MATH 2113 - Assignment 6

Due: Mar 4

1. Let S be the set of states for a finite automaton and let $\sum$ be the set of characters that it acts upon. We define the next state function $N : S \times \sum \to S$.

   a) Under what conditions is N onto?

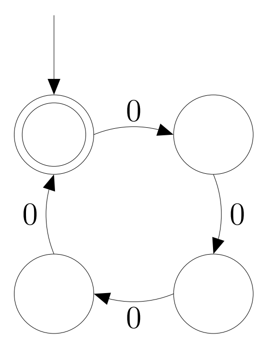
   N will be onto, if in the transition diagram, every state has at least one arrow pointing to it from another state.

   b) Under what conditions is N one to one?

   N will be one to one if in the transition diagram every state has exactly one arrow coming out of it. If it is a finite automaton, this means we can one have one character possible as input.

   c) Provide an example for both a) and b) where $|S| = 4$.

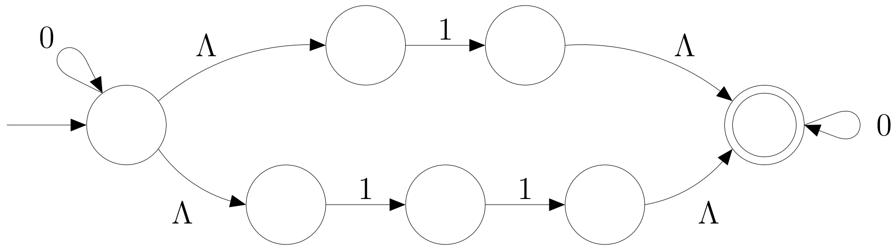
   The following finite automaton accepts strings that have $4k$ consecutive 0s for some $k \geq 0$:



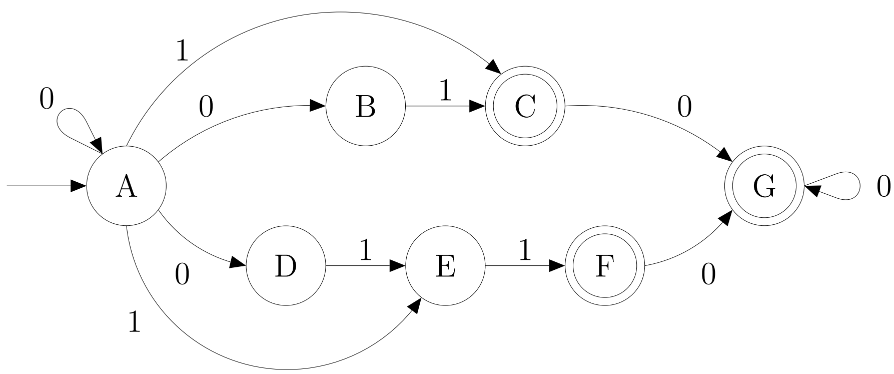
2. Find a finite automaton using the algorithm described in class which accepts the same set of strings generated by the regular expression:

$$0^*(1|11)0^*$$

The first step is to construct an NFA-Λ based on this regular expression:



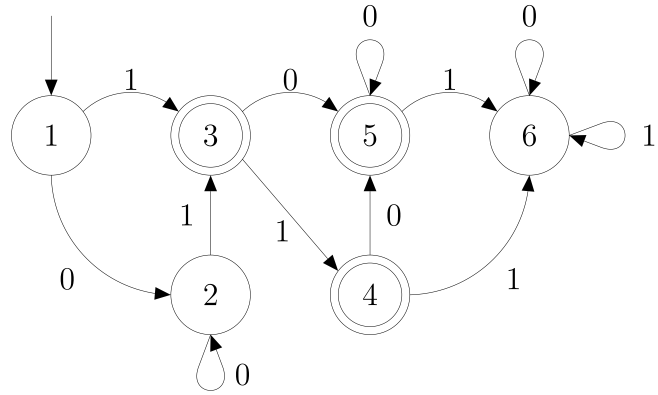
Then, we perform the Λ-replacement step to get:



Now we have an NFA, so we make a table as part of the "subset construction" step:

New Name	State	Accept?	Input 0	Input 1
1	A		ABD	CE
2	ABD		ABD	CE
3	CE	✓	G	F
4	G	✓	G	$\emptyset$
5	F	✓	G	$\emptyset$
6	$\emptyset$		$\emptyset$	$\emptyset$

Therefore, our FA will have 6 states (including the dead state) with the next state function defined by the above table.

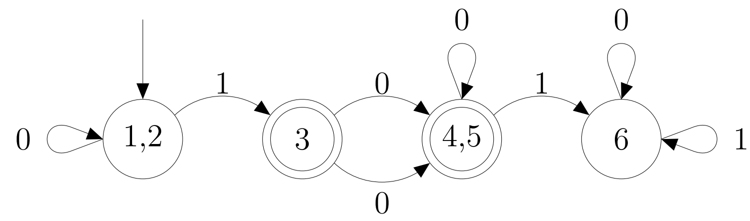


This is sufficient to answer the question, but we can simplify this FA using k-equivalence classes as follows:

