CS 374 HW 2 Problem 1

Aldo Sanjoto, quddus2, Hieu Huynh

TOTAL POINTS

80 / 100

QUESTION 1

1 1A **10 / 10**

✓ **- 0 pts** Correct

- 10 pts Answer is not legible -or- Logic is too hard to follow

- 10 pts Very wrong DFA -or- Wrong DFA and no explanation

- 10 pts Explanation correct but DFA slightly wrong, -or- Explanation slightly wrong but DFA correct

- 5 pts DFA is right, Explanation has a small bug or a typo

QUESTION 2

2 1B **10 / 10**

✓ **- 0 pts** Correct

- 10 pts Answer is not legible -or- Answer is not legible

- 10 pts Very wrong DFA -or- Wrong DFA and no explanation

- 10 pts Explanation correct but DFA slightly wrong, -or- Explanation slightly wrong but DFA correct

- 5 pts DFA is right, Explanation has a small bug or a typo

QUESTION 3

3 1C **20 / 20**

✓ **- 0 pts** Correct

- 20 pts Answer is not legible -or- Logic is hard to follow

- 20 pts Very wrong NFA -or- Wrong NFA and no explanation

- 10 pts Explanation correct but NFA slightly wrong, -or- Explanation slightly wrong but NFA correct

- 5 pts NFA is right, Explanation has a small bug or a typo

QUESTION 4

4 1D **20 / 20**

✓ **- 0 pts** Correct

- 20 pts Answer is not legible -or- Logic is hard to follow

- 20 pts Very wrong NFA -or- Wrong NFA and no explanation

- 10 pts Explanation correct but NFA slightly wrong, -or- Explanation slightly wrong but NFA correct

- 5 pts NFA is right, Explanation has a small bug or a typo

- 5 pts IDK

QUESTION 5

5 1E **0 / 20**

- 0 pts Correct

- 10 pts Explanation correct but NFA slightly wrong, -or- Explanation slightly wrong but NFA correct

- 10 pts More than 22 nodes

- 20 pts Answer is not legible -or- Logic is hard to follow

✓ **- 20 pts Very wrong NFA -or- Wrong NFA and no explanation**

- 5 pts NFA is right, Explanation has a small bug or a typo

- 5 pts IDK

11 is accepted

QUESTION 6

6 1F **20 / 20**

✓ **- 0 pts** Correct

- 20 pts Answer is not legible -or- Logic is hard to follow

- 20 pts Very wrong NFA -or- Wrong NFA and no explanation

- **10 pts** Explanation correct but NFA slightly wrong, -or- Explanation slightly wrong but NFA correct

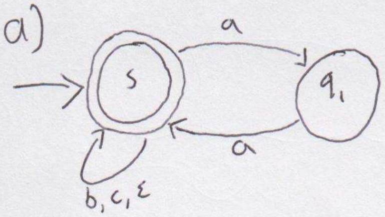
- **5 pts** NFA is right, Explanation has a small bug or a typo

- **10 pts** Almost correct logic, but the number of states is wrong

- **5 pts** IDK

- **0 pts** Click here to replace this description.

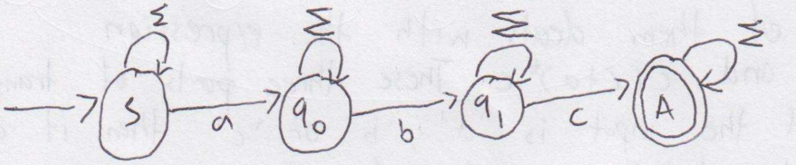
1a)



Explanation:

In our NFA, the starting state S is an accepting state thus it accepts an empty string. The state S has a self-loop of ε, b and c which accept any number occurence of bs and Cs which satifies the expression $(b + c)^*$. The state q1 ensures that every nonempty maximal substring of consecutive a's is of even length that it must have two a's to be accepted otherwise it will dies in state q1. Thus, the expression of all strings over $\{a, b, c\}^*$ in which every nonempty maximal substring of consecutive a's is of even length is satisfied by the NFA.

1b)



Explanation:

In our NFA, the starting state S has a self loop of Σ and has a transition to state q_0 if the input is 'a' which ensures there must be at least a letter 'a' in the string. The state q_0 also has a self loop of Σ and a transition to q_1 if the input letter is 'b' to ensure there's a letter 'b' in the string. In q_1, there's also a self loop of Σ and a transition to the accepting state if the input letter is 'c'. Thus, the overall NFA ensures that there must be a sequence of a, b, c before it enters the accepting state.

