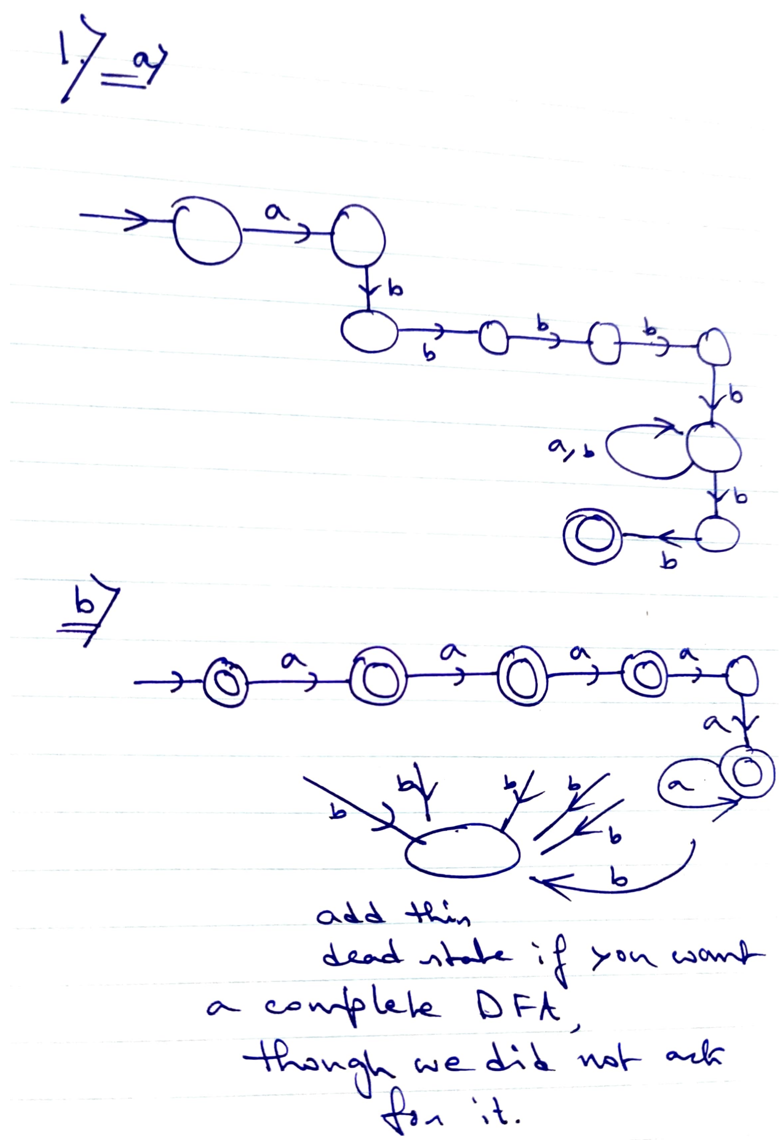
Formal Language And Automata Theory (CS21004) - Class Test 1
IIT Kharagpur, CSE Dept., Spring'23

Answer all questions. In case of reasonable doubt, make practical assumptions.
Marks will be deducted for sketchy proofs and claims without proper reasoning.

Marks = 20 | Time = 1 hour

1. Construct a DFA for
   (a) $L = \{ab^5wb^2 \mid w \in \{a,b\}^*\}$. **[2.5]**
   (b) $L = \{a^n \mid n \geq 0, n \neq 4\}$. **[2.5]**
   Solution: 1(a) solution shown is NFA. On this apply NFA-2-DFA conversion technique. In



order to make it simple, you can just apply the technique on the NFA part : wb^2 and then attach the previous part of the original NFA to the converted DFA. (Surely, alternate answers if correct will also be awarded marks, goes for any question in general.)

2. Let L be a regular language on some alphabet Σ and let $\Sigma_1 \subset \Sigma$ be some smaller alphabet. Prove that $L_1 = L \cap \Sigma_1^*$ is also regular. **[5]**
   Solution: Note that Σ_1^* is a regular language over the alphabet Σ_1. Since, $\Sigma_1 \subset \Sigma$, Σ_1^* is also

a regular language over the alphabet Σ. This can be argued as follows. There exists a FA for Σ_1^* which has one initial=accept state and a self loop with all symbols $\in \Sigma_1$. Add a dead state to this automaton such that for all symbols $\in \Sigma \setminus \Sigma_1$, the FA transitions to it from the initial state. This is now an FA for Σ_1^*, but defined over the alphabet Σ. Since regular languages are closed over concatenation, $L_1 = L \cap \Sigma_1^*$ is also regular over the alphabet Σ.

3. For each of the following languages, construct a regular expression that generates it:

   (a) the set of binary strings that have both 00 and 11 as substrings; **[2.5]**
   Solution: $(1+0)^*((11(1+0)^*00)+(00(1+0)^*11))(1+0)^*$ is the language of strings containing 11 and 00.

   (b) the set of strings over the alphabet {x, y, z} in which each y is immediately followed by x; **[2.5]**
   Solution: $(x + yx + z)^*$

4. For languages A and B, let the perfect shuffle of A and B be the language

$$\{w \mid w = a_1 b_1 \cdots a_k b_k, \text{where } a_1 \cdots a_k \in A \text{ and } b_1 \cdots b_k \in B, \text{ each } a_i, b_i \in \Sigma\}$$

   Show that the class of regular languages is closed under perfect shuffle. **[5]**

Answer: Let $D_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ and $D_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ be two DFAs that recognize A and B, respectively. Here, we shall construct a DFA $D = (Q, \Sigma, \delta, q, F)$ that recognizes the perfect shuffle of A and B.

The key idea is to design D to alternately switch from running D_A and running D_B after each character is read. Therefore, at any time, D needs to keep track of (i) the current states of D_A and D_B and (ii) whether the next character of the input string should be matched in D_A or in D_B. Then, when a character is read, depending on which DFA should match the character, D makes a move in the corresponding DFA accordingly. After the whole string is processed, if both DFAs are in the accept states, the input string is accepted; otherwise, the input string is rejected.

Formally, the DFA D can be defined as follows:

(a) $Q = Q_A \times Q_B \times \{A, B\}$, which keeps track of all possible current states of D_A and D_B, and which DFA to match.

(b) $q = (q_A, q_B, A)$, which states that D starts with D_A in q_A, D_B in q_B, and the next character read should be in D_A.

(c) $F = F_A \times F_B \times \{A\}$, which states that D accepts the string if both D_A and D_B are in accept states, and the next character read should be in D_A (i.e., last character was read in D_B).

(d) δ is as follows:

   i. $\delta((x, y, A), a) = (\delta_A(x, a), y, B)$, which states that if current state of D_A is x, the current state of D_B is y, and the next character read is in D_A, then when a is read as the next character, we should change the current state of A to $\delta_A(x, a)$, while the current state of B is not changed, and the next character read will be in D_B.

   ii. Similarly, $\delta((x, y, B), b) = (x, \delta_B(y, b), A)$.