Problem Set 4 - CPSC 326
Solutions

Fernando, Dang, Raj, Eric M.

October 23, 2025

Problem 1

Consider the following Language:

$L_1 = \{w \in \{a, b, c, d\}^* \mid (w \text{ contains the strings } abb \text{ and } bbc) \text{ or } (w \text{ contains the string } abc)\}$

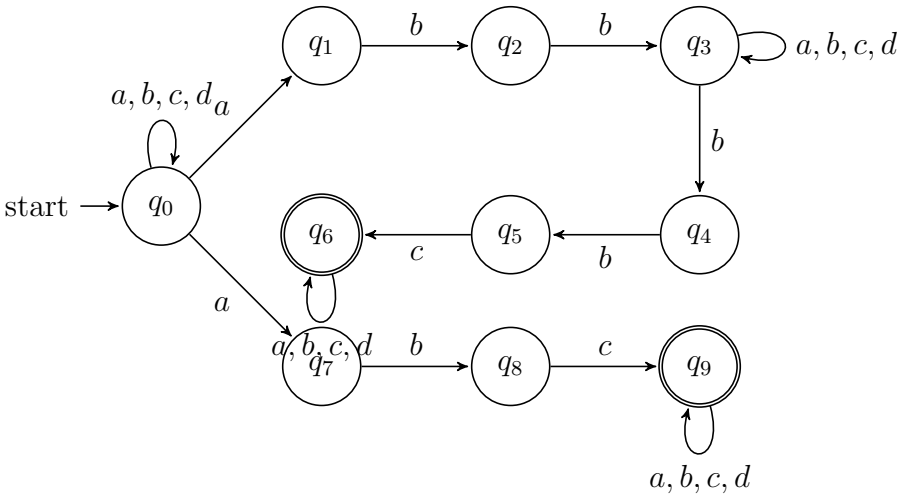
Develop a NFA for L_1.
Answer:
The NFA uses nondeterminism to check for either:

- Case 1: Both abb and bbc appear in w
- Case 2: abc appears in w

States:

- q_0: Initial state
- q_1, q_2, q_3: Track progress toward finding abb
- q_4, q_5, q_6: Track progress toward finding bbc (after finding abb)
- q_7: Found both abb and bbc (accepting)
- q_8, q_9, q_{10}: Track progress toward finding abc
- q_{11}: Found abc (accepting)

NFA Diagram:



Explanation: The NFA nondeterministically chooses between:

- Following the upper path to find *abb*, then continuing to find *bbc*
- Following the lower path to find *abc*

Once either accepting state is reached, the string is accepted.

Problem 2

Let $L_2 = \{w \in \{a, b, c\}^* \mid w = (abc)^* a^*\}$. Design a NFA that accepts the language L_2.

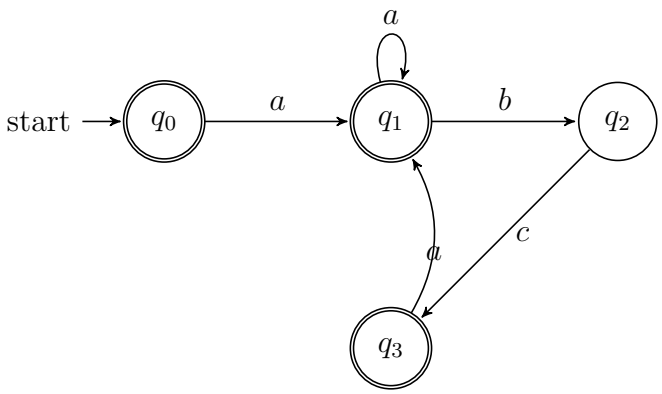
Answer:

The language consists of zero or more repetitions of abc, followed by zero or more a's.

States:

- q_0: Initial and accepting state
- q_1: Just read a (could be start of abc or final a's)
- q_2: Read ab (continuing abc)
- q_3: Read abc (back to accepting, or continue with more a's)

NFA Diagram:



Explanation:

- From q_0, we can either accept immediately (ε is in the language) or read a
- After reading a, we nondeterministically choose:
  - Continue with bc to complete an abc block
  - Stay in q_1 reading more a's (the final a^* part)
- After completing abc (reaching q_3), we can start another abc block or accept
- States q_0, q_1, and q_3 are accepting to handle $(abc)^* a^*$ properly

Problem 3

Let $L_3 = \{w \in \{0,1,2,3,4,5,6,7,8,9\}^* \mid$ the final digit of the string w has not appeared before in $w\}$.

