

Homework 3 - Solutions

CPSC 326

Dang Phung

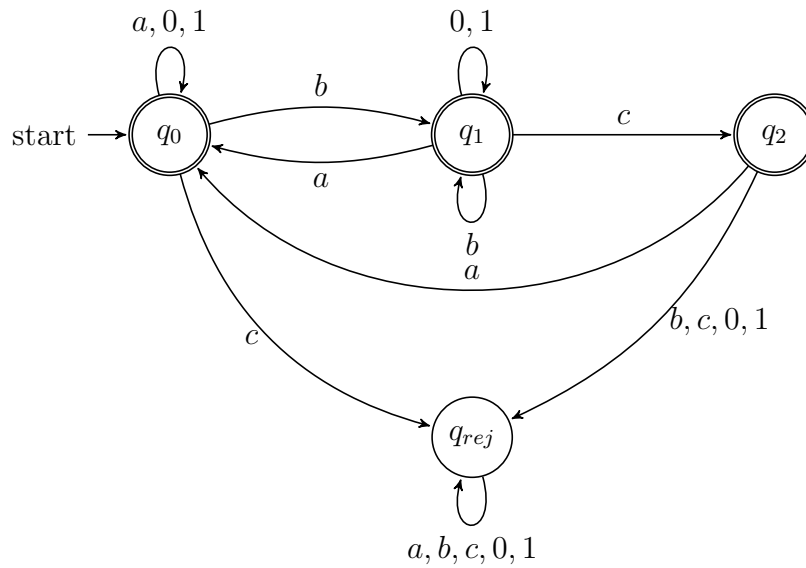
October 26, 2025

Question 1 (20 points)

For each of the following languages, state whether or not the language is regular. Make sure to prove your answer.

1. $\{w \in \{a, b, c, 0, 1\}^* \mid \text{every } c \text{ in } w \text{ must appear in the subword } bca\}$

Answer: This language is regular.



2. $\{w \in \Sigma_{bool}^* \mid w = 0^n 1^n, n \geq 5\}$

Answer: This language is not regular.

Proof by pumping lemma:

Assume for contradiction that the language is regular. Let p be the pumping length.

Consider the string $w = 0^{p+5}1^{p+5} \in L$ (since $p + 5 \geq 5$).

We have $|w| = 2(p + 5) \geq p$, so by the pumping lemma, we can write $w = w_1 w_2 w_3$ where:

1. $|w_1 w_2| \leq p$

2. $|w_2| \geq 1$

3. $w_1 w_2^i w_3 \in L$ for all $i \geq 0$

Since $|w_1 w_2| \leq p < p + 5$ and $w = 0^{p+5} 1^{p+5}$, the substring $w_1 w_2$ is entirely 0's. Therefore, $w_2 = 0^k$ for some $k \geq 1$.

Now consider $w_1 w_2^0 w_3 = w_1 w_3 = 0^{p+5-k} 1^{p+5}$.

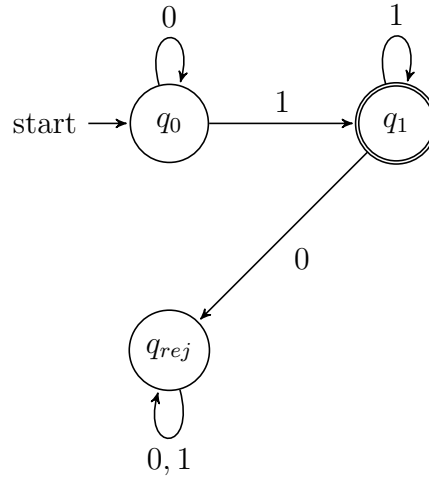
Since $k \geq 1$, we have $p + 5 - k < p + 5$, so the number of 0's is less than the number of 1's.

Therefore, $w_1 w_3 \notin L$, which contradicts the pumping lemma.

Therefore, the language is not regular.