1 1A 10 / 10

✓ - 0 pts Correct

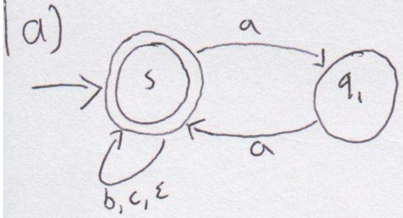
- 10 pts Answer is not legible -or- Logic is too hard to follow

- 10 pts Very wrong DFA -or- Wrong DFA and no explanation

- 10 pts Explanation correct but DFA slightly wrong, -or- Explanation slightly wrong but DFA correct

- 5 pts DFA is right, Explanation has a small bug or a typo

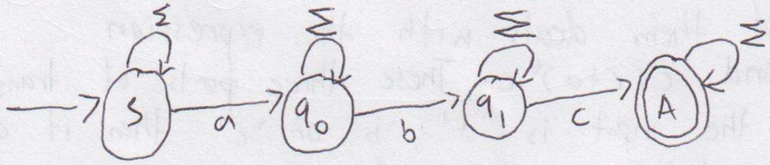
1a)



Explanation:

In our NFA, the starting state S is an accepting state thus it accepts an empty string. The state S has a self-loop of ε, b and c which accept any number occurence of bs and Cs which satifies the expression $(b + c)^*$. The state q1 ensures that every nonempty maximal substring of consecutive a's is of even length that it must have two a's to be accepted otherwise it will dies in state q1. Thus, the expression of all strings over $\{a, b, c\}^*$ in which every nonempty maximal substring of consecutive a's is of even length is satisfied by the NFA.

1b)



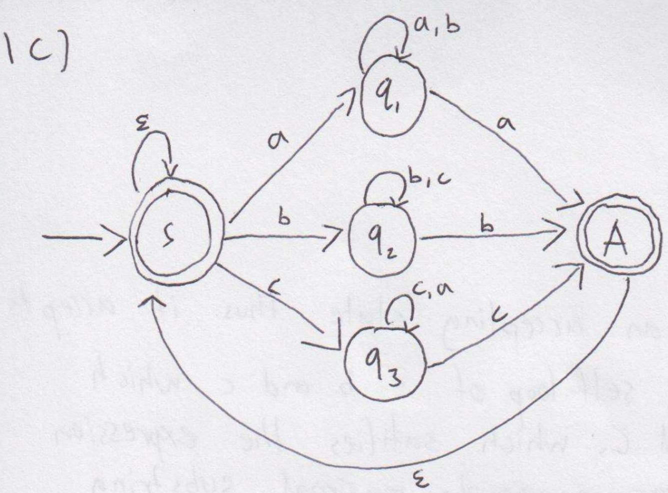
Explanation:

In our NFA, the starting state S has a self loop of Σ and has a transition to state q_0 if the input is 'a' which ensures there must be at least a letter 'a' in the string. The state q_0 also has a self loop of Σ and a transition to q_1 if the input letter is 'b' to ensure there's a letter 'b' in the string. In q_1, there's also a self loop of Σ and a transition to the accepting state if the input letter is 'c'. Thus, the overall NFA ensures that there must be a sequence of a, b, c before it enters the accepting state.

2 1B 10 / 10

- ✓ - 0 pts Correct
- - 10 pts Answer is not legible -or- Answer is not legible
- - 10 pts Very wrong DFA -or- Wrong DFA and no explanation
- - 10 pts Explanation correct but DFA slightly wrong, -or- Explanation slightly wrong but DFA correct
- - 5 pts DFA is right, Explanation has a small bug or a typo

1c)



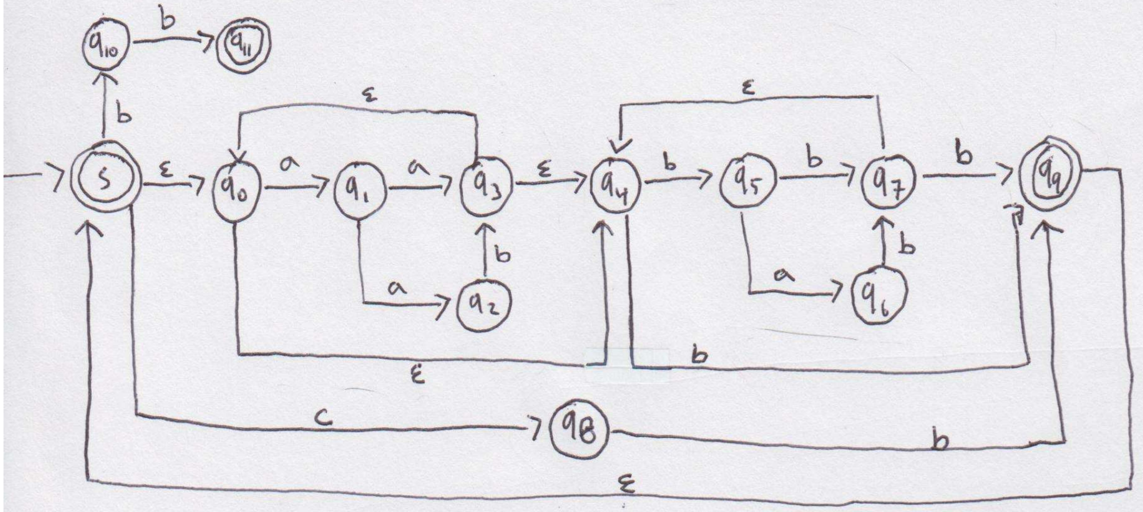
Explanation:

