Myhill-Nerode theorem proves another property of all regular languages. Analogously to the pumping lemma, we can use this property to prove that a language is not regular. But first we need to give some definitions.

**Definition.** Let $x$ and $y$ be two strings and $L$ be a language (not necessarily regular). We say that $x$ and $y$ are distinguishable by $L$ if there exists such a string $z$ that exactly one string of $xz$ and $yz$ belongs to $L$. Otherwise we call $x$ and $y$ indistinguishable by $L$.

**Definition.** Let $X$ be a set of strings and $L$ be a language (not necessarily regular). We say that the set $X$ is pairwise distinguishable by $L$ if every two distinct strings in $X$ are distinguishable by $L$.

**Definition.** Index of a language $L$ is the maximum number of elements in any set that is pairwise distinguishable by $L$.

To prove the Myhill-Nerode theorem, we will need two lemmas.

**Lemma A.** If $L$ is recognised by a DFA with $k$ states, then $L$ has index at most $k$.

**Proof.** We will prove by contradiction.

Let $M$ be a DFA with $k$ states that recognise $L$. Suppose that index of $L$ is greater than $k$. Then there is a set $X$ with more than $k$ elements such that $X$ is pairwise distinguishable by $L$.

Since $M$ has only $k$ states, there exist two distinct strings $x_1, x_2 \in X$ such that $\delta(q_0, x_1) = \delta(q_0, x_2)$. That is, after reading $x_1$ or $x_2$, $M$ is in the
same state. Then $\delta(q_0, x_1z) = \delta(q_0, x_2z)$ for any string $z$, i.e. $M$ is in the same state after reading $x_1z$ or $x_2z$. Therefore $x_1z$ and $x_2z$ are either both accepted or both rejected by $M$ for any string $z$. This means $x_1$ and $x_2$ are indistinguishable by $L$. Contradiction!

Therefore the assumption was wrong and index of $L$ is not more than $k$.

Lemma B. If index of a language $L$ is a finite number $k$ then $L$ is recognised by a DFA with $k$ states.

Proof. Let $X = \{x_1, x_2, \ldots, x_k\}$ be pairwise distinguishable by $L$. We construct a DFA $M = (Q, \Sigma, \delta, q_0, F)$ that recognises $L$.

Let $Q = \{q_1, q_2, \ldots, q_k\}$. For each $q_i \in Q$ and $a \in \Sigma$ we define: $\delta(q_i, a) = q_j$ where $j$ is such that $x_i a$ is indistinguishable from $x_j$. Such $x_j$ exists and is unique.

Indeed, $x_i a$ should be indistinguishable from some $x_j$ because otherwise we could increase the set $X$ by adding $x_i a$ and that would contradict the fact that the index of $L$ is $k$. Such $x_j$ is also unique because otherwise there would be indistinguishable strings in $X$.

Let $F = \{q_i \mid x_i \in L\}$ and $q_0 = q_j$ such that $\epsilon$ is indistinguishable from $x_j$.

Automaton $M$ is constructed in the way that for any $q_i$:

$$\{x \mid \delta(q_0, x) = q_i\} = \{\text{strings indistinguishable from } x_i\}.$$ 

Every string $y$ is indistinguishable from some $x_i \in X$ (otherwise we could include this $y$ in $X$ which contradicts that index of $X$ is $k$). Having fixed $y$ and $x_i$, consider all strings $z$: for any $z$ two strings $yz$ and $x_iz$ are either both belong to $L$ or none of them belong to $L$ (because $y$ and $x_i$ are indistinguishable by $L$).

It is also true for any particular $z$, for example $z = \epsilon$. It means that if $y \in L$ than $x_i \epsilon = x_i \in L$ and the automaton $M$ finishes in an accept state. But if $y \notin L$ then $x_i \notin L$ and the automaton $M$ finishes in non-accept state. Therefore $M$ accepts exactly strings from $L$. 

Myhill-Nerode theorem. Language $L$ is regular if and only if it has a finite index. Moreover, its index is the size of the smallest DFA that recognises $L$.

Proof. Suppose that $L$ is regular. Let $k$ be the number of states in DFA that recognises $L$. Then, from lemma A, $L$ has index at most $k$.

