Properties of Regular Languages

Reading: Chapter 4
Pumping Lemma for Regular Languages

**Lemma:** (the pumping lemma)
Let M be a DFA with |Q| = n states, and let x be a string in L(M), where |x| ≥ n. Then x = uvw, where u,v, and w are all in Σ* and:

- 1 ≤ |uv| ≤ n
- |v| ≥ 1
- uv^i w is in L(M), for all i ≥ 0

**Note** this statement of the pumping lemma is slightly different than the books’.
Pumping Lemma for Regular Languages

• Proof:
Let M be a DFA where |Q| = n, and let x be a string in L(M) where |x|>=n. Furthermore, let x = a_1a_2 ... a_m where δ(q_0, a_1a_2 ... a_p) = q_{jp}.

\[
\begin{array}{c}
a_1 \\
q_{j_0}
\end{array}
\begin{array}{c}
a_2 \\
q_{j_1}
\end{array}
\begin{array}{c}
a_3 \\
q_{j_2}
\end{array}
\ldots
\begin{array}{c}
a_m \\
q_{j_m}
\end{array}
\begin{array}{c}
m>=n
\end{array}
\begin{array}{c}
\text{and q}_{j_0}\text{ is } q_0
\end{array}
\]

Consider the first n symbols, and first n+1 states on the above path:

\[
\begin{array}{c}
a_1 \\
q_{j_0}
\end{array}
\begin{array}{c}
a_2 \\
q_{j_1}
\end{array}
\begin{array}{c}
a_3 \\
q_{j_2}
\end{array}
\ldots
\begin{array}{c}
a_n \\
q_{j_n}
\end{array}
\]

Since |Q| = n, it follows from the pigeon-hole principle that j_s = j_t for some 0<=s<=t<=n, i.e., some state appears on this path twice (perhaps many states appear more than once, but at least one does).
• Let:
  
  - $u = a_1 \ldots a_s$
  - $v = a_{s+1} \ldots a_t$

• Since $0 \leq s < t \leq n$ and $uv = a_1 \ldots a_t$ it follows that:
  
  - $1 \leq |v|$ and therefore $1 \leq |uv|$ 
  - $|uv| \leq n$ and therefore $1 \leq |uv| \leq n$

• In addition, let:
  
  - $w = a_{t+1} \ldots a_m$

• It follows that $uv^i w = a_1 \ldots a_s (a_{s+1} \ldots a_t)^i a_{t+1} \ldots a_m$ is in $L(M)$, for all $i \geq 0$.

In other words, when processing the accepted string $x$, the cycle was traversed once, but could have been traversed as many times as desired, and the resulting string would still be accepted.
• Example:

\[ n = 5 \]

\[ x = 0001000 \text{ is in } L(M) \]

\[
\begin{array}{ccccccc}
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
q_0 & q_1 & q_2 & q_3 & q_1 & q_2 & q_3 & q_4 \\
\end{array}
\]

\[ \text{first } n \text{ symbols} \]

\[ u = 0 \]
\[ v = 001 \]
\[ w = 000 \quad uv^iw \text{ is in } L(M), \text{ i.e., } 0(001)^i000 \text{ is in } L(M), \text{ for all } i \geq 0 \]
• Note this does not mean that every string accepted by the DFA has this form:

  – 001 is in L(M) but is not of the form 0(001)^i000

• Similarly, this doesn’t even mean that every long string accepted by the DFA has this form:

  – 0011111 is in L(M), is very long, but is not of the form 0(001)^i000

• Note, however, in this latter case 0011111 could be similarly decomposed.
• **Note:** It may be the case that no $x$ in $L(M)$ has $|x| \geq n$. 

```plaintext
\[ q_0 \xrightarrow{0,1} q_1 \xrightarrow{0,1} q_2 \]
```
• Example:

![Diagram of a nondeterministic finite automaton](image)

\[ bbbab \text{ is in } L(M) \]

\[ |x| = 5 \]

\[ u = \varepsilon \]
\[ v = b \]
\[ w = bbab \]

\((b)^i bbab \text{ is in } L(M), \text{ for all } i \geq 0\)
- Example:

\[
\begin{array}{c}
q_0 \xrightarrow{a} q_1 \\
q_1 \xrightarrow{b} q_2 \xleftarrow{a,b} q_3 \\
q_2 \xleftarrow{a} q_0
\end{array}
\]

\[
x = \text{bbbab is in } L(M) \\
|x| = 5
\]

\[
u = b \\
v = b \\
w = \text{bab}
\]

\[
b(b)^i\text{bab is in } L(M), \text{ for all } i \geq 0
\]

\[
n = 4
\]

\[
\begin{array}{cccccc}
q_0 & q_0 & q_0 & q_0 & q_1 & q_3 \\
b & b & b & a & b
\end{array}
\]
• Example:

\[ n = 4 \]

\[ x = bbbab \text{ is in } L(M) \]
\[ |x| = 5 \]

\[ u = bb \]
\[ v = b \]
\[ w = ab \]

\[ bb(b)^iab \text{ is in } L(M), \text{ for all } i \geq 0 \]
• Example:

\[ x = bbbab \text{ is in } L(M) \]
\[ |x| = 5 \]

\[ u = b \]
\[ v = bb \]
\[ w = ab \]

\[ b(bb)^iab \text{ is in } L(M), \text{ for all } i \geq 0 \]
NonRegularity Example

• **Theorem:** The language:

\[ L = \{0^k1^k \mid k \geq 0\} \]  \hspace{1cm} (1)

is not regular.

