CMPSC 461: Programming Language Concepts, Fall 2025
Assignment 1 Practice Notes Packet

Prof. Suman Saha

August 25, 2025

Problem 1: Regular Expression I [5 pts]

Given an alphabet {0, 1, 2, ..., 8, 9} construct a regular expression for strings that matches any four character string that does not end with the digits 4, 6, 1. Come up and write two different answers (There are more than two correct answers).

Solution

$[0-9][0-9][0-9][023578-9]$ or $[0-9][0-9][0-9][\hat{4}61]$

1. $[0-9]$: matches any digit from 0 to 9.

2. $[\hat{4}61]$: matches any digit except 4,6,1.

Problem 2: Regular Expression II [5 pts]

Given an alphabet {a,b,c,d,e} construct a regular expression that matches strings that contain the sub string ”ace” and ends in the letter ’b’.

Solution

$[abcde]^*(ace)[abcde]^*b$

1. $[abcde]^*$: matches any single character in a,b,c,d,e 0 or more times.

2. (ace): matches ’ace’.

Problem 3: Regular Expression III [5 pts]

Assuming all valid digits and special characters, construct a regular expression that matches integers from ranges [-461,-311] and [461, 572].

Solution

Break down each range and create patterns that match the numbers within these ranges.

- $[-461, -460]$: $-46(0|1)$
- $[-459, -400]$: $-4[0-5][0-9]$
- $[-399, -320]$: $-3[2-9][0-9]$
- $[-319, -311]$: $-31[1-9]$
- $[461, 469]$: $46[1-9]$
- $[470, 499]$: $4[7-9][0-9]$
- $[500, 569]$: $5[0-6][0-9]$
- $[570, 572]$: $57[0-2]$

Combining these patterns:

$$-((46(0|1)|4[0-5][0-9])|3(1[1-9]|[2-9][0-9]))|(4(6[1-9]|[7-9][0-9])|5([0-6][0-9]|7[0-2]))$$

Problem 4: Regular Expression IV [5 pts]

Given the following regular expression a{4}b?[hello](goodluck) give the longest possible string that is matched by the regular expression. If it is infinite or null, explain why.

Solution

Break down the regex.

1. $a\{4\}$: 'a' must appear exactly 4 times.
2. $b?$: 'b' can appear 0 or 1 time.
3. $[hello]$: matches any single character from h, e, l, o
4. $(goodluck)$: matches "goodluck"

To maximize the length of the matched string, choose 'b' to appear 1 time.

$aaaabhgoodluck$

Problem 5: Regular Expression V [6 pts]

Write regular expressions for the following languages.

1. The set of strings of 0's and 1's whose tenth symbol from the right end is 1.
2. The set of strings of 0's and 1's with at most one pair of consecutive 1's.

Solution

1. The string can have any number of characters before this '1' and exactly 9 characters after this '1'.

   $(0 \mid 1)^*1(0 \mid 1)\{9\}$

2. Before and after the "11" substring, the string can have any combination of '0's and '1's, but no other "11" substrings. This means zero or more occurrences of either '0' or '1' before and after "11" while the substring before "11" dose not end with '1' and the substring after "11" does not start with '1'. Also note that string "11" can be optional.

   $(0 \mid 10)^*(11 \mid \epsilon)(0 \mid 01)^*$

Problem 6: Regular Expression VI [5 pts]

Give English descriptions of the languages of the following regular expressions.

1. $((10)^*000(01)^*$
2. $(0 \mid 10)^*1^*$

Solution

1. This is a language of binary strings with at least three consecutive 0s in the middle with possibility of at most 5 consecutive 0s when both left and right parts are non-empty.
2. This is the language of binary strings in which there are no two consecutive 1's, except for possibly a string of 1's at the end.

Problem 7: Regular Expression VII [5 pts]

Write down the set of strings recognized by the following regular expressions. If the set is infinite, write down the first 4 shortest elements.

1. $abc \mid (ed \mid f) \mid a$
2. $a^*(ab \mid b)^*$

Solution

1. {“a”, ”f”, ”ed”, “abc”}
2. {“”, “a”, “b”, “ab”}

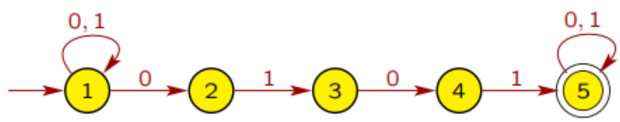
Problem 8: Finite Automata I [8 pts]

Give NFAs with the specified number of states recognizing each of the following languages. In all cases, the alphabet is $\sum = \{0, 1\}$.

