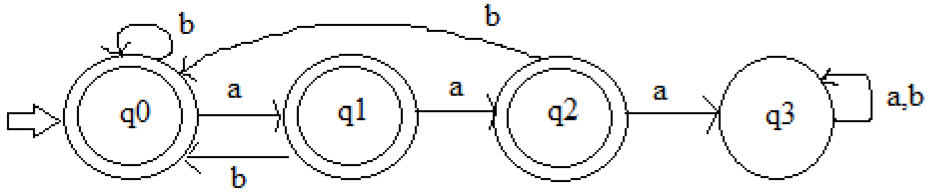
COT 4210 Final Exam Part A: Regular Languages 5/4/2021 Solutions

1) (5 pts) Design a DFA over the alphabet $\Sigma = \{a, b\}$ that accepts all strings that have no more than 2 a's in a row, and no other strings. Either draw the resulting DFA or give the formal result clearly specifying each portion of the 5-tuple definition clearly typed.

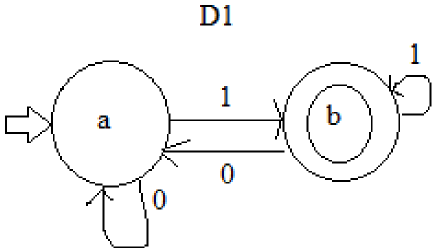
Solution

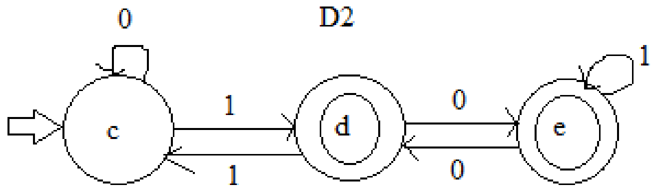
The key to this design is just to have the states keep track of the current number of a's in a row, treating three a's in a row as a permanent reject state. Here is one possible DFA for the language:



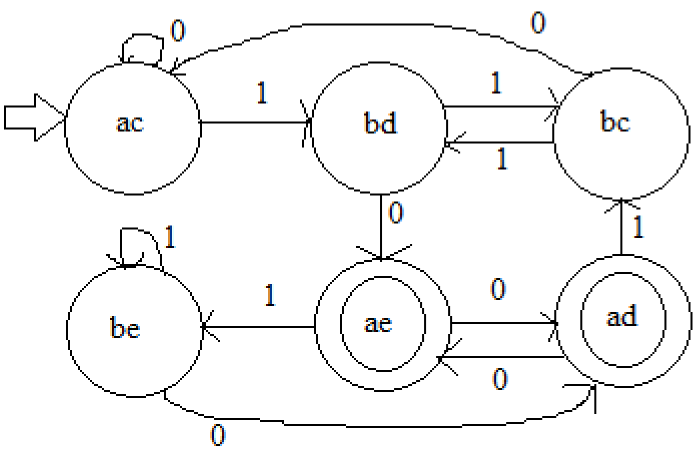
Grading: 1 pt valid DFA, 2 pts for accepting all strings it should, 2 pts for rejecting all strings it should, can give partial for the last two parts.

2) (10 pts) Below are the drawings of two DFAs: D1 and D2, both over the alphabet {0, 1}. Let L(D1) and L(D2) represent the language described by D1 and the language described by D2, respectively. Use the construction algorithm mentioned in class to design a DFA for the language $\overline{L(D1)} \cap L(D2)$. The states in D1 are {a, b} and the states in D2 are {c, d, e}. Please label the states in your answer as {ac, ad, ae, bc, bd, be}. You may represent your answer as either a drawing or in the formal manner with each item in the 5-tuple definition clearly typed. (Both forms must unambiguously convey the same information.)





Solution



Grading: 2 pts for having 6 states labeled ac, ad, ae, bc, bd, be.

1 pt for labeling ac as the start state

2 pts for labeling ae and ad as accept states (give 0 if these aren't exactly the 2 accept states labeled)

5 pts total for the transitions - eyeball and give a proportional score

3) (10 pts) Let $\Sigma = \{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}\}$. Define L = { w | the binary string read left to right of the top of the tiles is the reverse of the binary string on the bottom of the tiles }. For example, $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} \in L$, since the top of the tiles reads "11001" and the bottom of the tiles reads "10011". If we reverse "11001", we obtain "10011". Prove, via the pumping lemma for regular languages, that L is not regular.

Solution

According to the pumping lemma, if a language is regular, then for all strings s of length p or greater in the language, where p is the pumping length for the language, there exists a way to split s into three pieces s = xyz, such that for all non-negative integers i, xy^iz is in the language, $|xy| \le p$ and $|y| > 0$. We will prove that the string s = $\begin{bmatrix}0\\1\end{bmatrix}^p \begin{bmatrix}1\\0\end{bmatrix}^p$, which is a string in L of length p or greater does NOT satisfy the pumping lemma. Once this is proven, we can conclude that L is not a regular language.

First, we can verify that this string is in the language because the top of the tiles read 0^p1^p and the bottom tiles in reverse read 0^p1^p as well.

Now, let's consider all ways to split s = xyz with $|xy| \le p$ and $|y| > 0$. It is necessarily the case that:

x = $\begin{bmatrix}0\\1\end{bmatrix}^i$, for some non-negative integer i and y = $\begin{bmatrix}0\\1\end{bmatrix}^j$, for some positive integer j and that $i+j \le p$.

Now, consider the string xy^2z = $\begin{bmatrix}0\\1\end{bmatrix}^{p+j} \begin{bmatrix}1\\0\end{bmatrix}^p$, since we are simply adding a copy of y in the middle of the string. This string isn't in L because the top of the tiles read $0^{p+j}1^p$ but the bottom tiles in reverse read 0^p1^{p+j} and we know $j > 0$, so the $(p+1)^{st}$ character from the left on the top will be 0, but the $(p+1)^{st}$ character from the right on the bottom will be 0. Thus, this proves that xy^2z can not be in the language no matter how we select x, y and z. It follows that L is not regular, as desired.

Grading: 2 pts for picking a string of length p or greater that is in L.
3 pts for considering all splits
2 pts for picking a string of the form xy^iz to examine
3 pts for proving that the string selected is not in L.