

*Starred subproblems cover advanced material that will *not* appear on homeworks or exams.

1. Let $L = \{w \in \{0, 1\}^* \mid w \text{ starts and ends with } 0\}$.
 - (a) Construct an NFA for L with exactly three states.
 - (b) Convert the NFA you just constructed into a DFA using the incremental subset construction. Draw the resulting DFA. Your DFA should have four states, all reachable from the start state.
 - (c) Convert the DFA you constructed in part (b) into a regular expression using the state elimination algorithm.
 - (d) Write a simpler regular expression for L .

2. Let L be the set of all strings that contain either 001 or 011 as a substring.
 - (a) Construct an NFA for L with exactly four states.
 - (b) Convert the NFA you just constructed into a DFA using the incremental subset construction. Draw the resulting DFA. Your DFA should have eight states, all reachable from the start state.
 - (c) Convert the NFA you constructed in part (a) into a regular expression using the state elimination algorithm.

3.
 - (a) Convert the regular expression $(0^*1 + 01^*)^*$ into an NFA using Thompson's algorithm.
 - (b) Convert the NFA you just constructed into a DFA using the incremental subset construction. Draw the resulting DFA. Your DFA should have four states, all reachable from the start state. (Some of these states are obviously equivalent, but keep them separate.)
 - (c) Convert the DFA you constructed in part (b) into a regular expression using the state elimination algorithm. You should not get the same regular expression you started with.
 - * (d) **Work on this later:** Find the smallest DFA that is equivalent to your DFA from part (b), using Moore's algorithm (in Section 3.6 of the notes).
 - * (e) **Work on this later:** Convert the minimal DFA from part (d) into a regular expression using the state elimination algorithm. Again, you should not get the same regular expression you started with.
 - (f) What is this language?

4. **Work on this later:**
 - (a) Convert the regular expression $(\epsilon + (0 + 11)^*0)1(11)^*$ into an NFA using Thompson's algorithm.
 - (b) Convert the NFA you just constructed into a DFA using the incremental subset construction. Draw the resulting DFA. Your DFA should have six states, all reachable from the start state. (Some of these states are obviously equivalent, but keep them separate.)

- (c) Convert the DFA you constructed in part (b) into a regular expression using the state elimination algorithm. You should *not* get the same regular expression you started with.
- * (d) Find the smallest DFA that is equivalent to your DFA from part (b), using Moore's algorithm (in Section 3.6 of the notes).
- * (e) Convert the minimal DFA from part (d) into a regular expression using the state elimination algorithm. Again, you should *not* get the same regular expression you started with.
- (f) What is this language?