CS 181 Spring 2020 Homework Week 5 Solutions

1. **Proof idea** As always we try to see if it is possible to make case analysis simple by picking s such that the first p characters are simple, e.g. just p copies of the same symbol. We know that the rule from *the pumping lemma* $|xy| \leq p$ will then allow us to argue that y consists of only this character. Then we can pump up or down to change the count of that character; in this scenario we want to create an input where we can pump it so that we no longer have more than half of the characters be c's in the pumped string. So this motivates choosing a string such that the number of c's is just barely a majority of c's. I.e., the number of c's is just one more than the rest of the symbols combined. That way, any decrease in the number of c's will cause the string to no longer be in the language. A string, s, that would work well is to have all the c's at the beginning of the string, and the rest consists of symbols except c. Then we can pump down the c's to generate a string that does not have a majority of c's. This is the intuition, and now we present the actual solution.

   Proof. Suppose this language is FS. Then by *the pumping lemma* there exists a pumping length $p > 0$ for this language. Consider the string in the language $s = c^p a^{p-1} \in L_1$ of length at least p. By the pumping lemma we know that there is a partition $s = xyz$ such that

   $$\begin{aligned} &|y| > 0, \\ &|xy| \leq p, \\ &xy^i z \in L_1 \quad \forall i = 0, 1, 2, \ldots. \end{aligned}$$

   Since one of the conditions of the pumping lemma is that $|xy| \leq p$, it must be the case that y is a string of c characters since it comes from the beginning of the string. Moreover $|y| > 0$ tells us that this string has a nonzero number of c characters, say $y = c^\beta$. Pumping down (i.e. selecting $i = 0$) gets rid of these nonzero number of c characters in the original string. In the new string the number of c characters will be *at most* $p - 1$, with *the same* number of a characters coming afterwards, i.e. $p - 1$. Thus c does not have a strict majority of characters, and so this pumped down string cannot be in the language. In other words,

   $$xz = c^{p-\beta} a^{p-1} \notin L_1 \qquad \text{because } p - \beta \leq p - 1.$$

   This is a contradiction, so the language is not finite-state. $\square$