Homework 4

Aleck Zhao

March 15, 2018

1. Prove that the following languages are not regular.

   (a) $\{0^n 1^m 0^{n+m} : m, n \geq 0\}$

   Proof. Suppose this language was regular. Then by the pumping lemma, there exists an integer p which is the pumping length. Let $s = 0^p 1^p 0^{2p}$ be in this language. Here, $|s| \geq p$, so there exists a decomposition $s = xyz$ with $|y| > 0$ and $|xy| \leq p$. Thus, y can contain only 0s, so suppose $y = 0^k$. We have

   $$\begin{aligned} s &= xyz = 0^{p-k} 0^k 1^p 0^{2p} \\ \implies s' &= xy^2 z = 0^{p-k} 0^{2k} 1^p 0^{2p} = 0^{p+k} 1^p 2^{2p} \end{aligned}$$

   but s' is not in our language, a contradiction, so the language is not regular. □

   (b) $\left\{w : w \in \{0,1\}^* \text{ is not a palindrome}\right\}$.

   Proof. Consider the complement language $\left\{w \in \{0,1\}^* \text{ is a palindrome}\right\}$. If we show the complement is not regular, then this language is not regular by the closure of regular languages.

   Suppose the complement was regular. Then by the pumping lemma, there exists an integer p which is the pumping length. Let $s = 0^p 110^p$ be in this language. Here $|s| \geq p$, so there exists a decomposition $s = xyz$ with $|y| > 0$ and $|xy| \leq p$. Thus, y can contain only 0s, so suppose $y = 0^k$. We have

   $$\begin{aligned} s &= xyz = 0^{p-k} 0^k 110^p \\ \implies s' &= xy^2 z = 0^{p-k} 0^{2k} 110^p = 0^{p+k} 110^p \end{aligned}$$

   which is not a palindrome, and not in this language. Thus, this complement language is not regular, so the original language is also not regular. □

2. (a) Let $B = \left\{1^k y : y \in \{0,1\}^* \text{ and } y \text{ contains at least } k \text{ 1s, for any } k \geq 1\right\}$. Is B a regular language? Prove your answer.

Proof. The only strings not in B are those starting with 0 and the strings of the form 10^k. Let L_1 be the set of strings starting with 0, and L_2 be the set of strings of the form 10^k. Then L_1 and L_2 are clearly regular languages (it is trivial to design DFAs that recognize these), and thus $L_1 \cup L_2$ is regular, so $B = (L_1 \cup L_2)^c$ is also regular, as desired. □

(b) Let $C = \left\{1^k y : y \in \{0,1\}^* \text{ and } y \text{ contains at most } k \text{ 1s, for any } k \geq 1\right\}$. Is C a regular language? Prove your answer.

Proof. Suppose C was a regular language. Then by the pumping lemma there exists an integer p which is the pumping length. Let $s = 1^p 0 1^p \in C$. Here $|s| \geq p$, so there exists a decomposition $s = xyz$ with $|y| > 0$ and $|xy| \leq p$. Thus, y can contain only 1s, so suppose $y = 1^n$. we have

$$\begin{aligned} s &= xys = 1^{p-n} 1^n 0 1^p \\ \implies s' &= xy^0 z = 1^{p-n} 0 1^p \end{aligned}$$

which is not in C, because k can be at most $p - n$, but then the rest of the string would contain at least p 1s. This is a contradiction, so C is not regular. □

3. Let $\Sigma = \{0, 1\}$. Say that a language $L \subset \Sigma^*$ satisfies property (a) if the following holds: there exists a string $w \in L$ such that $0w \in L$. Give an (efficient) algorithm that takes as input any DFA M, tests if $L(M)$ satisfies property (a). Describe your algorithm and briefly explain (informally) why it is correct.

   Solution. Given a DFA M, construct a new DFA N with a new start state that transitions to the original start state along 0, and transitions to a garbage state along 1, keeping everything else the same. Then it's clear that N accepts all strings $0w$ where $w \in L(M)$. We require that $L = L(M) \cap L(N) \neq \varnothing$. Since regular languages are closed under intersection, L can be recognized by some DFA, which we can explicitly construct. From there, we just need to check that this DFA accepts some string, which can be done using a graph DFS from the start state. If there is a path to an accept state, then $L(M)$ satisfies property (a), and otherwise, it does not. □

4. Give a context-free grammar that generates the following language, where the alphabet Σ is $\{0, 1\}$: $\{w : w$ is not empty ands starts and ends with the same symbol$\}$.

   Solution. Let E be the start variable. Then we have the rules

$$E \to E1 \mid E0 \mid \varepsilon$$
$$F \to 0 \mid 1 \mid 0E0 \mid 1E1$$

   Here, the first rule generates any string, and the second rule generates any string that has either 0s on both ends or 1s on both ends, or is just a single element 0 or 1. $\square$