# Midterm Solutions, CPSC 421/501, 2017W1

N. Bayless, J. Friedman, R. Stiyer, and S. Yang

## 1

1. True: the intersection of two regular languages is regular; see Sections 1.1 and 1.2 of [Sip].

2. False: for example, the intersection of the empty language with any nonregular language is the empty language.

3. False: for example, the language $L = \{1^{n^2} \mid n \in \mathbb{Z}_{\geq 0}\}$ is nonregular, but $L^* = 1^*$.

4. True: if $L_1$ is regular, then $L_2$ is regular implies that $L_1 \cap L_2$ is regular (see the first question). Hence if $L_1$ is regular, then if $L_1 \cap L_2$ is nonregular then $L_2$ cannot be regular.

5. True: if $L$ is recognized by a DFA, from the DFA one can build a Turing machine that moves right and transitions through states just like the DFA; upon reading a blank symbol the Turing machine transitions to $q_{\text{accept}}$ or $q_{\text{reject}}$ according to whether or not the current state is a final state of the DFA. (See also Problem 2 of this exam: this involves a regular language for which you are asked to give a Turing machine that recognizes it.)

## 2

The transition diagram below uses a condensed version of the notation used by Sipser: an edge labeled $\{0,1\} \to R$ is equivalent to an edge labeled with both $0 \to R$ and $1 \to R$. This, of course, is a further simplification of the full $\delta$ function definition, and is equivalent to writing $\delta(q_{start}, 0) = (q_{end}, 0, R)$

and $\delta(q_{start}, 1) = (q_{end}, 1, R)$, where $q_{start}$ is the state from which the edge leaves and $q_{end}$ is the state in which the edge terminates.

Note that this language is regular, so a Turing machine which decides this language will not need to backtrack (i.e. move left). Since we want to accept only strings whose length is divisible by 3, we need 5 states: one state each for the possible remainders modulo 3 ($q_0$ for 0 mod 3, $q_1$ for 1 mod 3, and $q_2$ for 2 mod 3), an accepting state ($q_{acc}$), and a rejecting state ($q_{rej}$). It doesn't matter what symbols make up the string as long as its length is divisible by 3, so the behaviour should be same whether the next symbol read is a 0 or a 1.

We start before having read the first symbol on the tape. If the first symbol is a blank symbol, $\sqcup$, then the input string has length 0. 0 is divisible by 3, so we should accept this string. If the first symbol is not a blank symbol, the length of the string that has been read so far is now 1, which is equal to 1 mod 3, so we move the next state. Continuing in this fashion, we will read each subsequent symbol and cycle through states $q_0$, $q_1$, and $q_2$ as the length of the string that has been read increases. If we are in either state $q_1$ or $q_2$ and encounter a blank symbol, this means we have reached the end of the input string and that its length is not divisible by 3, so we transition to the rejecting state. If we encounter the blank symbol for state $q_0$, however, it means the input string's length is divisble by 3, so we transition to the accepting state, as in the case of the empty string. In this way, we can see that the Turing machine will only accept inputs whose length is divisible by 3 and will reject all other input.

So, we can formally define the Turing machine as follows:

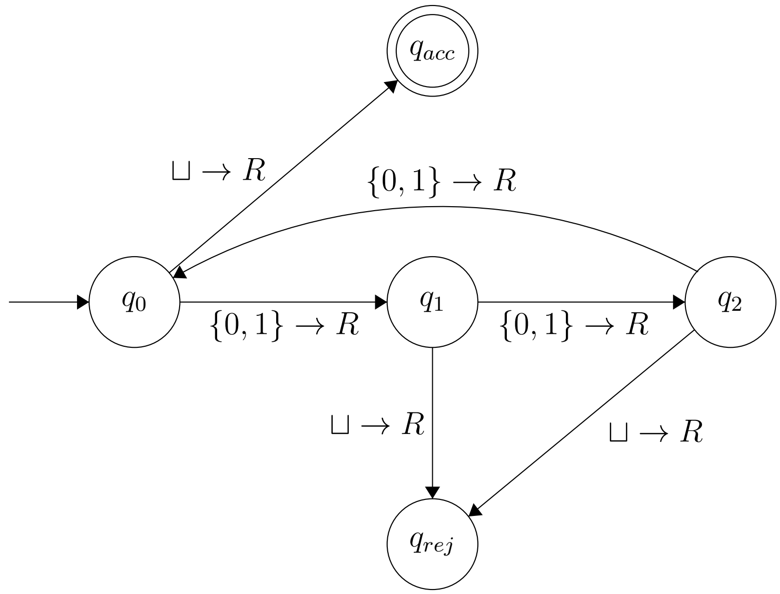
$Q = \{q_0, q_1, q_2, q_{acc}, q_{rej}\}$

$\Sigma = \{0, 1\}$

$\Gamma = \Sigma \cup \{\sqcup\}$

$q_0, q_{acc}, q_{rej}$ are the initial, accepting, and rejecting states as expected

$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ is given by the transition diagram below




# 3

## 3.1

Assume—for the sake of contradiction—that there is a DFA with five states. Then according to the pumping lemma, the word $1^5 \in L$ can be written as $xyz$ such that $y \neq \epsilon$ and $xy^iz \in L$ for all $i$. But then $xy^2z \in L$ and yet $xy^2z$ has length at least six, which is impossible.

## 3.2

We compute

1. $\mathrm{AF}(L^*, \epsilon) = L$ (where AF means AcceptingFutures);
2. $\mathrm{AF}(L^*, 1) = \{1^n \mid n = 2, 4, 5 \text{ or } n \geq 7\}$;
3. $\mathrm{AF}(L^*, 1^2) = \{1^n \mid n = 1, 3, 4 \text{ or } n \geq 6\}$;
4. $\mathrm{AF}(L^*, 1^3) = \{1^n \mid n = 0, 2, 3 \text{ or } n \geq 5\}$;
5. $\mathrm{AF}(L^*, 1^4) = \{1^n \mid n = 1, 2 \text{ or } n \geq 4\}$;

6. $\mathrm{AF}(L^*, 1^5) = \{1^n \mid n = 0, 1 \text{ or } n \geq 3\}$;

7. $\mathrm{AF}(L^*, 1^6) = \{1^n \mid n = 0 \text{ or } n \geq 2\}$;

8. $\mathrm{AF}(L^*, 1^7) = \{1^n \mid n \geq 1\}$;

9. $\mathrm{AF}(L^*, 1^8) = 1^*$.

For $n = 0, 1, \ldots 7$ we see that the longest word not in $\mathrm{AF}(L^*, 1^n)$ is $1^{7-n}$, and there is no word not in $\mathrm{AF}(L^*, 1^8)$. Hence all these values of $\mathrm{AF}(L^*, 1^n)$ are distinct; hence, by the Myhill-Nerode theorem, any DFA recognizing $L$ must have at least nine states.

## 3.3

Here are some examples.