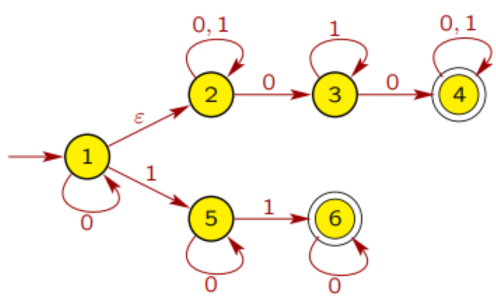
1. The language $\{ \text{ w} \in \sum^* \mid \text{w contains the substring 0101, i.e., w} = \text{x0101y for x, y} \in \sum^* \}$ with five states.
2. The language $\{ \text{ w} \in \sum^* \mid \text{w contains at least two 0s, or exactly two 1s} \}$ with six states.

Solution

1. Regular expression: $(0 \mid 1)^*(0101)(0 \mid 1)^*$ As the substring '0101' needs to be present, we are ensuring this in the regex, allowing it to be preceded and followed by either '0' or '1'.



2. Regular expression: $((0 \mid 1)^*0(1)^*0(0 \mid 1)^*) \mid (0^*10^*10^*)$ As there need to be atleast two 0s, or exactly two 1s

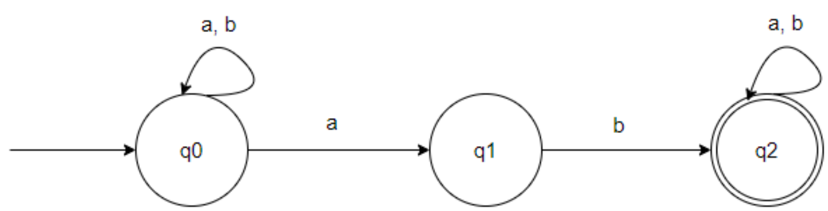


Problem 9: Finite Automata II [7 pts]

1. What is a Finite Automata? Explain how it is related to regular languages. [2 pts]
2. For the given alphabet $\{a, b\}$, design a NFA which contains **ab** as substring (*e.g., ab, aab, bab, abab, aaba etc.*). Neatly describe the **initial state**, **accepting state**, and **transitions**, if any. [5 pts]

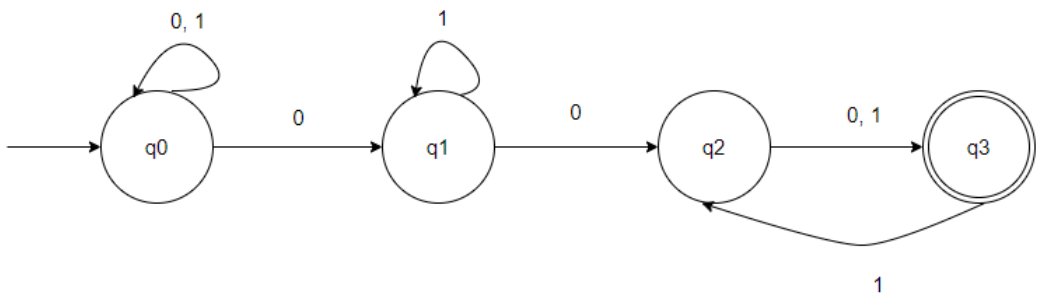
Solution

1. Finite Automata is the simplest machine to recognize patterns. Regular languages are a class of languages that are recognized by a Finite Automata.
2. Regular expression: $(a \mid b)^*(ab)(a \mid b)^*$.
   NFA -



Problem 10: Finite Automata III [10 pts]

1. For the NFA given below, construct a DFA using subset construction method. Neatly describe the transition tables associated with it. [7.5 pts]



2. Is the language described by the above NFA regular? Explain your reasoning. [2.5 pts]

Solution

1.

NFA table		
state	0	1
q_0	q_0, q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	-	q_2

Since there's no ϵ transitions, the start state of our DFA would be q_0.