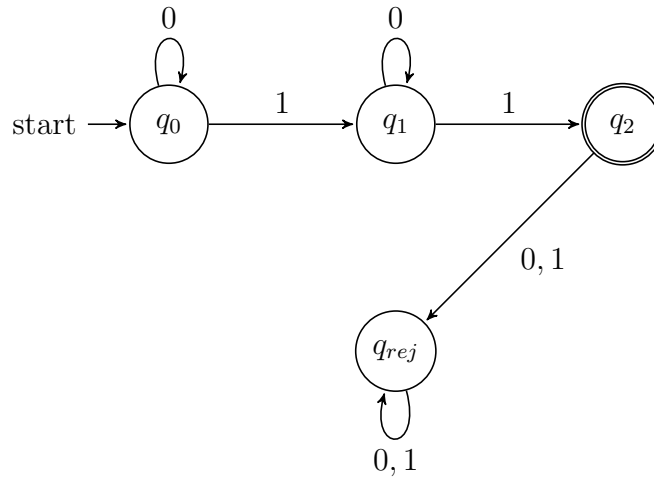
3. $\{w \in \Sigma_{bool}^* \mid w = 0^n 1^m, n \geq 0, m \geq 1\}$

Answer: This language is regular.



4. $\{w \in \Sigma_{bool}^* \mid w = w_1 w_2, w_1 \in 0^* 1, w_2 \in 0^* 1, \text{ and } |w_1|, |w_2| \geq 1\}$

Answer: This language is regular.

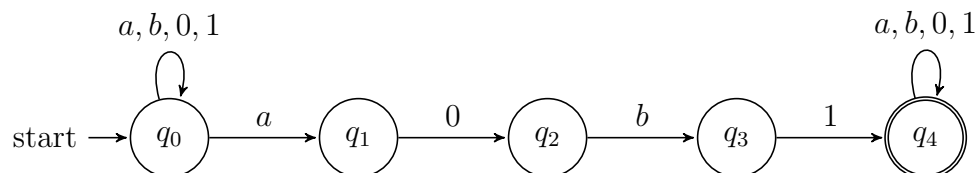


Question 2 (10 points)

Design a NFA for the following language:

$$L_2 = \{w \in \{a, b, 0, 1\}^* \mid w = w_1 a 0 b 1 w_2, w_1, w_2 \in \{a, b, 0, 1\}^*\}$$

Solution: The language consists of all strings that contain the substring $a0b1$.

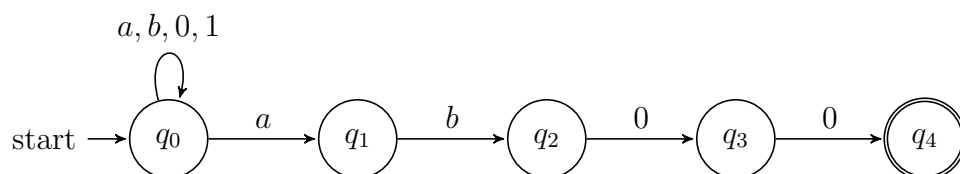


Question 3 (10 points)

Design a NFA for the following language:

$$L_3 = \{w \in \{a, b, 0, 1\}^* \mid w \text{ ends in the string } ab00\}$$

Solution:

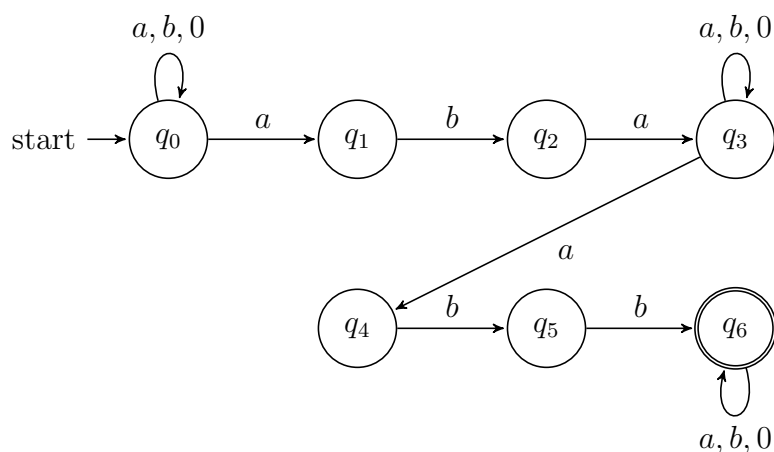


Question 4 (10 points)

Design a NFA for the following language:

$$L_4 = \{w \in \{a, b, 0\}^* \mid w \text{ contains the subwords } aba \text{ and } abb\}$$

Solution: The NFA needs to find both substrings aba and abb in the input.



Question 5 (20 points)

Please answer the following sub-questions:

1. Is the set of languages accepted by NFAs equal to, larger, or smaller than the set of languages accepted by DFAs?

Answer: The set of languages accepted by NFAs is **equal to** the set of languages accepted by DFAs.

Explanation: NFAs and DFAs are the same in terms of power. For every NFA, there exists a DFA that accepts the same language, and every DFA is an NFA. Therefore, they accept the same class of languages (regular languages).

