Quiz 1 Solutions

Automata Theory Monsoon 2023, IIIT Hyderabad

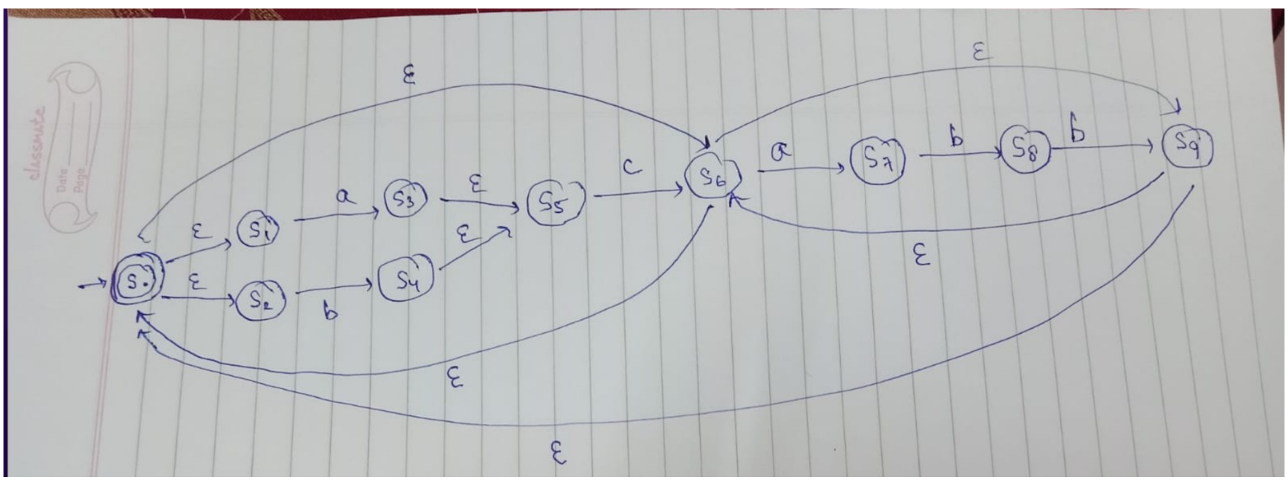
September 7, 2023

Please read the solution and marking scheme carefully, asking for increase in marks in spite of the marking scheme's guidelines may lead to reduced marks.

Solutions

1. [2 points] Rubrik :

- **2 marks** for any NFA which is correct
- **1 mark** if minor mistake in transitions.
- **0 mark** if NFA is entirely wrong.



2. [3 points] **Solution :** We notice that the language can be written down as

$$
\begin{aligned}
L &= \{\ w \mid w \equiv 1 \bmod 2 \text{ and } w \not\equiv 0 \bmod 3\ \} \\
&= \{\ w \mid w \equiv 1 \bmod 2 \text{ and } (w \equiv 1 \bmod 3 \ \text{ or } w \equiv 2 \bmod 3\ )\} \\
&= \{\ w \mid (w \equiv 1 \bmod 2 \text{ and } w \equiv 1 \bmod 3) \text{ or } (w \equiv 1 \bmod 2 \text{ and } w \equiv 2 \bmod 3)\} \\
&= \{\ w \mid (w \equiv 1 \bmod 2 \text{ and } w \equiv 1 \bmod 3) \text{ or } (w \equiv -1 \bmod 2 \text{ and } w \equiv -1 \bmod 3)\} \\
&= \{\ w \mid w \equiv 1 \bmod 6 \text{ or } w \equiv -1 \bmod 6\ \} \\
&= \{\ w \mid w \equiv 1 \bmod 6 \text{ or } w \equiv 5 \bmod 6\ \}
\end{aligned}
$$

Now, we can construct the DFAs for both $w \equiv 1 \bmod 6$ and $w \equiv 5 \bmod 6$. All that needs to be done is to construct their union by constructing an NFA whose start states connect with the start states of both the DFAs through an ϵ-transition.

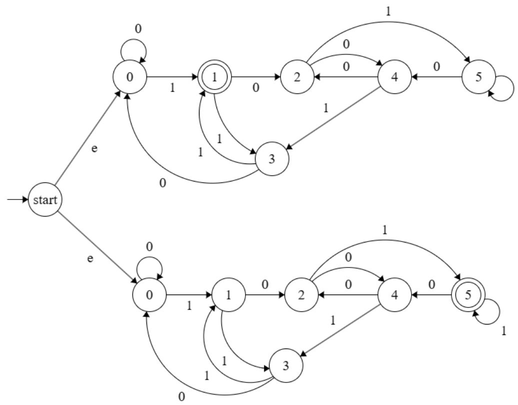


Figure 1: Final NFA

Other Solutions, such as a DFA for divisibility by 6 with the states representing remainders, and any other correct solution will be accepted.

