

1.
 - (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) **Think about later:** Convert the DFA you constructed in part (b) into a regular expression using Han and Wood's algorithm. You should *not* get the same regular expression you started with.
 - (d) **Think about later:** Find the smallest DFA that is equivalent to your DFA from part (b) and convert that smaller DFA into a regular expression using Han and Wood's algorithm. Again, you should *not* get the same regular expression you started with.
 - (e) What is this language?

2.
 - (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) **Think about later:** Convert the DFA you constructed in part (b) into a regular expression using Han and Wood's algorithm. You should *not* get the same regular expression you started with.
 - (d) **Think about later:** Find the smallest DFA that is equivalent to your DFA from part (b), using Moore's algorithm (in Section 3.10 of the notes), and convert that smaller DFA into a regular expression using Han and Wood's algorithm. Again, you should *not* get the same regular expression you started with.
 - (e) What is this language?