• **Proof:** (by contradiction) Suppose that \( L \) is regular. Then there exists a DFA \( M \) such that:

\[ L = L(M) \]  \hspace{1cm} (2)

We will show that \( M \) accepts some strings not in \( L \), contradicting (2).

Suppose that \( M \) has \( n \) states, and consider a string \( x = 0^m1^m \), where \( m \gg n \).

By (1), \( x \) is in \( L \).

By (2), \( x \) is also in \( L(M) \).
Since $x$ is very long, i.e., $|x| = 2^m \gg n$, it follows from the pumping lemma that:

- $x = uvw$
- $1 \leq |uv| \leq n$
- $1 \leq |v|$, and
- $uv^iw$ is in $L(M)$, for all $i \geq 0$

Since $1 \leq |uv| \leq n$ and $n \ll m$, it follows that $1 \leq |uv| \leq m$.

Also, since $x = 0^m1^m$ it follows that $uv$ is a substring of $0^m$.

In other words $v = 0^j$, for some $j \geq 1$.

Since $uv^iw$ is in $L(M)$, for all $i \geq 0$, it follows that $0^{m+cj}1^m$ is in $L(M)$, for all $c \geq 1$.

But by (2), $0^{m+cj}1^m$ is in $L$, for any $c \geq 1$, a contradiction. •
NonRegularity Example

• **Theorem:** The language:
  \[ L = \{0^k1^k2^k \mid k \geq 0\} \]  
  (1)
  is not regular.

• **Proof:** (by contradiction) Suppose that \( L \) is regular. Then there exists a DFA \( M \) such that:
  \[ L = L(M) \]  
  (2)
  We will show that \( M \) accepts some strings not in \( L \), contradicting (2).

  Suppose that \( M \) has \( n \) states, and consider a string \( x=0^m1^m2^m \), where \( m \gg n \).

  By (1), \( x \) is in \( L \).

  By (2), \( x \) is also in \( L(M) \).
Since is very long, i.e., $|x| = 3^m n$, it follows from the pumping lemma that:

- $x = uvw$
- $1 \leq |uv| \leq n$
- $1 \leq |v|$, and
- $uv^i w$ is in $L(M)$, for all $i \geq 0$

Since $1 \leq |uv| \leq n$ and $n << m$, it follows that $1 \leq |uv| \leq m$.

Also, since $x = 0^m 1^m 2^m$ it follows that $uv$ is a substring of $0^m$.

In other words $v = 0^j$, for some $j \geq 1$.

Since $uv^i w$ is in $L(M)$, for all $i \geq 0$, it follows that $0^{m+cj} 1^m 2^m$ is in $L(M)$, for all $c \geq 1$.

But by (2), $0^{m+cj} 1^m 2^m$ is in $L$, for any $c \geq 1$, a contradiction. •
NonRegularity Example

- **Theorem:** The language:
  \[ L = \{ 0^m 1^n 2^{m+n} \mid m, n \geq 0 \} \quad (1) \]
  is not regular.

- **Proof:** (by contradiction) Suppose that \( L \) is regular. Then there exists a DFA \( M \) such that:
  \[ L = L(M) \quad (2) \]

  We will show that \( M \) accepts some strings not in \( L \), contradicting (2).

  Suppose that \( M \) has \( n \) states, and consider a string \( x = 0^m 1^n 2^{m+n} \), where \( m \gg n \).

  By (1), \( x \) is in \( L \).

  By (2), \( x \) is also in \( L(M) \).
Since \(|x| = 2(m+n) >> n\), it follows from the pumping lemma that:

- \(x = uvw\)
- \(1 <= |uv| <= n\)
- \(1 <= |v|, \text{ and}\)
- \(uv^i w\) is in \(L(M)\), for all \(i >= 0\)

Since \(1 <= |uv| <= n\) and \(n << m\), it follows that \(1 <= |uv| <= m\).

Also, since \(x = 0^m 1^n 2^{m+n}\) it follows that \(uv\) is a substring of \(0^m\).

In other words \(v = 0^j\), for some \(j >= 1\).

Since \(uv^i w\) is in \(L(M)\), for all \(i >= 0\), it follows that \(0^{m+cj} 1^m 2^{m+n}\) is in \(L(M)\), for all \(c >= 1\). In other words \(v\) can be “pumped” as many times as we like, and we still get a string in \(L(M)\).

But by (2), \(0^{m+cj} 1^n 2^{m+n}\) is in \(L\), for any \(c >= 1\), a contradiction.

- Note that the above proof is almost identical to the previous proof.