Not All Languages Are Regular

**What we will do.** We will use Pumping Lemma to prove that some languages are not regular.

Let us recall what is a regular language and what is the Pumping Lemma.

**What is a regular language: reminder.** A language is regular if there exists a finite automaton that:

- accepts all the words from this language and
- rejects all the words which are not in this language.

**Note.** We have shown, in the previous lectures, that:

- for each regular expression, there exists a finite automaton that accepts exactly the word from the language described by this regular expression, and
- for every finite automaton, there exists a regular expression for which the corresponding language contains all the words accepted by this automaton and none of the words rejected by this automaton.

Thus, we can alternatively define a regular language as a language described by a regular expression.

**Pumping Lemma: reminder.** For every regular language $L$, there exists a natural number $p$ such that every word from $L$ whose length $\text{len}(w)$ is at least $p$ can be represented as a concatenation $w = xyz$, where:

- $y$ is non-empty;
- the length $\text{len}(xy)$ does not exceed $p$, and
- for every natural number $i$, the word $xy^i z \overset{\text{def}}{=} xy \ldots yz$, in which $y$ is repeated $i$ times, also belongs to the language $L$.

**Theorem.** The following language is not regular:

$$L = \{a^n b^n : n = 0, 1, 2, \ldots\} = \{\Lambda, ab, aabb, aaabbb, \ldots\}.$$
**Why this language is important.** The same result can be proven for languages
\[ \{A, \{\}, \{\{\}, \{\{\}}\}, \ldots \} \]
corresponding to the natural requirement that we must exactly as many closing curly brackets as opening ones. This result means that we cannot test this requirement by using finite automata.

**Proof:** by contradiction. Let us assume that the language \( L \) is regular, and let us show that this assumption leads to a contradiction.

Since this language is regular, according to the Pumping Lemma, there exists an integer \( p \) such that every word from \( L \) whose length \( \text{len}(w) \) is at least \( p \) can be represented as a concatenation \( w = xyz \), where:

- \( y \) is non-empty;
- the length \( \text{len}(xy) \) does not exceed \( p \), and
- for every natural number \( i \), the word \( xy^iz \) \( \overset{\text{def}}{=} xy \ldots yz \), in which \( y \) is repeated \( i \) times, also belongs to the language \( L \).

Let us take the word
\[ w = a^pb^p = a \ldots ab \ldots b, \]
in which first the letter \( a \) is repeated \( p \) times and then the letter \( b \) is repeated \( p \) times. The length of this word is \( p + p = 2p > p \). So, by pumping lemma, this word can be represented as \( w = xyz \) with \( \text{len}(xy) \leq p \). The word \( w \) starts with \( xy \), and the length of \( xy \) is smaller than or equal to \( p \). Thus, \( xy \) is among the first \( p \) symbols of the word \( w \) – and these symbols are all \( a \)'s. So, the word \( y \) only has \( a \)'s.

Thus, when we go from the word \( w = xyz \) to the word \( xyyz \), we add \( a \)'s, and we do not add any \( b \)'s. So, in the word \( xyyz \), there are more \( a \)'s than \( b \)'s. Thus, the word \( xyyz \) cannot be in the language \( L \), since by definition \( L \) only contains words which have equal number of \( a \)'s and \( b \)'s.

On the other hand, by Pumping Lemma, the word \( xyyz \) must be in the language \( L \). So, we proved two opposite statements:

- that this word is not in \( L \) and
- that this word is in \( L \).

This is a contradiction.

The only assumption that led to this contradiction is that \( L \) is a regular language. Thus, this assumption is false, so \( L \) is not regular.

**Important.** It is important to understand every step, and to be able to reproduce this proof verbatim – without skipping steps. Do not try to reformulate it in your own words, do not try to ask “Do you mean that ...” – asking to
compress this proof into a shorter text. It cannot be compressed, all parts are important.

This proof is a template for proving that other languages are not regular. To prove these other results, we need to modify this template.

**First example.** How can we prove that the language 

\[ L = \{ a^{n+1}b^n : n = 0, 1, 2, \ldots \} = \{ a, aab, aaabb, \ldots \} \]

is not regular? Here, as an example, we can take \( w = a^{p+1}b^p \), and the contradiction will be that in all words form \( L \) there is a balance: exactly one more \( a \) than \( b \). So, when we add \( y \) and thus, add \( a \)'s without adding \( b \)'s, we disrupt this balance, so \( xyyz \not\in L \) and we get a contradiction.

**Second example.** Similarly, all words from the language

\[ L = \{ a^n b^{2n} : n = 0, 1, 2, \ldots \} = \{ \Lambda, abb, aabbb, \ldots \} \]

has exactly twice many \( b \)'s than \( a \)'s. If we take \( w = a^pb^2p \) and use the Pumping Lemma to add \( a \)'s, we disrupt this balance, so we still get a contradiction.

**Third example.** What if we have a language consisting of all possible words repeated twice

\[ L = \{ ww \} = \{ \text{catcat, dogdog, bagbag, \ldots} \} \]

In this case, the word \( a^p b^p \) used in the original proof is not a good example, since it does not belong to this language, but we can repeat this word twice and get the word \( w = a^pb^pa^pb^p \) which is in this example’s language \( L \). In this case, when we go from \( w = xyz \) to \( xyyz \), we add \( a \)'s to the first part of the word, and get \( a^{p+q}b^p a^pb^p \) for some \( q > 0 \) – so the word \( xyyz \) is not a repetition of the same word and thus, it is not in \( L \).

**Practice.** Try to convert these hints into a detailed proofs – using the above proof as a template.