Conversely, if $L$ has index $k$, from lemma B there exists DFA that recognises it; and this DFA has $k$ states, and thus $L$ is regular.
Next, we show that the index of $L$ the size of the smallest DFA accepting it. Suppose that the index of $L$ is exactly $k$. Then, by lemma B, there is a $k$-state DFA accepting $L$. If there were a smaller DFA accepting $L$, we could show by lemma A that the index of $L$ is smaller than $k$.

**Practise session**

Since it is the last week of part 2, we solve here some problems on different topics.

1. Show that $L = \{1^{2^n} \mid n \geq 0 \}$ is not regular. (It a set of all strings of ones of length $2^n$ for $n \geq 0$.)

2. Let $L$ be the language of all strings consisting of some positive number of zeros, followed by some string twice, followed by some positive number of zeros:
$$L = \{0^k w 0^m \mid k, m \geq 1, w \in \{0,1\}^*\}.$$

For example,
$$0000 10101 \underbrace{10101}_{w} \underbrace{00}_{w} \in L$$

Show that $L$ is not regular.

3. Let $C_5 = \{x \mid x$ is a binary number that is multiple of 5 $\}$. Show that $C_5$ is regular.

4. Are the following statements true or false?

   (a) If $L_1 \cup L_2$ is regular and $L_1$ is finite, then $L_2$ is regular.

   (b) If $L_1 \cup L_2$ is regular and $L_1$ is regular, then $L_2$ is regular.

   (c) If $L_1 L_2$ is regular and $L_1$ is regular, then $L_2$ is regular.

   (d) If $L^*$ is regular then $L$ is regular.
Solutions to the practice session questions

Since it is the last week of part 2, we solve here some problems on different topics.

1. Show that \( L = \{1^{2^n} \mid n \geq 0\} \) is not regular. (It a set of all strings of ones of length \( 2^n \) for \( n \geq 0 \).)

Solution. We will use a pumping lemma (see previous lecture for details).

Assume that \( L \) is regular and \( p \) is its pumping length give by the pumping lemma. Choose \( w = 1^{2p} \). Clearly, \( w \in L \) and \( |w| \geq p \). Therefore \( w = xyz \), where \( |xy| \leq p \), \( |y| > 0 \) and for all \( i \geq 0 \) we have \( xy^i z \in L \).

Since \( p < 2^p \) for any \( p \geq 0 \), so \( |y| < 2^p \). Thus \( |xyyz| = |xyz| + |y| < 2^p + 2^p = 2^{p+1} \). That is why

\[
2^p < |xyyz| < 2^{p+1}
\]

and the length of the word \( xyyz \) is not a power of 2. It means that \( xyyz \not\in L \). Contradiction. Therefore \( L \) is not regular.

2. Let \( L \) be the language of all strings consisting of some positive number of zeros, followed by some string twice, followed by some positive number of zeros:

\[
L = \{0^k w0^m \mid k, m \geq 1, \ w \in \{0, 1\}^*\}.
\]

For example,

\[
\underline{0000} \underline{10101} \underline{10101} \underline{00} \in L
\]

Show that \( L \) is not regular.

Solution. We will use Myhill-Nerode’s theorem. More precisely, we show that there is an infinite set of strings, such that any two of them are distinguishable with respect to \( L \). This means that index of \( L \) is infinite and \( L \) is not regular.

Consider the set \( \{01^k 0 \mid k \geq 1\} \). Choose two arbitrary words from this set, \( 01^{k_1} 0 \) and \( 01^{k_2} 0 \) where \( k_1 \neq k_2 \). Let \( z = 1^{k_1} 00 \). On the one hand, \( 01^{k_1} 0z = 01^{k_1} 01^{k_1} 00 \) obviously belongs to \( L \).

On the other hand, \( 01^{k_2} 0z = 01^{k_2} 01^{k_1} 00 \). If it is in \( L \), then it is of the form \( 0^k w0^m \) for some \( w \). Then, \( w \) should contain at least one zero. Then, \( w \) should end with zero, and so it is \( 0ww0 \). Then, \( w \) should be \( 1^{k_1} \) and \( 1^{k_2} \) at the same time. It is impossible. So all strings from the set \( \{01^k 0 \mid k \geq 1\} \) are distinguishable, and the index of \( L \) is infinity, i.e. it is not regular.

