Recall from previous lectures that regular languages are languages that can be recognized by DFAs and regular expressions. Moreover, regular languages are closed under operations including complement, union, intersection, and Kleene star.

Topics Covered

1. Nonregular Languages
2. The Pumping Lemma
3. Applications of the Pumping Lemma

1 Nonregular Languages

Consider the language $A = \{0^k1^k \mid k \geq 0\}$. That is, $A = \{\epsilon, 01, 0011, 000111, \ldots\}$. We claim that $A$ is not regular.

Proof that $A = \{0^k1^k \mid k \geq 0\}$ is Not Regular

Suppose that $A$ were regular. Let $M$ be a DFA recognizing $A$, and let $n$ be the number of states in $M$. Consider an input $w = 0^n1^n$, so $w \in A$. Let $r_0, r_1, \ldots, r_{2n}$ be $M$’s computation history on input $w$. Consider the first $n+1$ states $r_0, r_1, \ldots, r_n$. There are $n+1$ visits to states but only a total of $n$ states. By the pigeonhole principle, some state must have been visited at least twice. We call this state $q = r_i = r_j$.

We know that the first $0^i$ zeroes in $w$ took us from the start state to $q$, and the next $0^{j-i}$ zeroes took us from $q$ back to $q$ again. The remainder of the input, $0^{n-j}1^n$, takes us from $q$ to an accept state. We can thus write $w$ as $w = 0^i0^{j-1}0^{n-j}1^n = xyz$ with $x = 0^i$, $y = 0^{j-i}$, and $z = 0^{n-j}1^n$. But then the string $w' = xyyz$ is also accepted by $M$. $M$’s computation history on $w'$ is $r_0, \ldots, r_i = q, r_j = q, \ldots, r_j = q, r_{j-1}, \ldots, r_{2n}$. Just as it was in the CH of $w$, $r_{2n}$ is an accepting state, so there is an accepting CH for $w'$. However, $w'$ is not in $A$ as it contains more zeroes than ones. Thus $M$ cannot be a DFA for $A$, a contradiction. It follows that $A$ is not regular. ■
Regular Operations and Nonregular Languages  What happens when we perform regular operations on nonregular languages? It depends on the languages. For example, if we take the union of \( \{0^k1^k\} \) (nonregular) and \( \Sigma^* \) (regular), the result is \( \Sigma^* \) (regular). If we instead take the union of \( \{0^k1^k\} \) (nonregular) and \( \emptyset \) (regular), the result is \( \{0^k1^k\} \) (nonregular). The union of \( \{0^k1^k\} \) (nonregular) and its complement (nonregular) is \( \Sigma^* \) (regular). Finally, the union of \( \{0^k1^k\} \) (nonregular) and \( \{1^k0^k\} \) (nonregular) is also nonregular. In other words, we cannot make guarantees about what happens when we apply regular operations to nonregular languages.

For example, let \( B = \{w \mid w \text{ is a binary string that has as many 0s as 1s}\} \). To prove that \( B \) is not regular, we first suppose that it were. Now consider the language of the regular expression \( 0^*1^* \). The intersection of \( B \) and \( L(0^*1^*) \) is equal to \( A \), which we know is not regular. The closure of regular languages under intersection would imply otherwise, so we reach a contradiction. Thus, \( B \) is not regular.

Moreover, the complement of a nonregular language \( A \) is not regular. If \( A^C \) were regular, then \( (A^C)^C = A \) would be regular, a contradiction.

2 The Pumping Lemma

The Pumping Lemma  Let \( L \) be a regular language. Then there exists an integer \( p > 0 \), called \( L \)'s pumping length, such that for any string \( s \in L \) such that \( |s| \geq p \), then we can write \( s = xyz \) such that:

1. \( xy^iz \in L \) for all \( i \geq 0 \)
2. \( |y| \geq 1 \)
3. \( |xy| \leq p \)

The pumping lemma essentially says that if a language is regular, a string that is long enough can be pumped and still be in the language. If all strings \( s \) in the language have length less than \( p \), then the lemma is vacuously satisfied. In particular, any finite language is regular.