Initialize the table:

state	0	1
q_0	q_0q_1	q_0

Find destination states for q_0q_1. On input '0', q_0 goes to q_0 and q_1; and q_1 goes to q_2, so q_0q_1 goes to $q_0q_1q_2$ on '1'. On input '1', q_0 goes to q_0 and q_1 goes q_1, so q_0q_1 goes to q_0q_1.

state	0	1
q_0	q_0q_1	q_0
q_0q_1	$q_0q_1q_2$	q_0q_1

We got a new state $q_0q_1q_2$. Find destination states for it. On input '0', q_2 goes to q_3 and q_0q_1 goes to $q_0q_1q_2$, so $q_0q_1q_2$ goes to $q_0q_1q_2q_3$. On input '1', q_2 goes to q_3 and q_0q_1 goes to q_0q_1, so $q_0q_1q_2$ goes to $q_0q_1q_3$.

state	0	1
q_0	q_0q_1	q_0
q_0q_1	$q_0q_1q_2$	q_0q_1
$q_0q_1q_2$	$q_0q_1q_2q_3$	$q_0q_1q_3$

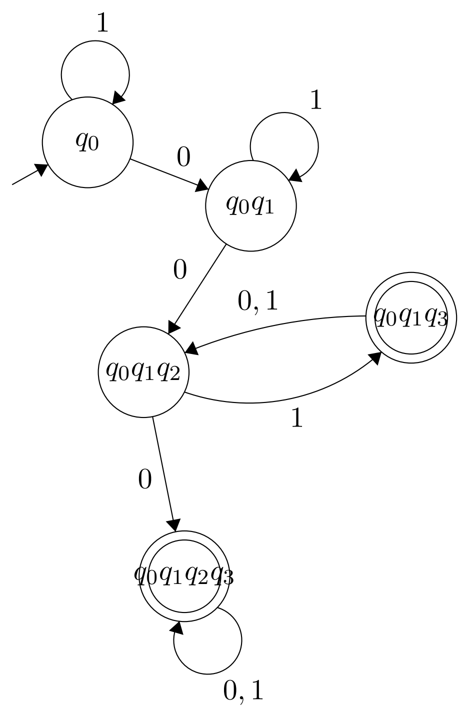
Repeat this process for new states $q_0q_1q_2q_3$ again. On input '0', $q_0q_1q_2$ goes to $q_0q_1q_2q_3$ so $q_0q_1q_2q_3$. On input '1', $q_0q_1q_2$ goes to $q_0q_1q_3$ and q_3 goes to q_2, so $q_0q_1q_2q_3$ goes to $q_0q_1q_2q_3$.

state	0	1
q_0	q_0q_1	q_0
q_0q_1	$q_0q_1q_2$	q_0q_1
$q_0q_1q_2$	$q_0q_1q_2q_3$	$q_0q_1q_3$
$q_0q_1q_2q_3$	$q_0q_1q_2q_3$	$q_0q_1q_2q_3$

Don't forget $q_0q_1q_3$ we got before. On input '0', q_0q_1 goes to $q_0q_1q_2$ so $q_0q_1q_3$ goes to $q_0q_1q_2$. On input '1', q_0q_1 goes to q_0q_1 and q_3 goes to q_2, so $q_0q_1q_2$ goes to $q_0q_1q_2$.

DFA table		
state	0	1
q_0	q_0, q_1	q_0
q_0, q_1	q_0, q_1, q_2	q_0, q_1
q_0, q_1, q_2	q_0, q_1, q_2, q_3	q_0, q_1, q_3
q_0, q_1, q_2, q_3	q_0, q_1, q_2, q_3	q_0, q_1, q_2, q_3
q_0, q_1, q_3	q_0, q_1, q_2	q_0, q_1, q_2

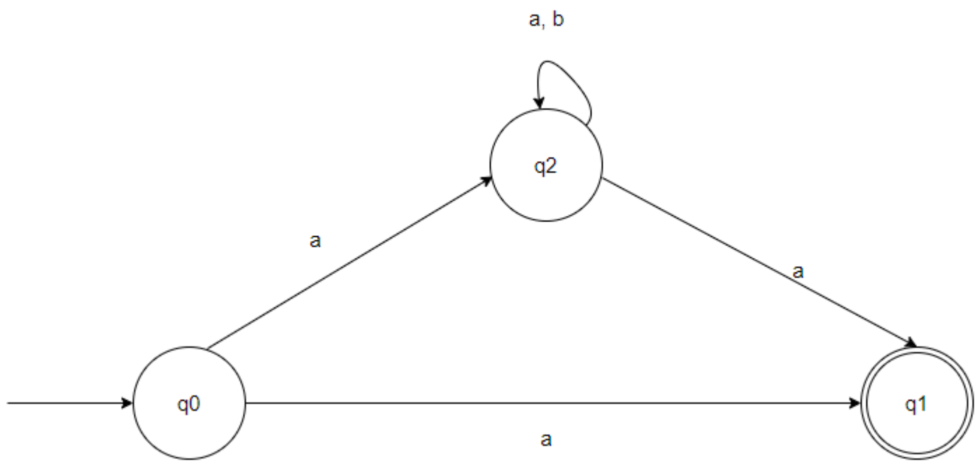
Any state that contains q_3 would be an accepting state.



