

1. Construct an NFA to recognize the language generated by the regular expression  $1^*(0011^*)^*$ .
2. Construct an NFA to recognize the language

$$L_1 = \{w \in \{0, 1\}^* \mid w \text{ does not contain } 111 \text{ as a substring}\}.$$

3. Can all NFAs be “re-designed” to have a single accept state? How would you prove something like that?
4. In many programming languages, there is a notion of a “comment”. Given a finite alphabet  $\Sigma$ , we can extend it by adding the symbols  $/$  and  $*$ . A comment is then a string that starts with the characters  $/*$  and ends with the first appearance  $*/$  after the initial  $/*$ . Anything can be in between *except* the ending sequence  $*/$  itself. For example,  $/*///*/$  is a valid comment, while  $/*/$  is **not** a valid comment.

Can you design an NFA to accept all valid comments?

5. Is the complement of an automatic language still automatic? Why is that? Can you repeat this argument for languages recognized by NFAs?
6. For a string  $w \in \{0, 1\}^*$ ,  $\text{rev}(w)$  is  $w$  written backwards (or  $\varepsilon$  if  $w = \varepsilon$ ). For a language  $L$  recognized by some DFA  $D$ , define  $\text{rev}(L) := \{\text{rev}(x) : x \in L\}$ . Can you design an NFA to recognize  $\text{rev}(L)$ ?