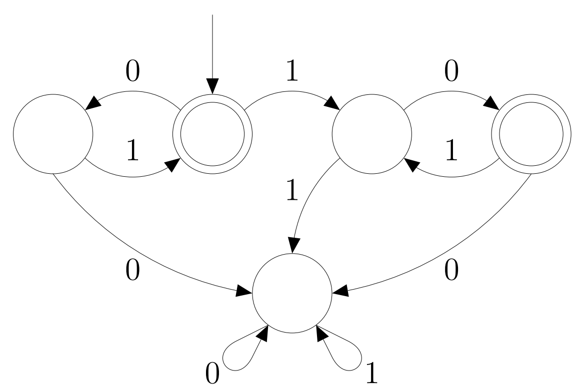
k	k-equivalence classes
0	$\{1,2,6\},\{3,4,5\}$
1	$\{1,2\},\{3\},\{4,5\},\{6\}$
2	$\{1,2\},\{3\},\{4,5\},\{6\}$

The 1-equivalence classes and the 2-equivalence classes are the same. So, we know that the appropriate states are going to be equivalent. We can combine states 1 and 2 as well as states 4 and 5.

We can now draw the simplified finite automaton:

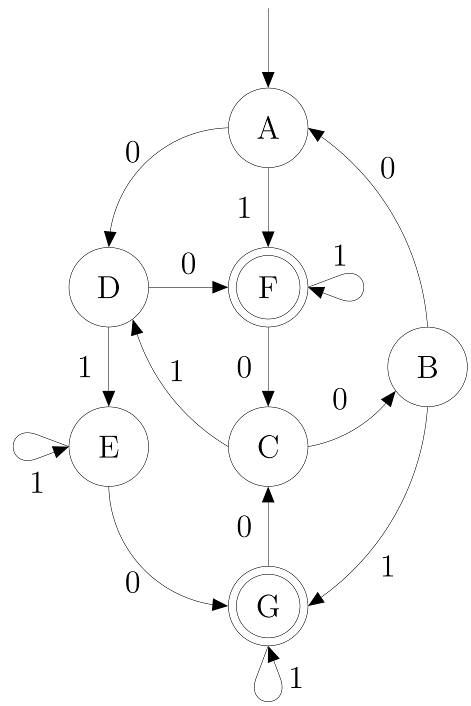


3. Find a regular expression which accepts the same language as the following finite automaton:



We can notice at first, that we must accept the empty string in our regular expression. Also, if we read in $(01)^*$ we will return to the initial accepting state. Therefore, these must also be accepted by the regular expression. Finally, note that from that accepting state, we will also accept $(10)^*$ and end up in the other accepting state. We can easily check that any deviation from what we have already mentioned will result in going to the dead state. Therefore, we conclude that the regular expression represented by this finite automaton is $(01)^*(10)^*$.

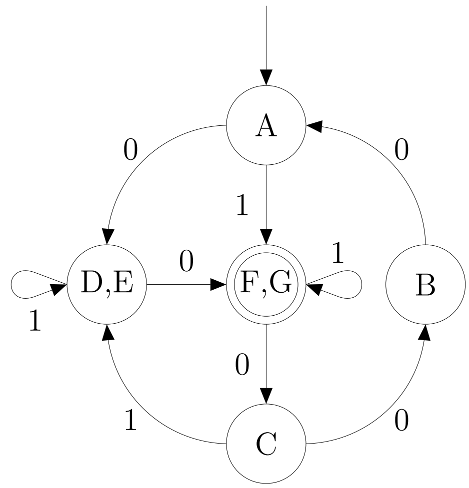
4. Find the finite automaton with the fewest possible states which is equivalent to the following FA:



We start by constructing a table showing the k-equivalence classes:

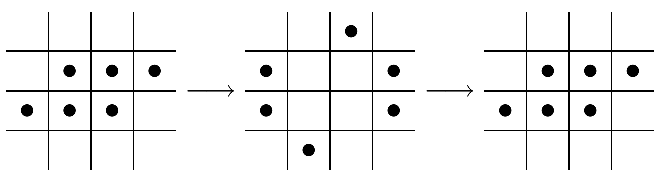
k	k-equivalence classes
0	$\{A, B, C, D, E\}, \{F, G\}$
1	$\{A, B\}, \{C\}, \{D, E\}, \{F, G\}$
2	$\{A\}, \{B\}, \{C\}, \{D, E\}, \{F, G\}$
3	$\{A\}, \{B\}, \{C\}, \{D, E\}, \{F, G\}$

We note that the 2- and 3-equivalence classes are the same, so we can simplify by combining states D, E and states F, G. The resulting finite automaton is:

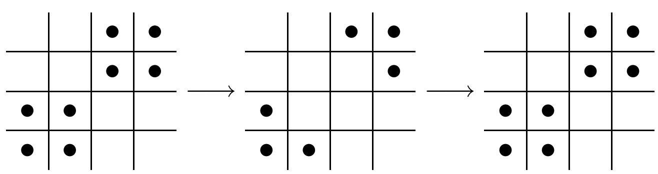


5. Find three "life" configurations which are periodic with $p \geq 2$ different from the ones discussed in class.

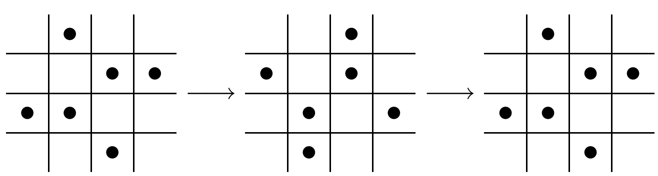
The Toad:



The Beacon:



The Clock:



There are many other configurations that have even longer periods.