CSCI 338: Assignment 2

Nathan Stouffer

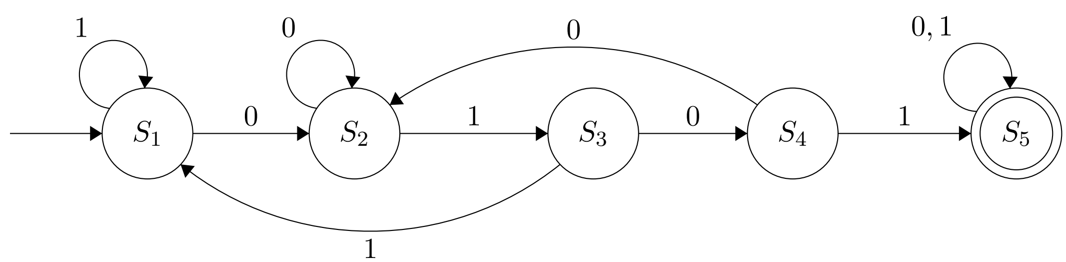
Problem 1

1.1 Give state diagrams of DFAs recognizing the following languages. The alphabet is $\{0, 1\}$.

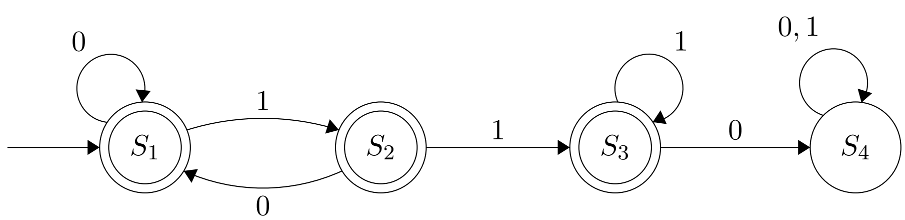
$$A_1 = \{w \mid w \text{ contains the substring } 0101\}$$

$$A_2 = \{w \mid w \text{ does not contain the substring } 110\}$$

Proof: We begin by giving the DFA for A_1:



And now we show the DFA for A_2:



So we have the DFAs for languages A_1 and A_2.

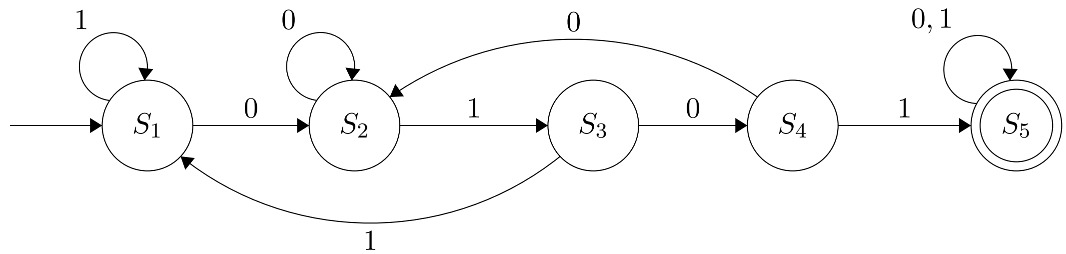
$\square$

1.2 Give state diagrams of the NFAs with the specified number of states recognizing each of the following languages. The alphabet is $\{0, 1\}$.

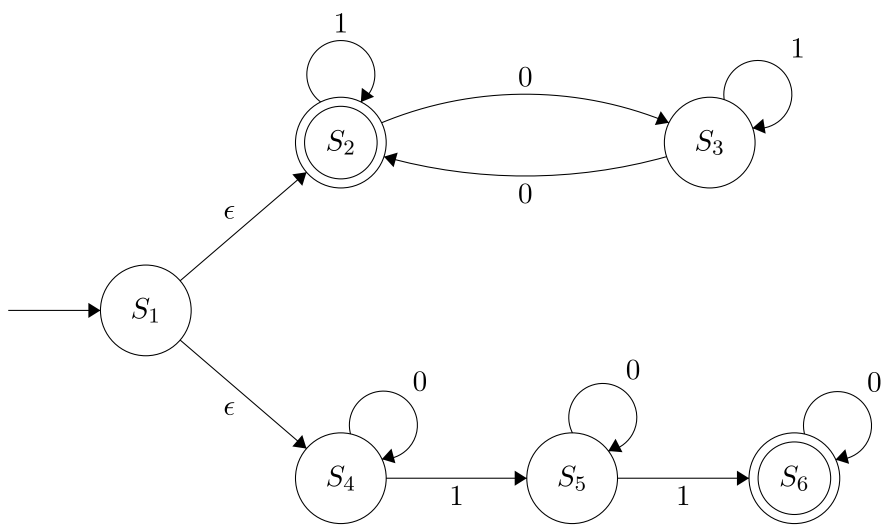
$$B_1 = \{w \mid w \text{ contains the substring } 0101\} \text{ using 5 states}$$

$$B_2 = \{w \mid w \text{ contains an even number of 0s, or exactly two 1s}\} \text{ with 6 states}$$