2. The language described by the NFA is regular because there exists a DFA that recognizes the same language.

Problem 11: Finite Automata IV [10 pts]

1. For the NFA given below, construct a DFA using subset construction method. Neatly describe the transition tables associated with it. [7.5 pts]



2. Is the language described by the above NFA regular? Explain your reasoning. [2.5 pts]

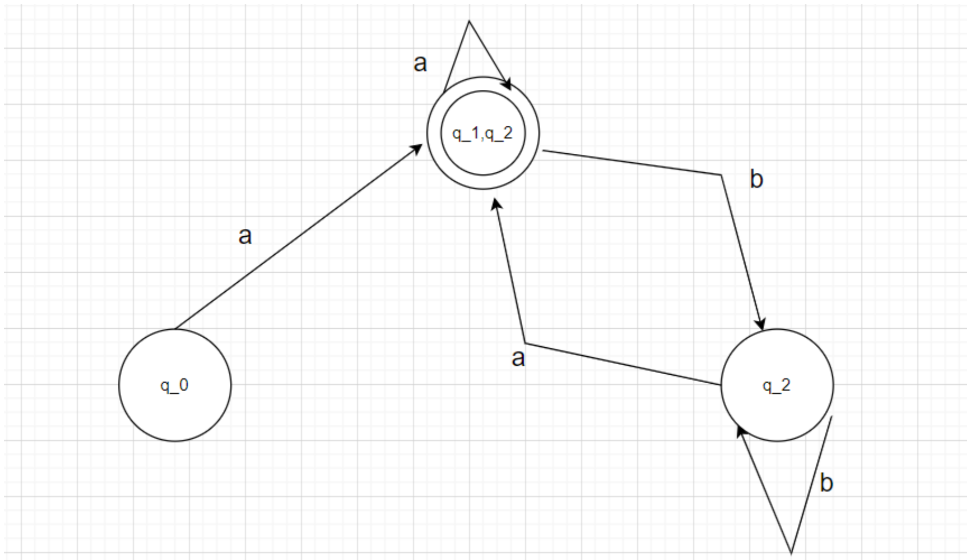
Solution

1.

NFA table		
state	a	b
q_0	q_1, q_2	-
q_1	-	-
q_2	q_1, q_2	q_2

q0 on input 'a' goes to q1,q2.
q1 on input 'a' goes to ϕ. q2 on input 'a' goes to q1,q2 hence q1,q2 on input 'a' goes to q1,q2.
q1 on input 'b' goes to ϕ. q2 on input 'b' goes to q2 hence q1,q2 on 'b' goes to q2.
q2 on input 'a' goes to q1,q2 and on input 'b' goes to q2.

DFA table		
state	a	b
q_0	q_1, q_2	-
q_1, q_2	q_1, q_2	q_2
q_2	q_1, q_2	q_2



2. The language described by the NFA is regular because there exists a DFA that recognizes the same language.

Problem 12: Finite Automata V [8 pts]

Draw a deterministic finite automata which accepts a string containing “ing” at the end of a string in a string of a-z, e.g., “anything” but not “anywhere”.

Solution Let’s consider q0 as the start state. Since the string must end with ’ing’, q0 transitions to q1 on input ’i’. For any other input (i.e., any letter from a-z except ’i’), we loop in q0 and wait until we get an ’i’. q1 transitions to q2 on input ’n’. q1 on input ’i’ loops back to itself and waits for an ’n’. q1 transitions back to q0 on input of any letter besides ’i’ or ’n’. On input ’g’, q2 transitions to q3, the accepting state. If q2 receives an ’i’, it transitions back to q1 and waits for an ’n’. If any other letter is input (i.e., any letter from a-z except ’i’ and ’g’), q2 transitions back to q0 and starts searching for the pattern again.