2. Regular languages are those languages accepted by NFAs, true or false?

Answer: True.

Explanation: Regular languages are defined as those languages that can be accepted by finite automata (either DFAs or NFAs). Since NFAs and DFAs are the same in terms of power, regular languages are accepted by both.

3. When converting a NFA to a DFA the resulting DFA will always have the same number of states as the NFA, true or false?

Answer: False.

Explanation: When converting an NFA with n states to a DFA using the subset construction algorithm, the resulting DFA can have up to 2^n states. The DFA may have the same number of states in special cases, but in general it can have a lot more states.

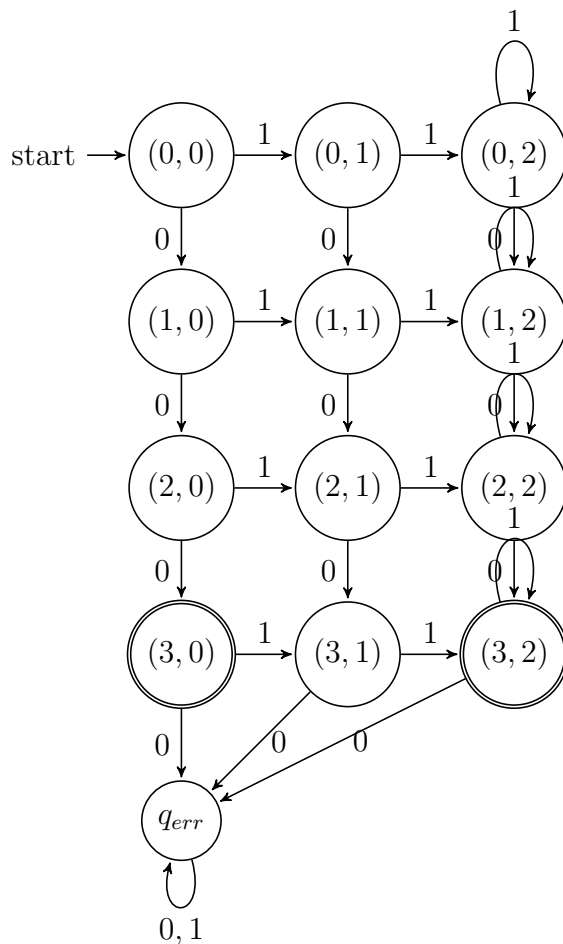
4. The pumping lemma doesn't prove anything about a language being accepted by a NFA. That is, it's only useful for showing a language can't be accepted by a DFA. True or False?

Answer: False.

Explanation: Since NFAs and DFAs accept regular languages, the pumping lemma applies to both equally. If a language fails the pumping lemma, it cannot be accepted by either a DFA or an NFA. The pumping lemma is a property of regular languages, and is not tied to one or another.

Question 6 (10 points)

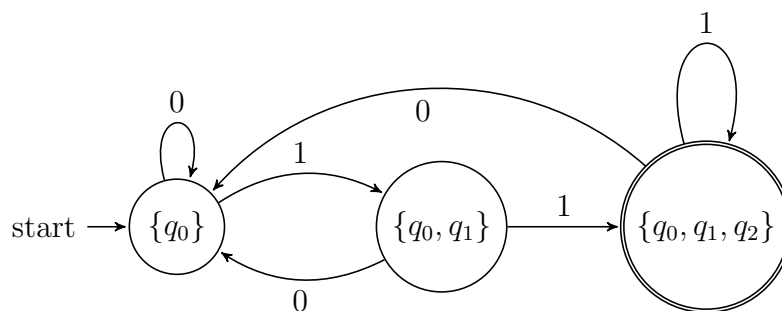
Let $L_6 = \{w \in \Sigma_{bool}^* \mid |w|_0 = 3 \text{ and } (|w|_1 \geq 2 \text{ or } |w|_1 = 0)\}$. Develop a Finite Automaton that recognizes L_6 .



Question 7 (10 points)

Convert the following NFA to a DFA:

Solution DFA:



Question 8 (10 points)

In-class we developed an algorithm that converts any NFA into a DFA. Why is there no algorithm to convert any DFA into an NFA?

Answer:

There is no need for such an algorithm because **every DFA is already an NFA**.

A DFA is a special case of an NFA where there are no ϵ transitions and for each state and each input symbol, there is exactly one transition. Since the definition of an NFA is more general than a DFA, any DFA could technically be an NFA.