CS 3719 (Theory of Computation and Algorithms) – Lecture 8

Antonina Kolokolova*

January 21, 2011

1 Nonregular languages

A natural question to ask is whether there are languages that are not regular, that is, whether any language (any subset of Σ^* for a finite Σ) can be described by a regular expression. In this lecture, we will show some examples of languages for which it is not possible to build a finite automaton or write a regular expression; we will call such languages nonregular.

How would one go about proving that some language is *not* regular? We cannot check all possible finite automata to make sure that none of them accepts the language, there are infinitely many automata to check. So instead we will identify a property that all regular languages satisfy, and show languages that do not satisfy this property.

Example 1. Consider the automaton accepting all strings with the number of 1s divisible by 3. There is one property you can notice about this automaton: if you take a string it accepts, and repeat any substring of it in which the number of 1s is divisible by 3, you again obtain a string in the language. For example, if 111011010 is in the language, so is 11 1011 1011 010, as well as 111 111 011010. If you look at the automaton, the reason becomes clear: repeating such a substring the automaton returns to the same state. In case of 11 1011 1011 010, the automaton was in the state q_2 after seeing the first 11, and it is again in q_2 after seeing 1011, and back to q_2 after seeing the second block of 1011... and after that its execution on the remaining 010 is the same as it would be without the additional 1011.

This observation will lead us to a property that holds for all regular languages.

Lemma 7 (Pumping lemma)**.** *If A is a regular language, then there exists a number $p \in \mathbb{N}$ called "pumping length" such that for any string $s \in A$, if $|s| \geq p$ then $s = xyz$, where*

1) $\forall i \geq 0, xy^iz \in A$. That is, if s is in A than so is any string obtained from s by repeating the block y 0 or more times.

*The material in this set of notes came from many sources, in particular "Introduction to Theory of Computation" by Sipser and course notes of U. of Toronto CS 364.

2) $|y| > 0$. *Otherwise the lemma would be trivial: you can always repeat an empty substring as many time as you like.*

3) $|xy| \leq p$. *This property comes out of the proof of the lemma and helps with some nonregularity proofs.*

Proof. Since A is regular, there exists a DFA D, $\mathcal{L}(D) = A$. Suppose D has p states; this number will be the pumping length.

Consider any string s of length at least p. Since during a computation on a string of length p the DFA goes through a sequence of $p + 1$ states, by the Pigeonhole Principle at least one state in this sequence repeats. Now, take y to be the substring between the first two repetitions of the first repeating state. Then repeating this y 0 or more times does not change the acceptance of s by D: just add or remove the sequence of states of D on y from the corresponding position in the sequence of states. The $|y| > 0$ because this is a DFA with no ϵ-transitions. And the last condition is satisfied because there are at most p states the DFA can go through before repeating a state, again by the Pigeonhole Principle: here, x is a string from the start of s to the first occurrence of the first repeating state. □

Now we will use this lemma to show that several languages are not regular; in particular, we will show that "counting" cannot be done by finite automata.

Example 2. The language $A = \{0^n 1^n\}$ is not regular.

For the sake of contradiction, suppose that it is regular. Then there exists a natural number p such that any string longer than p satisfies the condition of the pumping lemma. Here, we will present a string that is in A, but cannot be "pumped".

Consider $s = 0^p 1^p$. Clearly, $|s| \geq p$. By the lemma, then $s = xyz$ such that $\forall i \geq 0, xy^i z \in A$. There are three possibilities: y can consist just of 0s, or just of 1s, or of a mix of 0s and 1s. In the first case, repeating y would increase the number of 0s, but not the number of 1s, so $xyyz \notin A$. Similarly, if y consists of only 1s, repeating it would make a string not in A Finally, suppose that y consists of some 0s followed by some 1s. But then the string $xyyz$ has some 1s before 0s, which again makes it not in the language.[1]

Therefore, s cannot be written as $s = xyz$ in a way that satisfies the pumping lemma. Thus, the language $\{0^n 1^n\}$ is not regular.

Example 3. The language $A = \{ww | w \in \{0,1\}^*\}$ is not regular. This also holds for any other Σ with $|\Sigma| > 1$.

Again, assume for the sake of contradiction that A is regular; then by the pumping lemma there exists a pumping length p, such that for any $s \in A$, $|s| > p$, $s = xyz$ where y's can be repeated.

[1] Note that the last two cases can be also eliminated by using the $|xy| < p$ condition of the pumping lemma.

Take $s = 0^p10^p1$. By the 3rd condition of the pumping lemma, $|xy| \leq p$. And therefore the repeating part y would consist of only 0s. But adding 0s in the first half of the string will make the first half different from the second half, and thus not in the language.

Note, by the way, that if we would have taken a string $s = 0^p0^p$, then the argument would not work: take a y of a small even length, and it could be pumped indefinitely. So the choice of a string matters: it is not that we need to show that every string in the language cannot be pumped, we just need to present one such string.

Example 4. The language $A = \{w | w \in \{0,1\}^*$ contains the same number of 0s and 1s $\}$ is not regular.

Assume for the sake of contradiction that A is regular; let p be the pumping length. Take $s = 0^p1^p$, like in the first example. But now the case of "mixed 0s and 1s" does not apply: such a string would still be in the language. Here, we do need to refer to the $|xy| \leq p$ condition of the pumping lemma: by that condition, y would have to contain only 0s, so repeating y would make a string with more 0s than 1s.

Also note that here taking a string, for example, $(01)^p$ would not be useful: this string can be pumped.

Example 5. The language $A = \{0^i1^j | i > j\}$ is not regular.

(Again assume A is regular and get p from the pumping lemma). Take a string $s = 0^{p+1}1^p$. Now saying that $|xy| \leq p$ and so y consists of just 0s does not help: increasing the number of 0s would give us a string in the language. However, notice that in the first condition of the pumping lemma $i \geq 0$. That is, a string xz with no repetitions of y should still be in the language (think of skipping the first loop on that first repeating state of the automaton). However, here if we will remove even one 0 from the string the resulting string will not be in the language. Therefore, this language is also not regular.