ECE 573
Problem Set 1: Solutions

1. We first define a few helper expressions, which we will use to construct the larger regular expression.

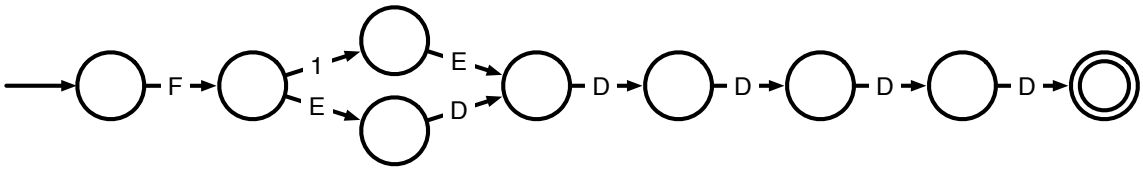
$$\begin{array}{lcll} D & \triangleq & [0-9] & \text{All digits, 0 through 9} \\ E & \triangleq & [0, 2-9] & \text{All digits except 1} \\ F & \triangleq & [2-9] & \text{All digits except 0 and 1} \end{array}$$

   We can now define the regular expression for phone numbers:

$$F(ED|1E)D^4$$

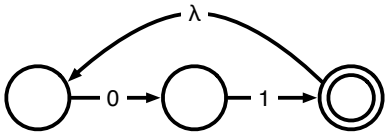
   In English, the number should start with a digit other than 0 or 1. The number should then should continue with a two digit number which is not 11 (which we represent by a number that either does not start with 1, or does start with 1 but is not 11). Finally, the number should end with 4 digits, which can be anything.

2. The DFA accepting this regular expression is given below.

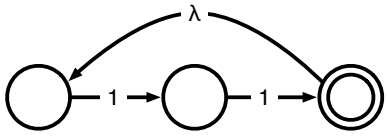


3. We begin by building the NFAs for each of the components of the regular expression. Note that for brevity, we have omitted the incoming arrow for the start state. Unless otherwise specified, the left-most state is the start state.

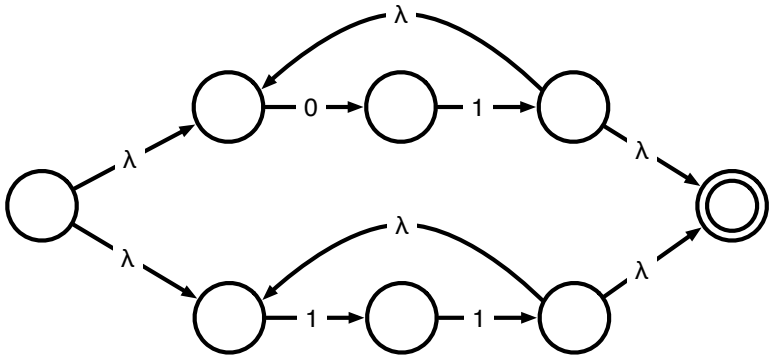
$(01)^+$ =



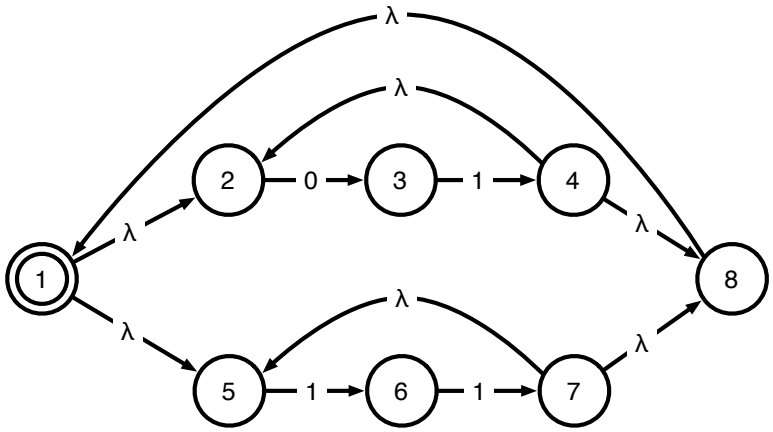
$(11)^+$ =



   We then produce the NFA for $((01)^+|(11)^+)$:



And finally, we produce the NFA for $((01)^+|(11)^+)^*$:

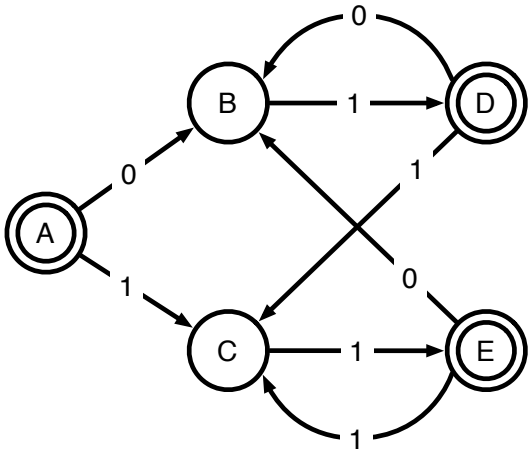


We have numbered the states for reference in the next problem.

4. We can walk through constructing the DFA for this NFA step by step, starting with the start state, which, including λ transitions, is $\{1, 2, 5\}$ (note that this is also a final state!).

State	**0**	**1**	Final?	New name
$\{1, 2, 5\}$	$\{3\}$	$\{6\}$	$\surd$	A
$\{3\}$	error	$\{1, 2, 4, 5, 8\}$		B
$\{4\}$	error	$\{1, 2, 5, 7, 8\}$		C
$\{1, 2, 4, 5, 8\}$	$\{3\}$	$\{6\}$	$\surd$	D
$\{1, 2, 5, 7, 8\}$	$\{3\}$	$\{6\}$	$\surd$	E

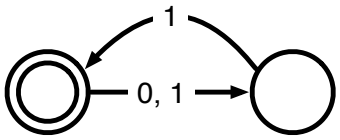
This encodes the following DFA:



5. To reduce the DFA, we begin by placing each of the final states in a single merged state, and each of the non-final states in a single merged state:

States	**0**	**1**
$\{A, D, E\}$	$\{B\}$	$\{C\}$
$\{B, C\}$	error	$\{D, E\}$

We immediately see that from each merged state, no transition takes us to more than one merged state, so we do not need to do any splitting. This table encodes the following minimal DFA



which we can verify accepts the strings in the regular expression $((01)^+|(11)^+)^*$ (which, as we can see from the reduced DFA, is equivalent to $((0|1)1)^*$).

6. Informally, the language cannot be recognized by an FSA. This is because for each (that we see, we must match a later), and we cannot record how many (s we have seen. Formally, we can appeal to the pumping lemma (which you will not have to know!) to show that this language is not regular:

We will show this by contradicition. Let us assume that L is regular. Let w in L be $({}^p g)^p$. By the pumping lemma, we must be able to decompose w into a string xyz such that $|y| \geq 1$, $|xy| \leq p$ and $xy^i z \in L$ for $i \geq 0$. We can see that xy will

contain only left parentheses, and all the right parentheses are in z. So if we repeat y more than once, then xy^iz will not be in L, leading to a contradiction. Hence, the language is not regular.

7. This language *can* be recognized by an FSA. The FSA will have $2 * k + 1$ states, with the first k states recognizing (, the $k + 1$th state recognizing g and the remaining k states recognizing).