MAU22C00: TUTORIAL 9 SOLUTIONS
FORMAL LANGUAGES AND GRAMMARS

1) Let L be the language over the alphabet $A = \{0, 1\}$ consisting of all words where the string 00 occurs as a substring.

(a) Prove from the definition of a regular language that the language L is regular.

(b) Draw a finite state acceptor that accepts the language L. Carefully label all the states including the starting state and the finishing states as well as all the transitions. Make sure you justify it accepts all strings in the language L and no others.

(c) Write down the transition mapping of the finite state acceptor you drew in the previous part of the problem.

(d) Let $\equiv_L$ be the equivalence relation defined in Lecture 23 before the statement and proof of the Myhill-Nerode theorem. Determine the equivalence classes into which this equivalence relation partitions L.

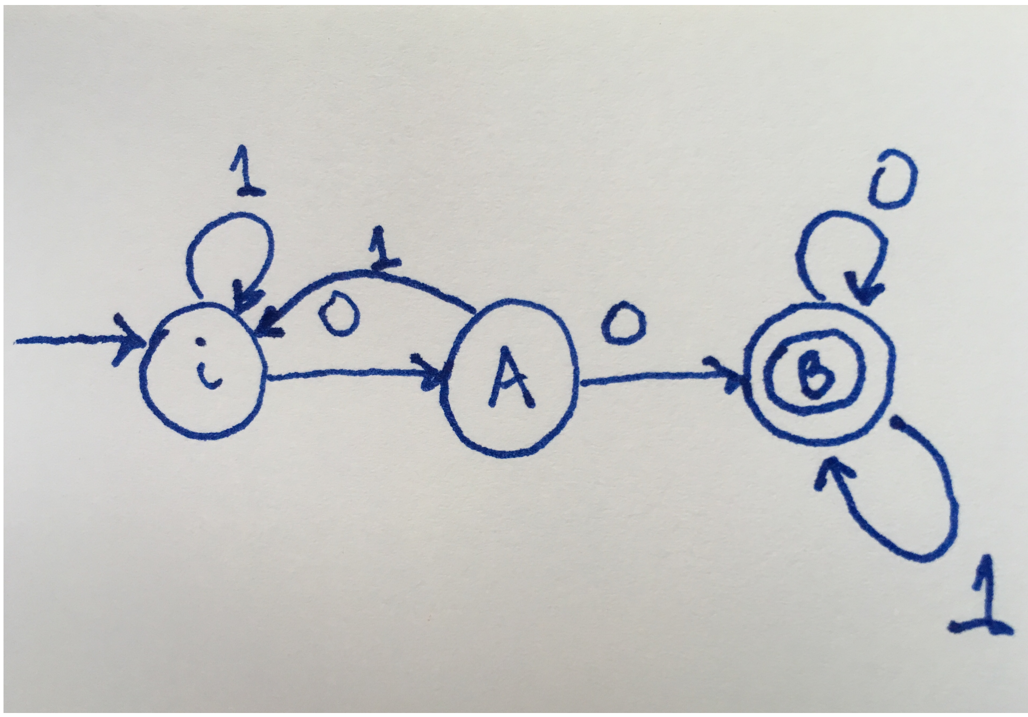
Solution: (a) Let the alphabet $A = \{0, 1\}$. Recall that the definition of a regular language allows for finite subsets of A^*, the Kleene star, concatenations, and unions. Note that

$$L = \{w \in A^* \mid w = u \circ 00 \circ v \quad u, v \in A^*\}.$$

Therefore, we can let $L_1 = \{00\}$ be the language consisting of just the string 00 of interest. L_1 is a finite set, so it is allowed in the definition of a regular language. Let $L_2 = \{0, 1\}$. L_2 is finite, hence likewise allowed. Let $L_3 = L_2^*$, the Kleene star applied to L_2. The language $L_3 = A^*$, i.e. it is the set of all words that can be formed over the alphabet $A = \{0, 1\}$. Set $L_4 = L_3 \circ L_1$, and then $L_5 = L_4 \circ L_3$. Note that the words in L_5 have exactly the structure of the words in L, and in fact, $L = L_5$. Note also that the solution here is by no means unique. No two of you will necessarily have arrived at the same exact expression, order or labelling of the intermediate languages L_i that come into the definition of a regular language as applied to L.

(b) See diagram of finite state acceptor on the next page.

We can use three states $\{i, A, B\}$, where i is the initial state. Since we must ensure the word contains the string 00, when 1 is the input, we stay in the initial state i. For input 0, we move to a new state A. A is not an accepting state as we have so far only half of the string 00, the first zero. If we get input 1, we have to restart the process of capturing the string 00, so we get back to the initial state i. If we get input 0, then we will have received the second zero we want, so we'll move to a new state B, which is an accepting state. Once we have the substring 00, we don't care what follows, so the transitions for both 0 and 1 out of state B are back into B itself.



(c)

$$\begin{aligned} t(i,0) &= A \qquad & t(i,1) &= i \\ t(A,0) &= B \qquad & t(A,1) &= i \\ t(B,0) &= B \qquad & t(B,1) &= B \end{aligned}$$

(d) Recall from Lecture 23 that the equivalence relation $\equiv_L$ is defined as $x \equiv_L y$ if $\forall w \in A^*, xw \in L \Leftrightarrow yw \in L$, where $x, y \in L$. As we saw in lecture, if $x \equiv_L y$, then x and y place our finite state acceptor into the <u>same state</u> s. We also know that a word is in the language L if it places the finite state acceptor into an accepting state. We have only one accepting state, which is B. Therefore, we have only one equivalence class, which is all of L since every word $x \in L$ will place the finite state acceptor into state B.

If we were to extend the definition of our equivalence relation to $x \equiv_L y$ if $\forall w \in A^*, xw \in L \Leftrightarrow yw \in L$, where $x, y \in A^*$, namely if we were to look at the equivalence classes that $\equiv_L$ determines on all of A^*, then the question would be a lot more interesting as then we would determine which words in A^* place our finite state acceptor into each of the states i, A, and B. For B we already have the answer. The equivalence class is exactly our language L. The words in A^* that place the finite state acceptor into state A are all words that do not contain the string 00 and end in 0. Finally, the words in A^* that place the finite state acceptor into state i are all words that do not contain the string 00 and end in 1.