Lecture Notes 6: Non-regular languages
Raghunath Tewari   IIT Kanpur

Limitations of finite automata.

1. Not all languages are regular.

2. Intuitively languages that "require" some sort of counting are generally not regular.

1   The Pumping Lemma

Theorem 1 (Pumping Lemma)**.** *Let L be a regular language. There exists an integer $p \geq 0$, such that for all $w \in L$ of length at least p, there exists a partition of $w = xyz$ such that $|xy| \leq p$, $|y| > 0$, and for all $i \geq 0$, $xy^i z \in L$.*

Informally pumping lemma says that if L is a regular language then all strings in L having length greater than some quantity, has a non-null substring that can be repeated as many times as you want and the resultant string is still in the language.

Note 1. Pumping lemma is a property of regular languages. In other words if a language is regular then it satisfies the above property. It may very well be true that certain non-regular languages also satisfy the above property.

To prove that certain languages are not regular, we will use the pumping lemma in the contrapositive form.

Theorem 2 (Contrapositive form of Pumping Lemma)**.** *Let L be a language. If*

- *$\forall p \geq 0$,*   *(opponent's move)*

- *$\exists w \in L$ with $|w| \geq p$, such that,*   *(your move)*

- *$\forall$ possible partitions of w as $w = xyz$, satisfying*   *(opponent's move)*

    - *$|xy| \leq p$, and*
    - *$|y| > 0$,*

- *$\exists i \geq 0$ such that $xy^i z \notin L$,*   *(your move)*

then L is not regular.

Think of it as a game between you and an all powerful opponent. Both you and your opponent have to follow the rules of the game (the conditions of the theorem). Your goal is to finally find an i such that $xy^i z$ is not in L and your opponents goal is to choose the partition of w in such a manner that no matter what i you choose, $xy^i z$ will be in L.

Pumping lemma gives a sufficient condition for non-regular languages. It is not a necessary condition. In other words, there exists non-regular languages that do not satisfy the pumping lemma. Therefore,

$$\begin{array}{lll} \text{You WIN} & \implies & \text{L is not regular, but} \\ \text{You LOSE} & \not\implies & \text{L is regular} \end{array}$$

2 Examples of Non-regular Languages

1.
$$L_1 = \{0^n 1^n \mid n \geq 0\}$$

Given $p \geq 0$, pick $w = 0^n 1^n$. Now for any partition of $w = xyz$ such that $|xy| \leq p$ and $|y| > 0$, it must be the case that $y = 0^t$ for some $t > 0$. Therefore $xy^0 z = 0^{p-t} 1^p \notin L_1$ since $t > 0$. Therefore L_1 is not regular.

We will denote the language L_1 as L_{01}.

2.
$$L_2 = \{a^l b^m c^n \mid l, m, n \geq 0 \text{ and } l + m = n\}$$

Given $p \geq 0$, pick $w = a^p b^p c^{2p}$. Now for any partition of $w = xyz$ such that $|xy| \leq p$ and $|y| > 0$, it must be the case that $y = a^t$ for some $t > 0$. Therefore $xy^2 z = a^{p+t} b^p c^{2p} \notin L_2$ since $t > 0$. Therefore L_2 is not regular.

3.
$$L_3 = \{w \in \{0,1\}^* \mid \#_0(w) = \#_1(w)\}$$

Note that $L_3 \cap L(0^* 1^*) = L_1$. Since we have shown that L_1 is not regular therefore it must be the case that L_3 is also not regular.

Remark. Intersection of a non-regular language with a regular/non-regular language can be both regular or non-regular. Same is true for union as well. Verify for yourself by constructing toy examples.

4.
$$L_4 = \{0^k \mid k \text{ is a prime}\}$$

Given $p \geq 0$, pick $w = 0^q$, where q is the smallest prime greater than or equal to p. Now for any partition of $w = xyz$ such that $|xy| \leq p$ and $|y| > 0$, let $|y| = l$. Then $0 < l \leq p$. Choose $i = q + 1$. Then, $|xy^i z| = q + l(i - 1) = q + lq = q(l + 1)$. Hence $xy^i z \notin L_4$. Therefore L_4 is not regular.

5.
$$L_5 = \{0^i 1^j \mid i \neq j\}$$

Note $L_{01} = \overline{L_5} \cap 0^* 1^*$. Since we know that L_{01} is not regular, therefore L_5 is also not regular.

Exercise 1. Show that the following languages are not regular.

1. $\{0^{n^2} 1^n \mid n \geq 0\}$

2. $\{0^n 1^m \mid n > m\}$

3. $\{ww \mid w \in \{0,1\}^*\}$

4. $\{a^i b^j c^k \mid i \neq 2 \text{ or } j = k\}$

Proof of Theorem 1. Let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA for L. Let $p = |Q|$. Let w be a string of length at least p in L and $|w| = t$. There is a sequence of states $q_0, q_1, \ldots q_t$ that the DFA traverses on the string w where $q_t \in F$. By pigeon hole principle, there exists integers m, n such that $0 \leq m < n \leq q_p$ and $q_m = q_n$. Now consider a partition $w = xyz$, such that

- x is the smallest string having the property that $\delta(q_0, x) = q_m$,

- y is the smallest non-empty string such that $\delta(q_m, y) = q_m$, and

- z is the remaining part of w.

In other words, the DFA traverses from q_0 to q_m on x, traverses from q_m to q_n (q_n is the same as q_m) on y and then proceeds to q_t. Since $m < n$ therefore $|y| > 0$. Now for all $i \geq 0$, $\delta(q_0, xy^i) = q_m$, as the automaton loops on the state q_m on the string y. Therefore $\delta(q_0, xy^i z) = q_t$ and hence $xy^i z \in L$. $\qquad\square$

Note 2. Observe that in the above proof we are "overloading" the definition of δ to accommodate strings as well instead of symbols only. We can formalize this by defining δ recursively as follows:

$$\delta : Q \times \Sigma^* \longrightarrow Q$$

such that

$$
\begin{aligned}
\delta(q, \epsilon) &= q, \\
\delta(q, xa) &= \delta(\delta(q, x), a).
\end{aligned}
$$