1. [10 marks] A run is a maximal block of one or more consecutive identical symbols inside a string. For example, aabccc has 3 runs and 01010 has 5 runs.

Give a regular expression for those strings over the alphabet \{0, 1, 2\} having exactly three runs. Justify briefly.

2. [10 marks] Draw the transition diagram of a DFA accepting precisely those strings in problem 1. Be sure that it is a complete DFA: you draw a transition from every state on every symbol. Justify briefly.

3. [10 marks]
   (a) [3 marks] Describe an algorithm \(E\) that takes a regular expression \(r\) as input, and returns true if \(\epsilon \in L(r)\), and false otherwise. We don’t care about efficiency (although this part can be done efficiently). Here you can assume that \(r\) uses only the operators concatenation, union, Kleene star, and positive closure. Justify correctness briefly.

   (b) [7 marks] Describe an algorithm \(A\) that takes a regular expression \(r\) as input, and returns a new regular expression \(r'\) such that \(L(r') = L(r) - \{\epsilon\}\). Here you can assume that \(r\) uses only the operators concatenation, union, Kleene star, and positive closure. Of course, you can use the algorithm in (a) as a subroutine.

   However, the output \(r'\) must use only the operators concatenation, union, and positive closure, and may not use any occurrences of the symbol \(\epsilon\). Kleene star is not allowed.

   Justify correctness briefly.

   In part (b) we are only concerned with whether your algorithm is correct and well-described, not how efficient it is. (It’s actually rather hard to make it efficient!)