

Worth: 8%

Due: By 6pm on Wednesday 2 December

For each question, please write up detailed answers carefully. Make sure that you use notation and terminology correctly, and that you explain and justify what you are doing. Marks **will** be deducted for incorrect or ambiguous use of notation and terminology, and for making incorrect, unjustified, ambiguous, or vague claims in your solutions.

- [20] 1. In *dyadic* notation, natural numbers are represented in base 2 using the digits 1 and 2 (instead of 0 and 1). For example, 34 is represented by the dyadic string “11122” because  $34 = 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 2 \cdot 2 + 2 \cdot 1$ . Note that each integer has a *unique* dyadic representation, because there cannot be any leading digits. This is unlike binary notation, where each integer has infinitely many representations (*e.g.*, 34 is represented by each of the infinitely many binary strings “100010”, “0100010”, “00100010”, ...). In particular, the unique dyadic representation of 0 is the empty string ( $\varepsilon$ ). For any dyadic string  $s \in \{1, 2\}^*$ ,  $\text{val}(s)$  represents the decimal value of  $s$  (*e.g.*,  $\text{val}(\text{“122”}) = 10$  and  $\text{val}(\text{“”}) = 0$ ).

- [10] (a) Construct a DFA  $A_1$  such that  $A_1$  accepts exactly those dyadic strings whose value is a multiple of 5, *i.e.*,

$$\mathcal{L}(A_1) = \{s \in \{1, 2\}^* : \text{val}(s) \bmod 5 = 0\}.$$

Justify that  $A_1$  is correct by stating explicit properties  $P_q(s)$  for each state  $q$  of  $A_1$ , such that for all strings  $s \in \{1, 2\}^*$ ,  $\delta^*(q_0, s) = q \Leftrightarrow P_q(s)$  (where  $q_0$  is the initial state of  $A_1$ ). **Do not write a formal proof of correctness for  $A_1$ . Simply give the properties for each state.**

(HINT: Figure out how the values  $\text{val}(s1)$  and  $\text{val}(s2)$  relate to  $\text{val}(s)$  for any dyadic string  $s$ . Then, consider what you know about the possible values of  $x \bmod 5$ , for any integer  $x$ , and use this to help in the design of your DFA.)

- [10] (b) Construct a regular expression  $R_1$  such that  $\mathcal{L}(R_1) = \mathcal{L}(A_1)$ .

Justify that  $R_1$  has this property. If you are using a construction from class, show your work, *i.e.*, explain what you are doing at each step of your construction.

- [20] 2. Consider the language  $L_2 = \{x \in \{0, 1\}^* : x \text{ does not contain } 000\}$ . Give a regular expression  $R_2$  such that  $L_2 = \mathcal{L}(R_2)$ . Write a careful, detailed proof that  $L_2 = \mathcal{L}(R_2)$ .

To some degree, your mark will be inversely proportional to the length of your regular expression, *i.e.*, you will be rewarded for coming up with a regular expression as short as possible.

- [20] 3. Let  $\Sigma = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$  be the alphabet of all  $2 \times 1$  matrices consisting of elements from  $\{0, 1\}$ . A string of symbols in  $\Sigma$  gives two rows of 0s and 1s. Consider each row to be a natural number written in binary, and let

$$L_3 = \{x \in \Sigma^* : \text{the bottom row of } x \text{ is smaller than the top row}\}.$$

For example,  $\begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in L_3$  because  $1010 > 1001$ , but  $\begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \notin L_3$  since  $010 < 101$ . Construct a DFA  $A_3$  that accepts  $L_3$  and write a careful, detailed proof that  $\mathcal{L}(A_3) = L_3$ .

To some degree, your mark will be inversely proportional to the size of your DFA, *i.e.*, you will be rewarded for coming up with a DFA as small as possible.

- [20] 4. Consider  $L_4 = \{0^n 1^m 2^{n-m} : n \geq m \geq 0\}$ .

- [10] (a) Write a careful, detailed proof that  $L_4$  is not regular.

- (b) **BONUS! See the bulletin board announcement for details.**

Give a context-free grammar that generates  $L_4$ . Write a careful argument that your CFG is correct, *i.e.*, that it generates every string in  $L_4$  but no string outside  $L_4$ .