Math 3 (V3020) Exercise 10

1. Let $G = (V, T, S, P)$ be the phrase-structure grammar with the vocabulary $V = \{0, 1, A, S\}$, the terminals $T = \{0, 1\}$, and the set of productions P consisting of $S \to 1S$, $S \to 00A$, $A \to 0A$, and $A \to 0$. S is the start symbol.

a) Show that 111000 belongs to the language generated by G.

We can generate the following derivation using the productions P

$$S \Rightarrow 1S \Rightarrow 11S \Rightarrow 111S \Rightarrow 11100A \Rightarrow 111000,$$

i.e. 111000 can be generated by G.

b) Show that 11001 does not belong to the language generated by G.

Every production ends in S, A, or 0. Therefore, a string ending in 1, such as 11001 cannot be generated using these productions.

c) What is the language generated by G?

Starting from the start symbol S, we can use the production $S \to 1S$ to generate a string starting with arbitrary number (also zero) of 1s.

After a string of 1s, $1^n S$ turns into $1^n 00A$ using the production $S \to 00A$.

We can generate arbitrarily many more 0s at the end of the string using $A \to 0A$ consecutively. At some point we use $A \to 0$, generating the final 0 at the end of the string (making the minimum number of 0s to be three).

Thus the language generated by G is
$\mathcal{L}(G) = \{ 1^n 0^m \mid n \geq 0,\ m \geq 3 \}$.

(2.) Construct the phrase-structure grammars to generate the following sets.

a) $\{1^n \mid n \geq 0\}$.

Let us have the start symbol S and from our "target" set we can see that the terminals are just $T = \{1\}$.

Using the produtions $P = \{S \to \lambda, S \to 1S\}$ we can generate an arbitrary number of consecutive 1s (also zero of them since λ is the empty string.)

Here $V = \{1, \lambda, S\}$ is the vocabulary.

b) $\{10^n \mid n \geq 0\}$.

Again, S is the start symbol and the terminals are $T = \{0, 1\}$.

The set of productions
$P = \{ S \to 1A,\ A \to 0A,\ A \to \lambda \}$
suffices to generate a string starting with 1 followed by arbitrarily many 0s.

$V = \{0, 1, \lambda, A, S\}$.

c) $\{ (11)^n \mid n \geq 0 \}$

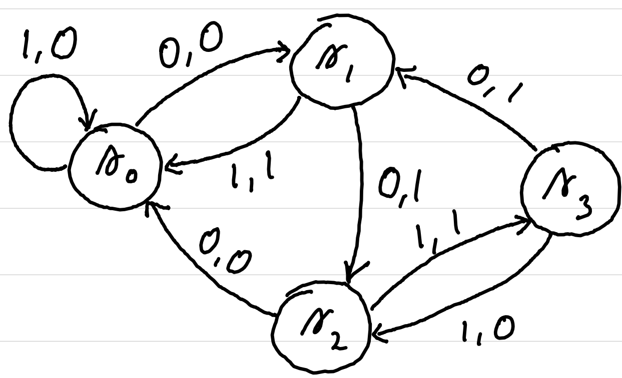
This time $T = \{1\}$ and $P = \{ S \to 11S,\ S \to \lambda \}$ suffices to generate the set.

$V = \{1, \lambda, S\}$.

3. Draw the state diagram for the finite-state machine with the following state table

State	f Input 0	f Input 1	g Output 0	g Output 1
s_0	s_1	s_0	0	0
s_1	s_2	s_0	1	1
s_2	s_0	s_3	0	1
s_3	s_1	s_2	1	0

What is the output generated by the input string 01110 ? s_0 is the initial state.



Now consider the input string 01110 and that we start from the initial state s_0.

Inputing 0 to s_0 takes us to s_1 and outputs 0.