In our NFA, we set the starting set to be an accepting state since the regex accepts empty string. Then, there are three transitions from the starting state where each of them deals with the expression $a(a+b)^* a$, $b(b+c)^* b$, and $c(c+a)^* c$ These three parts of transition have the same idea that if the input is 'a', 'b' or 'c' then it also must be ended with 'a', 'b' and 'c' respectively before it gets to the Accepting state. At the accepting state, there's a transition of empty string to the starting state to repeat the whole process if needed.

3 1C 20 / 20

- ✓ - 0 pts Correct
- - 20 pts Answer is not legible -or- Logic is hard to follow
- - 20 pts Very wrong NFA -or- Wrong NFA and no explanation
- - 10 pts Explanation correct but NFA slightly wrong, -or- Explanation slightly wrong but NFA correct
- - 5 pts NFA is right, Explanation has a small bug or a typo

10) $(((aa + aab)^* (bab + bb)^* + c) b)^* + bb$



Explanation:

The state q_0, q_1, q_2 and q_3 satisfy the expression $(aa + aab)^*$ where if the input is aa it follows the path of q_0 to q_1 to q_3 and q_0 to q_1 to q_2 to q_3 if the input is aab. It can repeat by using the epsilon transition in q_3 to q_0. If there's no such string, it can use the epsilon transition to stute q_4.

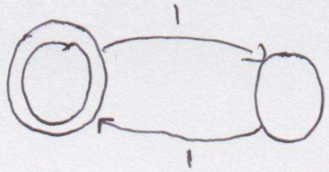
The state q_4, q_5, q_6 and q_7 satisfy the expression $(bab + bb)^*$ where if the input string is bb, it follows the path from q_4 to q_5 to q_7. And if the input string is bab, it follows the path from q_4 to q_5 to q_6 to q_7. It can repeat by using the epsilon transition in q_7 to q_4.
If there's no such input it can use the transition of b from q_4 to q_9.

If the input is C, the NFA provides a path from the starting state to q_8 and then a transition of input b to state q_9.
At q_9 the NFA will accept the state which satisfy the expression $((aa + aab)^* (bab + bb)^* + c) b)$
If the input has an input bb, it can use the epsilon transition to the starting state to q_{10} to q_{11} and will be accepted.

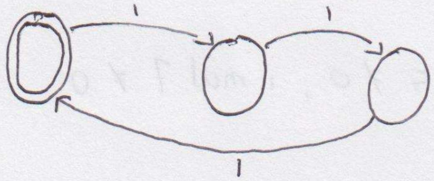
4 1D 20 / 20

- ✓ - 0 pts Correct
- - 20 pts Answer is not legible -or- Logic is hard to follow
- - 20 pts Very wrong NFA -or- Wrong NFA and no explanation
- - 10 pts Explanation correct but NFA slightly wrong, -or- Explanation slightly wrong but NFA correct
- - 5 pts NFA is right, Explanation has a small bug or a typo
- - 5 pts IDK

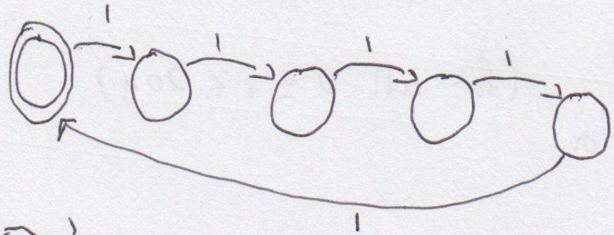
1E)



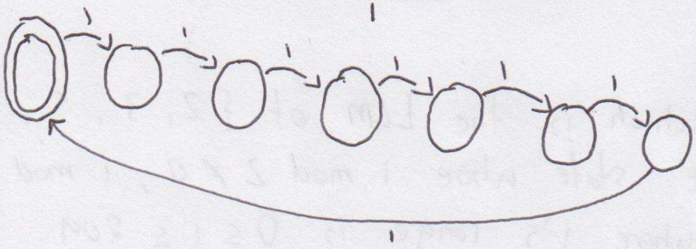
NFA: N_2 which accept $L_2 = \{ w \in 1^* \mid |w| \% 2 = 0 \}$