Proof: For B_1, we use the same DFA as for A_1 (since every DFA is an NFA). It is displayed below.



We now give the NFA for B_2:

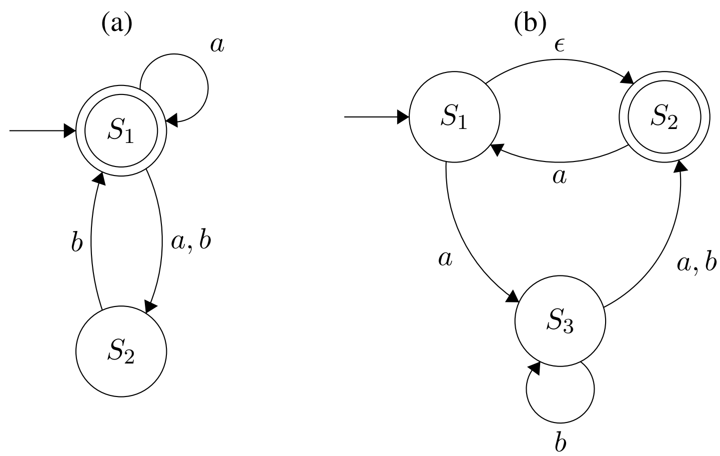


So we have the NFAs for languages B_1 and B_2.

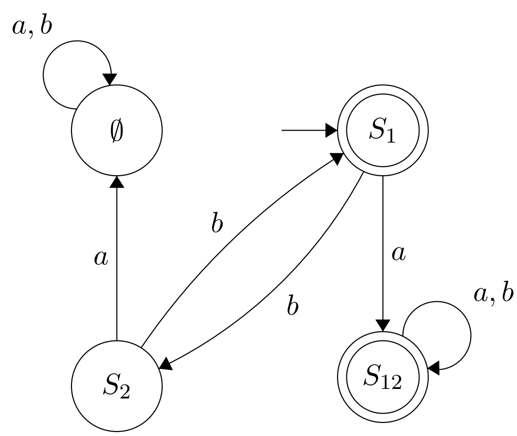
$\square$

Problem 2

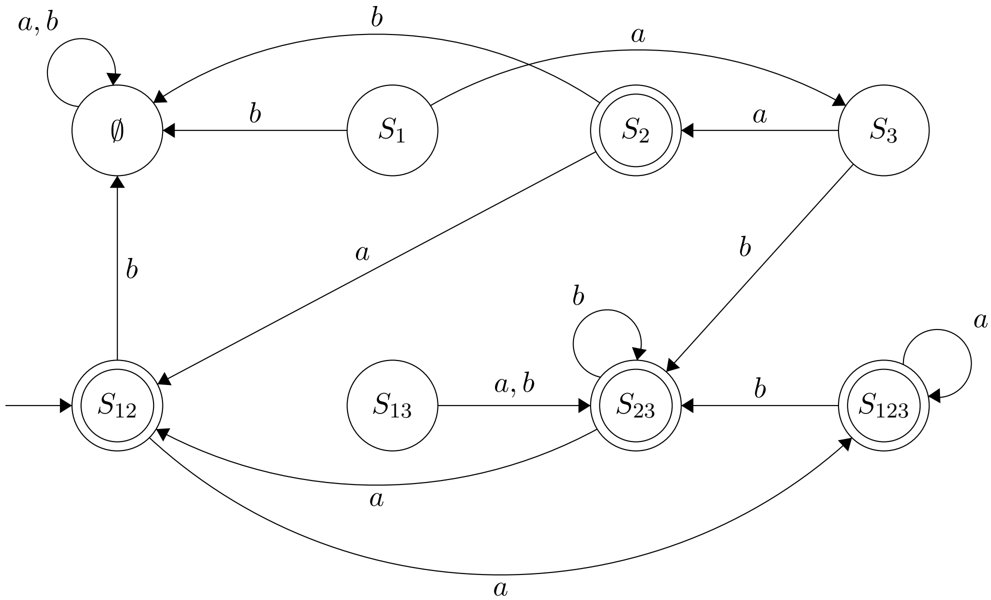
Use the construction given in Theorem 1.39 to convert the following two nondeterministic finite automata to to equivalent deterministic finite automata.



