

Pumping lemma for regular languages

While the regular languages have nice properties (as we have seen in the early part of the course), unfortunately many languages that we encounter in practice are nonregular. We discuss a method to prove that a given language is nonregular. Such a method is important for understanding the limits of state diagrams and regular expressions. If we can establish nonregularity of a language, there is no need to try construct a regular expression for it.

This material is from Section 9.4 in the textbook.

All regular languages L have the following property:

- corresponding to L there is a constant value called the “pumping length” such that all strings in the language of length at least the “pumping length” can be “pumped”, that is, some substring can be repeated arbitrarily many times and the resulting string must remain in the language L .

The result is stated more formally in the below pumping lemma. The pumping lemma gives a general technique for showing that certain languages are not regular. The proof of the pumping lemma will be discussed in class.

Pumping lemma. For every regular language L there exists a constant n such that any string $x \in L$ of length at least n can be written in three parts

$$x = p \cdot q \cdot r$$

where

$$(P1) \quad q \neq \varepsilon$$

$$(P2) \quad |p \cdot q| \leq n$$

$$(P3) \quad pq^k r \in L \text{ for all } k \geq 0.$$

Note: In the proof of the pumping lemma we can use as n the number of states of a state diagram accepting L . We know that n is a constant, but n can be arbitrarily large.

Examples. The following languages are not regular:

$$\{0^i 1^i \mid i \geq 0\}$$

$\{0^{2^i} \mid i \geq 0\}$ (this language consists of all strings of 0's having a length that is a power of 2)

How would you use the pumping lemma to show that the above languages are not regular?
(Will be done in class.)

Note 2. If L_f is a finite language, no string $x \in L_f$ can be written in three parts $p \cdot q \cdot r$ such that conditions (P1) and (P3) hold because, for large enough k , the strings $pq^k r$ will be longer than the longest string of L_f . Do finite languages satisfy the conditions of the pumping lemma? Why or why not? Are all finite languages regular?

The general method of using the pumping lemma to show that a given language L is not regular can be described as follows. The method uses *proof by contradiction*.

1. For the sake of contradiction we assume that L is regular. Then the pumping lemma gives us the pumping length n . We don't know what n is (it can be arbitrarily large), we only know that it is a constant (positive) integer.
2. Choose a string $x \in L$ of length at least n .
3. Consider all the possible decompositions of x into three parts (as specified in the pumping lemma). If no decomposition of x into three parts satisfies the conditions (P1), (P2), (P3) from the pumping lemma simultaneously, we obtain a *contradiction* and can conclude that L cannot be regular.

The crucial part in using the pumping lemma is usually to select a suitable string $x \in L$ in 2. above. The string x should be selected so that it cannot be pumped (without going “out” of the language L). Note that $x \in L$ has to be specified using the “unknown” constant n .

Above in stage 2., we can choose the string $x \in L$ in a way that we expect to cause problems with the “pumping property”. Note, however, that in the following stage 3. we need to show that no decomposition of x into three parts does satisfies the conditions (P1), (P2), (P3). That is, it is *not* sufficient to consider one particular decomposition.

Sometimes in order to show that given languages are not regular, together with the pumping lemma, we can rely on *closure properties* of the family of regular languages.

The class of regular languages has strong closure properties, in particular, regular languages are closed under the Boolean operations. This means that if S and T are regular languages then also the following languages are regular:

$$S \cup T$$

$$S \cap T$$

$$\overline{S} = \{w \in \Sigma^* \mid w \notin S\}$$

The regularity of $S \cup T$ follows directly from the definition regular expressions. How can you establish would you prove closure under intersection and complement? (Will be done in class.)

Example. Define S to consist of all strings over alphabet the $\{0, 1\}$ that have an equal number of occurrences of 0’s and 1’s. We show that S is not regular.

Proof by contradiction: If S is regular, then also the language

$$S \cap 0^*1^* = \{0^i1^i \mid i \geq 0\}$$

is regular. (Why?) We have in the earlier example shown that the language $\{0^i 1^i \mid i \geq 0\}$ is not regular. We conclude that S cannot be regular.

Example. Let $\Sigma = \{a, b, c\}$ and consider the language L_0 over Σ :

$$L_0 = \{a^k b^i c^i \mid i \geq 0, k \geq 1\} \cup \{b^r c^s \mid r \geq 0, s \geq 0\}.$$

Intuitively, L_0 should not be regular because after a positive number of symbols a , the numbers of b 's and c 's must match and clearly a state diagram cannot count arbitrarily large numbers. However, if we try to use the pumping lemma to prove non-regularity of L_0 , we run into trouble because in a string of the form $a^k b^i c^i$ a non-empty prefix consisting of a 's can always be repeated. Also if $k = 1$ and we try to “pump down” the result will be in the set $\{b^r c^s \mid r \geq 0, s \geq 0\}$.

In fact, directly using the pumping lemma it is not possible to prove the non-regularity of L_0 .¹ That is, L_0 satisfies the pumping lemma but L_0 is not regular (as we show below).

Using closure properties we can easily establish non-regularity of L_0 :

$$L_0 \cap \{ab^r c^s \mid r \geq 0, s \geq 0\} = \{ab^i c^i \mid i \geq 0\}.$$

Using the pumping lemma it is easy to establish that $\{ab^i c^i \mid i \geq 0\}$ is non-regular. Since $\{ab^r c^s \mid r \geq 0, s \geq 0\}$ is denoted by the regular expression ab^*c^* and regular languages are closed under intersection we can conclude that L_0 is non-regular.

¹If you are skeptical, you are welcome to try.