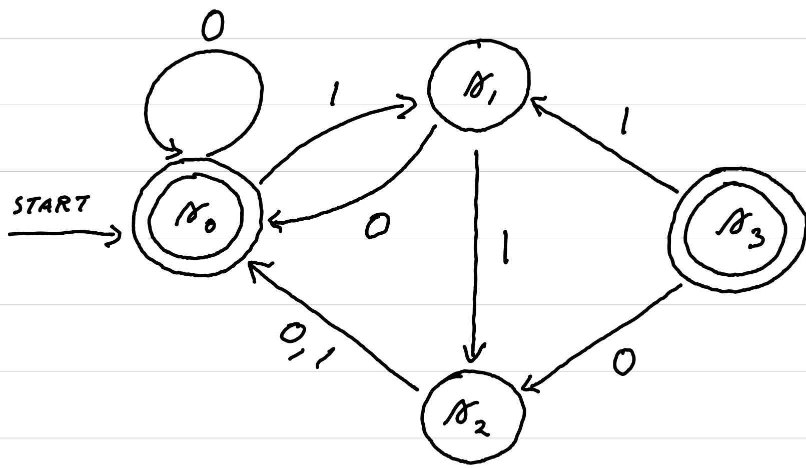
Inputing 1 to s_1 takes us to s_0 and outputs 1.

The next two consecutive inputs 1 and 1 to s_0 take us back to s_0, and output 0 and 0.

Finally, inputing 0 again to s_0 takes us to s_1 and outputs 0.

The full output string is thus 01000.

4. Determine whether each of these strings is recognized by the following finite-state automaton



a) 1101 ?

Starting from the initial state s_0, the input string moves us from state to state in the following manner

$$s_0 \xrightarrow{1} s_1 \xrightarrow{1} s_2 \xrightarrow{0} s_0 \xrightarrow{1} s_1$$

Since s_1 that we arrive to in the end is not a final state, 1101 is not accepted by the automaton.

b) 0101010?

This input moves us from state to state in the following manner

$s_0 \xrightarrow{0} s_0 \xrightarrow{1} s_1 \xrightarrow{0} s_0 \xrightarrow{1} s_1 \xrightarrow{0} s_0 \xrightarrow{1} s_1 \xrightarrow{0} s_0$.

s_0 is a final state, so 0101010 is recognized by the automaton.

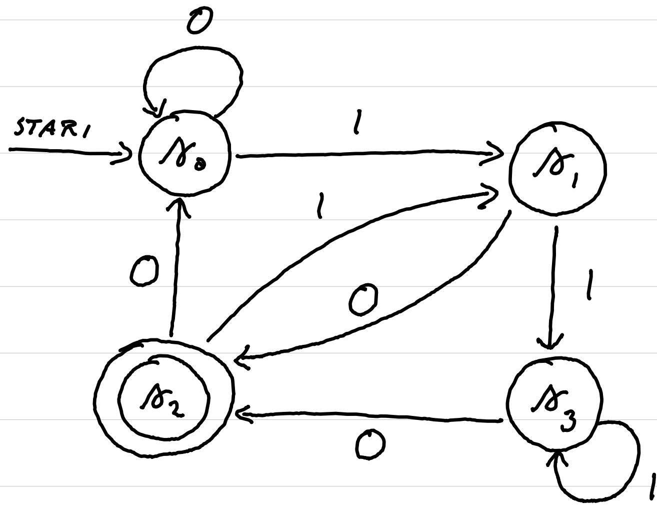
5. Construct a deterministic finite-state automaton that recognizes the set of all bit strings that end with 10.

The important function of this automaton is that it needs to keep track of which were the last two symbols of the input. There are four possibilities, 00, 01, 10, and 11, so we need four states s_0, s_1, s_2, and s_3. We use the correspondence

$s_0 \longleftrightarrow 00$, $s_1 \longleftrightarrow 01$, $s_2 \longleftrightarrow 10$, $s_3 \longleftrightarrow 11$

to store these states, so s_2 is the only final state.

We construct the state diagram in such a way that in order to arrive to a certain state the only possible last two transitions need to correspond to the bit string that the state stores.



Here, for example, to get to the only final state s_2, the state that we are interested in, the last two inputs need to be 10. In other words, the automaton recognizes the input strings ending in 10.

The rest of the diagram is constructed similarly so that the last two unique inputs take you to the other corresponding unique states.