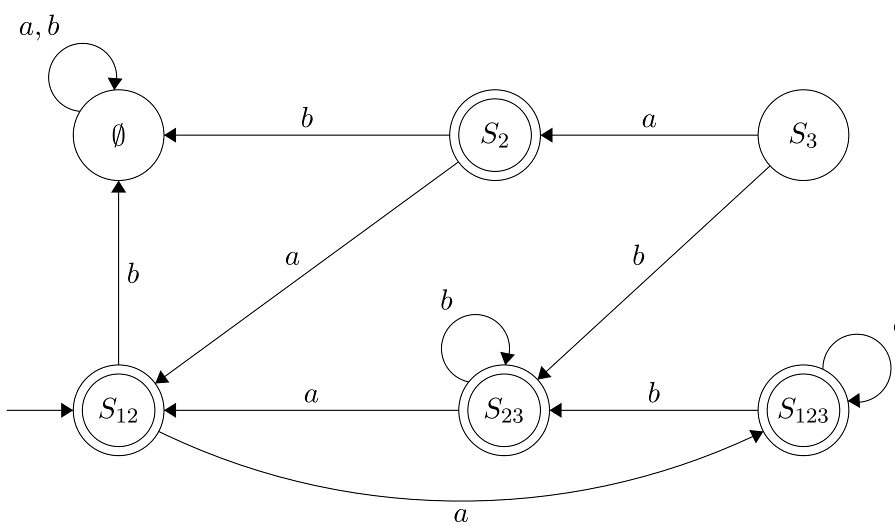
Proof: Using the construction theorem, we convert the NFA in (a) to the following DFA:



The NFA in (b) maps to the following DFA under the construction theorem:



However, it should be noted that the above DFA is inefficient since both states S_1 and S_{13} have no inputs. The following diagram removes them for simplicity.



So we have the equivalent DFAs for each of the given NFAs.

