are strings } x, y, z \text{ such that } w = xyz, y \neq e, |xy| \leq n, \text{ and } \forall i \geq 0 : xy^iz \in L. \)

If we can show that the language violates the pumping lemma, then it is not a regular language.

Example 1: \( L = \{a^n b^n \mid n \geq 0\} \) is not regular,
Proof. Assume that \( L \) is regular. Then by the pumping lemma, \( \exists \) a positive integer \( n \) such that \( \forall w \in L, \text{ if } |w| \geq n, \text{ then } \exists x, y, z \text{ such that } w = xyz, y \neq e, |xy| \leq n, \text{ and } \forall i \geq 0 : xy^iz \in L. \)

Select a particular string \( w_0 = a^n b^n \in L \) (we will show that this string violates the pumping lemma). Since \(|w_0| = 2n > n\), there are strings \( x, y, z \), such that \( w_0 = xyz, y \neq e, |xy| \leq n, \text{ and } \forall i \geq 0 : xy^iz \in L. \) Since \(|xy| \leq n\), \( xy \) must be a prefix of \( a^n \). So we have \( xy^0z = xz = a^{n-|y|} b^n \notin L \) because \( n - |y| < n \). A contradiction. Hence \( L \) is not regular.

Example 2: The palindrome language \( L = \{w \mid w = w^R \& w \in \{a, b\}^* \} \) is not regular.
Idea: Pick \( w_0 = a^n b a^n \in L \). Then \(|w_0| = 2n+1 > n\), but \( xy^0z = xz = a^{n-|y|} b a^n \notin L \).

Algorithm Finite Automata (p.47 in textbook)
Let \( L_1 \) and \( L_2 \) be regular, decide weather the following is true.
1. Membership Test: Given \( L_1 \) and \( x \), is \( x \in L_1 \) ?
2. Emptiness Test: Given \( L_1 \), is \( L_1 = \emptyset \) ?
3. Inclusion Test: Given \( L_1 \) and \( L_2 \), is \( L_1 \subseteq L_2 \) ?
   Note: \( L_1 \subseteq L_2 \) iff \( L_1 - L_2 = \emptyset \)
4. Equality Test: Given \( L_1 \) and \( L_2 \), is \( L_1 = L_2 \) ?
   Note: \( L_1 = L_2 \) iff \( L_1 \subseteq L_2 \& L_2 \subseteq L_1 \)