Rubrik : Can either chose to do the above solution, in which case the rubrik to be followed is

- **1 mark** for explanation
- **2 marks** for constructing the DFA/NFA

Any silly mistakes are up to the discretion of the TA (cut maybe 0.5 marks for small errors, and if the DFAs are entirely wrong then give 0-0.5 depending on their explanation)

3. [2 points] **Solution:** We can write this language as the union of three languages. That is,

$$\begin{aligned} L' &= \{a^n b^m \mid n \neq m\} \cup \{(a \cup b)^* b (a \cup b)^* a (a \cup b)^*\} \\ &= \{a^i b^j \mid i > j\} \cup \{a^i b^j \mid i < j\} \cup \{(a \cup b)^* b (a \cup b)^* a (a \cup b)^*\} \\ &= L_1 \cup L_2 \cup L_3. \end{aligned}$$

CFG for $L_1 = \{a^i b^j \mid i > j\},\ G_1$:
$S_1 \rightarrow aS_1b|aS_1|a$

CFG for $L_2 = \{a^i b^j \mid i < j\},\ G_2$:

$S_2 \rightarrow S_2b|b$
$B \rightarrow bB|b$

CFG for $L_3 = \{(a \cup b)^* b (a \cup b)^* a (a \cup b)^*\},\ G_3$:

$S_3 \rightarrow XbXaX$
$X \rightarrow aX|bX|\epsilon$

Hence, CFG for $L' = L_1 \cup L_2 \cup L_3,\ G$:

$S \rightarrow S_1|S_2|S_3$
$S_1 \rightarrow aS_1b|aS_1|a$
$S_2 \rightarrow S_2b|b$
$B \rightarrow bB|b$
$S_3 \rightarrow XbXaX$
$X \rightarrow aX|bX|\epsilon$

Rubrik: Up to +0.5 marks for the right idea and simplification of the language. +1 for either sub-grammars.

4. [4 points] Part (a) (Total: 2.5 points)
Yes, the pumping lemma for regular languages holds for F (1 point)
No / violates / contradicts / don't answer with yes or no (0 points)

State **all** 3 properties of the pumping lemma:
1. $|xy| \leq p$
2. $|y| > 0$
3. $xy^iz \in F\ \forall i \geq 0$
(0.5 points)

If someone does not answer with yes as listed above, the maximum attainable marks in part (a) is 0.5 points **if** they state all 3 properties as specified above.

Perfect case-wise elaboration (1 point)
Else: If they answer yes, based on **if** they cover any or some valid cases (0.5 points)

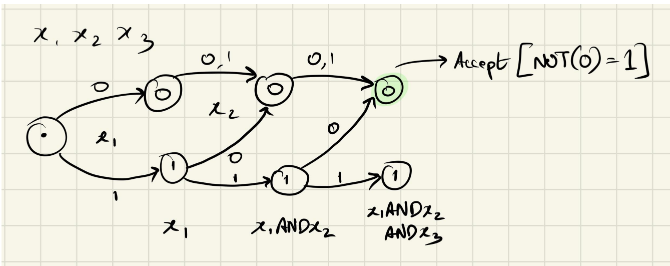
Take p $\geq$ 2. Consider any string $0^i1^j2^k$ in the language. If i = 0, take x = ϵ ; y = 1 if j > 0 else y = 2. Since strings of the form 1^j2^k are always in the language, we satisfy the conditions of the pumping lemma.
If i = 1, then j = k. Take x = ϵ ; y = 0. Adding any number of 0's preserves membership in the language.
If i = 2, take x = ϵ ; y = 00. On pumping the number of 0's must be an even number, and the resulting string is still in the language.
If i > 2, follow the approach similar to the case i = 1.

Part (b) (Total: 1.5 points)
It is a context free language (0.5 points)
Regular / Not answered / Not regular (0 points)
Justification: Memory / stack / Perfect PDA or CFG for the language / remember number of 1s and 2s (1 point)
Else if: Partially correct PDA or CFG (0.5 points)

Else: because it violates pumping lemma / Using DFA or NFA (0 points)

5. [3 points] You are given the following finite circuit. It has 3 boolean values as input and one output value. Convert the circuit into a DFA that takes in a size-3 binary string as input that corresponds to $x_1x_2x_3$ and accepts it if the circuit produces 1 as it's output. What can you say about the power of finite sized circuits and regular languages?

   Answer:

   Any finite sized circuit can be converted to a DFA, the power of a single finite sized circuit is same as a DFA, hence regular.

   Rubrik: The DFA could vary, verify if the only string that can not reach the accept state is 111.

   Note that the question asks for a DFA, Provide full only if the DFA is fully correct. One partial for partially correct DFA, NFA gets no marks. Cut one mark for invalid DFA.

   One mark for explaining the power of finite circuit is same as regular language. No explaination required.

6. [3 points] Write down the Context Free Grammar and the corresponding PDA for the language $L = \{0^n 1^{3n} | n \geq 1\}$.
   [CO-1, CO-2, CO-3]

   Solution: The grammar G generating L has the following rules:

$$S \rightarrow 0A111$$
$$A \rightarrow 0A111|\epsilon$$

   The PDA would have the following transition functions:

$$\delta(q_0, 0, \epsilon) = (q_0, XXX)$$
$$\delta(q_0, 1, X) = (q_1, \epsilon)$$
$$\delta(q_1, 1, X) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, \$) = (q_2, \$).$$

   Rubrik: +1.5 for writing the correct grammar, +1.5 for the correct PDA.