7 (100 pts.) Draw me a giraffe.

For each of the following languages in 7.A.–7.C., draw an NFA that accepts them. Your automata should have a small number of states. Provide a short explanation of your solution, if needed.

7.A. (25 pts.) All strings in \( \{0, 1, 2\}^* \) such that at least one of the symbols 0, 1, or 2 occurs at most 4 times. (Example: 12002120210 is in the language, since 1 occurs 3 times.)

7.B. (25 pts.) \(((01)^*(10)^* + 00)^* \cdot (1 + 00 + \varepsilon) \cdot (11)^*\).

7.C. (25 pts.) All strings in \( \{0, 1\}^* \) such that the last symbol is the same as the third last symbol. (Example: 1100101 is in the language, since the last and the third last symbol are 1.)

7.D. (25 pts.) Use the power-set construction (also called subset construction) to convert your NFA from 7.C. to a DFA. You may omit unreachable states.

8 (100 pts.) Fun with parity.

Given \( L \subseteq \{0, 1\}^* \), define \( \text{even}_0(L) \) to be the set of all strings in \( \{0, 1\}^* \) that can be obtained by taking a string in \( L \) and inserting an even number of 0’s (anywhere in the string). Similarly, define \( \text{odd}_0(L) \) to be the set of all strings \( x \) in \( \{0, 1\}^* \) that can be obtained by taking a string in \( L \) and inserting an odd number of 0’s.

(Example: if \( 01101 \in L \), then \( 01010000100 \in \text{even}_0(L) \).)

(Another example: if \( L \) is \( 1^* \), then \( \text{even}_0(L) \) can be described by the regular expression \( (1^*01^*)^*1^* \).)

The purpose of this question is to show that if \( L \subseteq \{0, 1\}^* \) is regular, then \( \text{even}_0(L) \) and \( \text{odd}_0(L) \) are regular.

8.A. (30 pts.) For each of the base cases of regular expressions \( \emptyset, \varepsilon, 0, \) and 1, give regular expressions for \( \text{even}_0(L(r)) \) and \( \text{odd}_0(L(r)) \).

8.B. (60 pts.) Given regular expressions for \( e_j = \text{even}_0(L(r_j)) \) and \( o_j = \text{odd}_0(L(r_j)) \), for \( j \in \{1, 2 \} \), give regular expressions for

(i) \( \text{even}_0(L(r_1 + r_2)) \)
(ii) \( \text{odd}_0(L(r_1 + r_2)) \)
(iii) \( \text{even}_0(L(r_1r_2)) \)
(iv) \( \text{odd}_0(L(r_1r_2)) \)
(v) \( \text{even}_0(L(r_1^*)) \)
(vi) \( \text{odd}_0(L(r_1^*)) \)

Give brief justification of correctness for each of the above.

8.C. (10 pts.) Using the above, describe (shortly) a recursive algorithm that given a regular expression \( r \), outputs a regular expression for \( \text{even}_0(L(r)) \) (similarly describe the algorithm for computing \( \text{odd}_0(L(r)) \)).
Let binary($i$) denote the binary representation of a positive integer $i$. (Note that the string binary($i$) must start with a 1.)

Given a language $L \subseteq \{0, 1\}^*$, define INC($L$) = \{binary($i + 1$) | binary($i$) $\in$ $L$\}. For the time being assume that $L$ does not contain any string of 1$^*$. (Example: for $L = \{100, 101011, 1101\}$, we have INC($L$) = \{101, 101100, 1110\}).

9.A. (30 pts.) Given a DFA $M = (Q, \Sigma, \delta, s, A)$ for $L$, describe informally (in a few sentences) how to construct an NFA $M_w$ for INC($L$).

9.B. (30 pts.) Given a DFA $M = (Q, \Sigma, \delta, s, A)$ for $L$, describe formally how to construct an NFA $M'$ for INC($L$).

9.C. (30 pts.) Prove formally the correctness of your construction from (9.B.). That is, prove that INC($L$) = $L(M')$.

9.D. (10 pts.) Describe formally how to modify the construction of $M'$ from above, to handle that general case (without the above assumption) that $L$ might also contain strings of the form 1$^*$. You do not need to provide a proof of correctness of the new automata.