$\square$

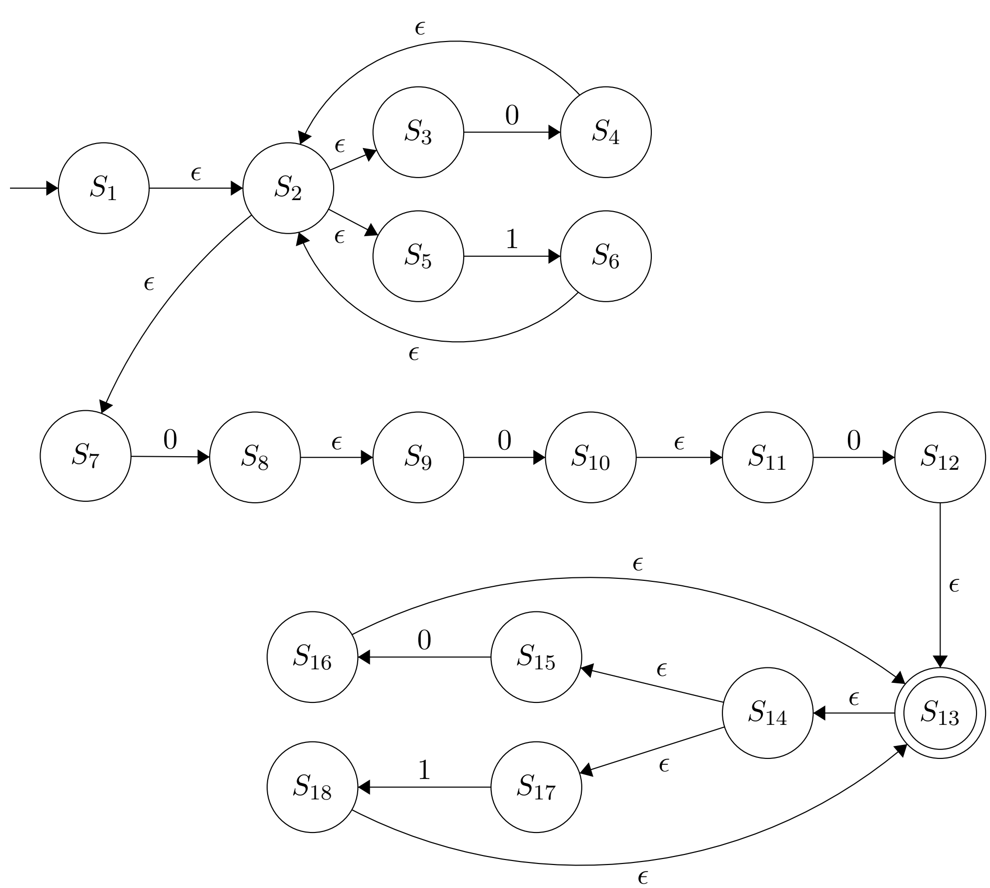
Problem 3

Use the procedure described in Lemma 1.55 to convert the following regular expressions to nondeterministic finite automata.

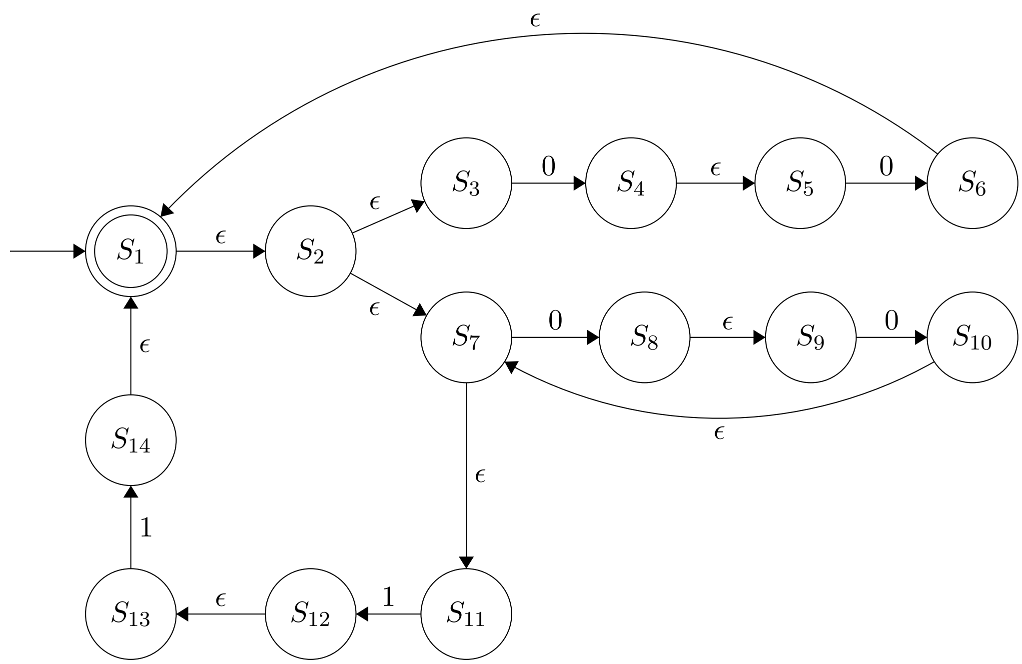
a. $(0 \cup 1)^* 000 (0 \cup 1)^*$

b. $(((00)^*(11)) \cup 01)^*$

Proof: The following NFA diagrams are not at all the simplest ones that could be produced. This is because the it is the result of a proof by construction (which will often produce an inefficient result). We now give the NFA for the regular expression $(0 \cup 1)^* 000 (0 \cup 1)^*$:



And now we give the NFA for $(((00)^*(11)) \cup 01)^*$:

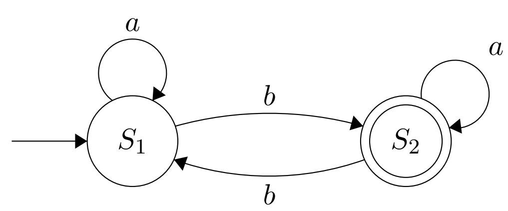


So we have the NFAs equivalent to the given regular expressions.