Design a NFA that accepts the language L_3.

Answer:

The NFA uses nondeterminism to guess which digit will be the final digit that hasn't appeared before.

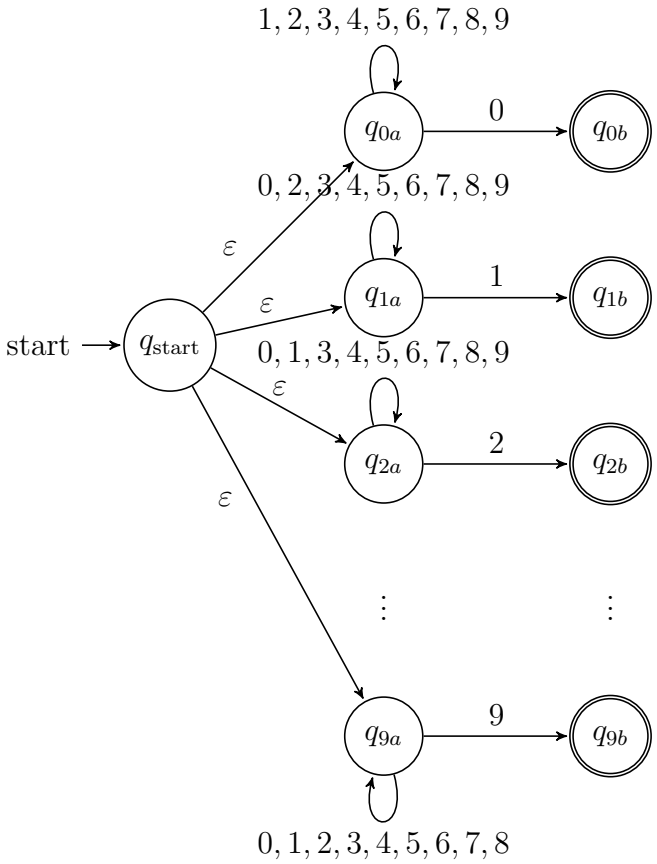
Strategy: The NFA nondeterministically guesses which digit will appear for the first time at the end. For each possible final digit, we have a separate branch that:

- Reads any other digits (but not that specific digit)
- Reads that specific digit once at the end
- Accepts

States:

- q_{start}: Initial state
- For digit 0: q_{0a} (reading, avoiding 0), q_{0b} (just read 0, accepting)
- For digit 1: q_{1a} (reading, avoiding 1), q_{1b} (just read 1, accepting)
- For digit 2: q_{2a} (reading, avoiding 2), q_{2b} (just read 2, accepting)
- ... (similar for digits 3-9)
- For digit 9: q_{9a} (reading, avoiding 9), q_{9b} (just read 9, accepting)

NFA Diagram:



Explanation:

The NFA has 21 total states: 1 start state + 10 branches (one for each digit 0-9), where each branch has 2 states.

How it works:

1. From q_{start}, use ε-transitions to nondeterministically guess which digit will be the final new digit
2. If we guess digit 0 will be the final new digit:
   - Go to state q_{0a}
   - Loop in q_{0a} reading any digit 1-9 (but never 0)
   - Read 0 and go to accepting state q_{0b}
3. If we guess digit 1 will be the final new digit:
   - Go to state q_{1a}
   - Loop in q_{1a} reading any digit except 1
   - Read 1 and go to accepting state q_{1b}
4. Similar logic applies for guessing digits 2 through 9

Example:
For string $w = 237$ (where 7 appears for the first time at the end):

- Start: q_{start}
- Guess 7 is final: $q_{\text{start}} \xrightarrow{\varepsilon} q_{7a}$
- Read 2: $q_{7a} \xrightarrow{2} q_{7a}$
- Read 3: $q_{7a} \xrightarrow{3} q_{7a}$
- Read 7: $q_{7a} \xrightarrow{7} q_{7b}$ (accept)

For string $w = 5$ (single digit, first occurrence):

- Start: q_{start}
- Guess 5 is final: $q_{\text{start}} \xrightarrow{\varepsilon} q_{5a}$
- Read 5: $q_{5a} \xrightarrow{5} q_{5b}$ (accept)