NFA: N_3 which accept $L_3 = \{ w \in 1^* \mid |w| \% 3 = 0 \}$

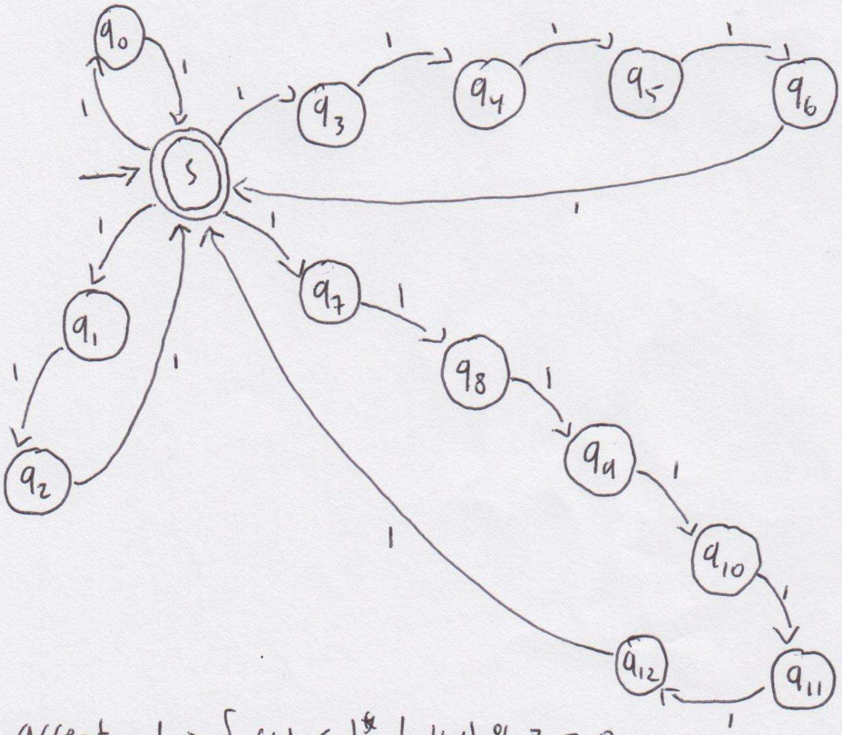


NFA: N_5 which accept $L_5 = \{ w \in 1^* \mid |w| \% 5 = 0 \}$



NFA: N_7 which accept $L_7 = \{ w \in 1^* \mid |w| \% 7 = 0 \}$

$L = L_2 \cup L_3 \cup L_5 \cup L_7$, Thus our NFA becomes:



which accept $L = \{ w \in 1^* \mid |w| \% 2 = 0, |w| \% 3 = 0, |w| \% 5 = 0, |w| \% 7 = 0 \}$ //

5 1E 0 / 20

- **- 0 pts** Correct
- **- 10 pts** Explanation correct but NFA slightly wrong, -or- Explanation slightly wrong but NFA correct
- **- 10 pts** More than 22 nodes
- **- 20 pts** Answer is not legible -or- Logic is hard to follow
- ✓ **- 20 pts** **Very wrong NFA -or- Wrong NFA and no explanation**
- **- 5 pts** NFA is right, Explanation has a small bug or a typo
- **- 5 pts** IDK

11 is accepted

1F) $\Sigma = \{ a \}$

$Q = \{ q_i \mid 0 \leq i \leq 209 \}$

$S = q_0$

$A = \{ q_i \mid i \bmod 2 \neq 0,\ i \bmod 3 \neq 0,\ i \bmod 5 \neq 0,\ i \bmod 7 \neq 0$ and $0 \leq i \leq 209 \}$

$$\delta(q_i, r) = \begin{cases} q_i & \text{if } r = \varepsilon \\ q_{(i+1) \% 210} & \text{if } r = a \end{cases} \qquad (\text{for all } 0 \leq i \leq 209)$$

Explanation:

Our NFA consists of 210 states which is the LCM of $\{2, 3, 5, 7\}$. The accepting states q_i is the state where $i \bmod 2 \neq 0$, $i \bmod 3 \neq 0$, $i \bmod 5 \neq 0$ and $i \bmod 7 \neq 0$ where i's range is $0 \leq i \leq 209$. Therefore, our NFA will only accept string of a's that length is NOT divisible by 2, 3, 5 and 7.

6 1F 20 / 20

✓ - **0 pts** Correct

- **20 pts** Answer is not legible -or- Logic is hard to follow

- **20 pts** Very wrong NFA -or- Wrong NFA and no explanation

- **10 pts** Explanation correct but NFA slightly wrong, -or- Explanation slightly wrong but NFA correct

- **5 pts** NFA is right, Explanation has a small bug or a typo

- **10 pts** Almost correct logic, but the number of states is wrong

- **5 pts** IDK

- **0 pts** Click here to replace this description.