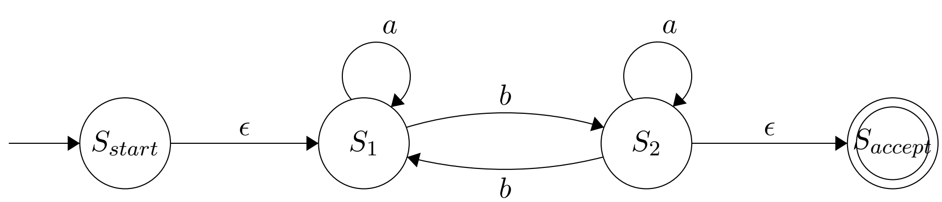
$\square$

Problem 4

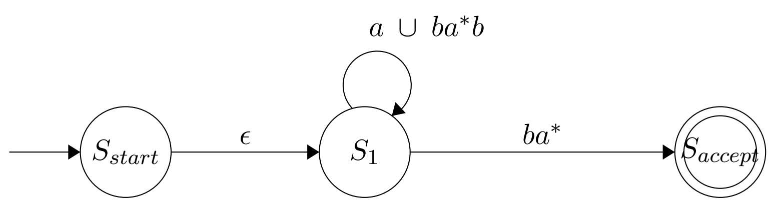
Use the procedure described in Lemma 1.60 to convert the following finite automata to regular expressions.



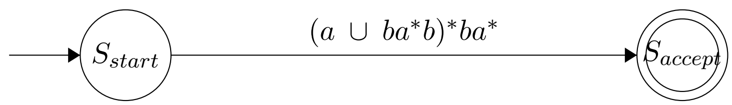
Proof: Using Lemma 1.60, we produce the following GNFA to represent the above DFA.



We then rip S_2 to produce



We then rip S_1 which gives



So our regular language that corresponds to the given NFA is $(a \cup ba^*b)^*ba^*$.

$\square$

Problem 5

Prove that the following languages are not regular.

5.1 $A = \{a^{n^3} \mid n \geq 0\}$ where a^x means a string of x a's.

Proof: We will prove that A is not regular by contradiction. That is, we will assume that A is regular. Since A is regular, we can apply the pumping lemma. Take p to be the pumping length of A. Then, for any string $s \in A$ where $|s| \geq p$ (we know s exists since p is finite and we could construct an element of A to be as large as needed), the following must be true:

1. $xy^iz \in A \ \forall i \geq 0$
2. $|y| > 0$
3. $|xy| \leq p$

Choose $s = a^{p^3} \in A$. We then decompose $s = xyz$. By condition 3, it follows that y can consist of no more than the first p letters of s. For the case of $i = 2$, we can then write that $|s| = |xyz| < |xyyz| = |xyz| + |y|$. Since $|xyz| = p^3$, we can then say that $p^3 < |xyyz|$. It is also true that $|xyyz| = |xyz| + |y| = p^3 + |y| \leq p^3 + p$. We now show some algebra:

$$p^3 + p < p^3 + 3p^2 + 2p + 1 = (p+1)(p^2 + 2p + 1) = (p+1)^3$$

We can now truthfully say that $p^3 < |xyyz| < (p+1)^3$. Since p is an integer, there is no integer in between p and $p + 1$, so $|xyyz|$ cannot be the cube of an integer. Since $|xyyz|$ is not a perfect cube, $xyyz \notin A$.

We have now shown that s cannot be pumped for $i = 2$, which means that the pumping lemma does not apply to A. Since the pumping lemma is not true for A, the language A cannot be regular.

□

5.2 $B = \{0^n 1^m 0^n \mid m, n \geq 0\}$

Proof: We will prove that B is not regular by contradiction. That is, we will assume that B is regular. Since B is regular, we can apply the pumping lemma. Take p to be the pumping length of B. Then, for any string $s \in B$ where $|s| \geq p$ (we know s exists since p is finite and we construct an element of B to be as large as needed), the following must be true:

1. $xy^iz \in B \; \forall i \geq 0$

2. $|y| > 0$

3. $|xy| \leq p$

We now choose $s = 0^p10^p$. Since B is regular, we can decompose $s = xyz$. Because of condition 3, y can consist of no more than the first p letters of s (which are all 0s). Now take $i = 2$. It now must be the case that $xyyz \in B$. Yet, the number of 0s before the 1 (in s) is $p + |y|$ while the number of 0s after the 1 (in s) is only p. By condition 2, $p + |y| > p$ so $xyyz \notin B$. Since s cannot by pumped for $i = 2$, the pumping lemma does not apply to B. Since the pumping lemma does not apply to B, it must be true that B is non-regular.

$\square$