3. Let \( C_5 = \{x \mid x \text{ is a binary number that is multiple of } 5 \} \). Show that \( C_5 \) is regular.
Solution. We construct DFA $M = (Q, \Sigma, \delta, q_0, F)$ that recognises $C_5$. Let $Q = \{q_0, q_1, q_2, q_3, q_4\}$, $\Sigma = \{0, 1\}$, $F = \{q_0\}$, start state be $q_0$ and transition function be

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$q_0$</td>
<td>$q_1$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$q_2$</td>
<td>$q_3$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$q_4$</td>
<td>$q_0$</td>
</tr>
<tr>
<td>$q_3$</td>
<td>$q_1$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_4$</td>
<td>$q_3$</td>
<td>$q_4$</td>
</tr>
</tbody>
</table>

State diagram of this automaton is as follows

![State diagram of DFA](image)

The state of the automaton stores the reminder of currently read input divided by 5: states $q_0, \ldots, q_4$ correspond to reminders 0, \ldots, 4, respectively (so $q_0$, i.e. reminder 0, is accept state).

If the number that we read so far is $x$ with remainder $x \mod 5 = r$ and we read one more digit:

<table>
<thead>
<tr>
<th>read number</th>
<th>remainder</th>
<th>remainders (respectively)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$x \mapsto 2x$</td>
<td>$r \mapsto 2r \mod 5$ $(0, 1, 2, 3, 4) \mapsto (0, 2, 4, 1, 3)$</td>
</tr>
<tr>
<td>1</td>
<td>$x \mapsto 2x + 1$</td>
<td>$r \mapsto (2r + 1) \mod 5$ $(0, 1, 2, 3, 4) \mapsto (1, 3, 0, 2, 4)$</td>
</tr>
</tbody>
</table>

According to respective change of remainders we build the transition function. For example, if we have a number with binary representation $x$ with remainder\(^1\) 3 and we read 1, then the new number will have binary representation $x1$ and remainder $(3 \cdot 2 + 1) \mod 5 = 7 \mod 5 = 2$; hence we put arrow $q_3 \to q_2$ with label “1”.

Let us see for instance how the automaton works on input 11110 (i.e. binary representation of 30):

<table>
<thead>
<tr>
<th>input value</th>
<th>1</th>
<th>11</th>
<th>111</th>
<th>1111</th>
<th>11110</th>
</tr>
</thead>
<tbody>
<tr>
<td>input value</td>
<td>= 3</td>
<td>= 7</td>
<td>= 15</td>
<td>= 30</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\)Note that exact $x$ is not important; it is only a remainder that matters.
4. Are the following statements true or false?

(a) If \( L_1 \cup L_2 \) is regular and \( L_1 \) is finite, then \( L_2 \) is regular.

(b) If \( L_1 \cup L_2 \) is regular and \( L_1 \) is regular, then \( L_2 \) is regular.

(c) If \( L_1L_2 \) is regular and \( L_1 \) is regular, then \( L_2 \) is regular.

(d) If \( L^* \) is regular then \( L \) is regular.

Solution.

(a) True. Note that 
\[ L_2 = (L_1 \cup L_2) \cap (L_1 \setminus L_2)^c, \]
where \(^c\) stand for complementary language. \( L_1 \cup L_2 \) is regular (given), \( L_1 \setminus L_2 \) is regular (every finite language is regular) and \((L_1 \setminus L_2)^c\) is regular as complementary to regular \(^2\). Therefore \( L_2 \) is an intersection of two regular languages and thus regular itself.

(b) False. Consider \( L_1 = \Sigma^* \) and \( L_2 \) being any nonregular language. Then \( L_1 \cup L_2 = \Sigma^* \) is regular but \( L_2 \) is not.

(c) False. Let \( L_1 = \{ \varepsilon, 0 \} \). This is a finite language and thus regular. Let 
\[ L_2 = (00)^* \cup \{ 0^n^2 \mid n \geq 0, n \text{ is prime} \}. \]
It could be shown this language is not regular. However \( L_1L_2 = 0^* \) is regular.

(d) False. Let \( L = \{ 0^n1^n \mid n \geq 0 \} \cup \{ 0, 1 \} \). This language is not regular (could be proven analogously to example 1 in lecture 8). But \( L^* = \Sigma^* \) is regular.

---

\(^2\)To see that complementary to a regular language \( L \) is also regular, we note that from DFA for \( L \) we build DFA for \( L^c \) by just making all accept states be non-accept, and vice versa.