Proof of Pumping Lemma  Let \( M = (Q, \Sigma, \delta, q_0, F) \) be a DFA that recognizes the language \( L \). Let \( p = |Q| \), the number of states of \( M \). Let \( s \) be in \( L \) such that \( n = |s| \geq p \). The CH of \( M \) on input \( s \) is \( r_0, r_1, \ldots, r_n \) such that \( r_0 = q_0 \) and \( r_n \in F \). By the pigeonhole principle, some state \( q \) appears at least twice in the first \( p + 1 \) visits in the CH of \( M \) on \( s \).
That is, there is a repeated state $q$ among $r_0, r_1, \ldots, r_p$. Let $i$ and $j$ be the positions in this sequence that correspond to revisiting the same state $q$. We know that $i \neq j$ and without loss of generality, we let $0 \leq i < j \leq p$. Let the string $s = s_1s_2 \ldots s_n$. If $i = 0$, let $x = \epsilon$; otherwise, let $x = s_1 \ldots s_i$. Then let $y = s_{i+1} \ldots s_j$ and $z = s_{j+1} \ldots s_n$. Now we need to show that all three components of the pumping lemma hold.

1. For all $i$, $s' = xy^i z \in L$ because the CH of $M$ on input $s'$ is $r_0, \ldots, r_i = q$ (from $x$), $r_{i+1}, \ldots, r_j = q$ (from $y$), $r_{i+1}, \ldots, r_j = q$ (from $y$), $r_{i+1}, \ldots, r_j = q$ (from $z$). The intermediate sequence of states $r_{i+1}, \ldots, r_j = q$ is repeated for each of the $i$ instances of $y$. For all such computation histories, $r_n \in F$, so $s' \in L$.

2. The condition $|y| \geq 1$ is satisfied based on our construction of $y$.

3. Similarly, the condition $|xy| \leq p$ is satisfied by our construction.

As all three conditions hold, we’ve shown that $p = |Q|$ is a pumping length for a regular language $L$. Hence, all regular languages have a pumping length satisfying the components of the pumping lemma. ■

3 Applications of the Pumping Lemma

The main application of the pumping lemma is to show that languages are not regular.

Proof that $E = \{1^q \mid q \text{ is prime}\}$ is Not Regular The language $E = \{11, 111, 11111, 1111111, \ldots\}$. We first suppose that $E$ were regular. Then let $p$ be its pumping length. Let $q \geq p$ be a prime number. Let the string $s = 1^q$. By the pumping lemma, we can write $s = xyz$ such that the three conditions of the lemma hold. Let $|y| = m$.

Consider the pumped string $s' = xy^{q+1}z$. This is $xyqz$ and consists only of 1s, so it can be expressed as $xyz \circ y^q = 1^{q+qm}$. By the first part of the pumping lemma, we conclude that $s' \in E$. However, $q +qm$ is divisible by $q$ and is thus not prime, a contradiction. This implies that $E$ is not regular. ■

Proof that $F = \{0^a1^b \mid a \geq b\}$ is Not Regular Suppose that $F$ were regular. Let $p$ be its pumping length. Let $s = 0^p1^p$. By the pumping lemma, we can divide $s = xyz$ to satisfy the three conditions. Note that $xy$ consists only of 0s because we must have $|xy| \leq p$. Then $y$ consists of a nonzero number of only 0s. Let $y = 0^k$ for some $1 \leq k \leq p$.

Then $s' = xy^0z = xz = 0^{p-k}1p$. The string $s'$ has more 1s than 0s, so $s' \notin F$. This contradiction implies that $F$ is not regular. (Note that we could also choose a string $s = 0^{p/2}1^{p/2}$, but this would not guarantee a string $y$ consisting only of 0s. Removing $y$
would not tell us anything about the number of 0s an 1s in a string \( s' \), but pumping \( y \) would change the order of 1s and 0s. This approach works and requires case analysis.)

**Proof that** \( G = \{ w \mid w \text{ is a palindrome} \} \) **is Not Regular**

Let the alphabet \( \Sigma = \{0, 1\} \). Suppose that \( G \) were regular. Let \( p \) be its pumping length. Let \( s = 0^p1^p \), a palindrome. By the pumping lemma, we can write \( s = xyz \) where \( x \) and \( y \) must consist entirely of 0s. Let \( y = 0^m \). Then \( s' = xyyz = 0^{p+m}1^p \), which is not a palindrome. As a result, \( G \) is not regular. ■