Ben Smith
Email: bxs566@case.edu
Course: CSDS 337 - Compiler Design
Instructor: Dr. Vipin Chaudhary

Problem Set - 2
ID: 3559750
Term: Spring 2024
Due Date: 14^{th} February, 2024

Number of hours delay for this Problem Set: 2
Cumulative number of hours delay so far: 26

I discussed this homework with: Jackson Schuetzle

SUBMISSION GUIDELINES: Submit a zip file that includes the written answers and the flex file for Problem 4.

Problem 1 - 5 points
Describe the language denoted by the following regular expression?
$(aa|bb)^*((ab|ba)(aa|bb)^*(ab|ba)(aa|bb)^*)^*$

Solution: This language is the set of all strings over the alphabet $\{a, b\}$ that contain an even number of a's and an even number of b's.

Problem 2 - 35 points
Write regular definitions for the following languages:

a All strings of lowercase letters that contain the five vowels in reverse order.

b Binary strings that has at least 3 characters, and the third character is 0.

c Binary strings that has number of 0s which is a multiple of 3

d Binary strings that starts and ends with the same character

e Binary strings that has odd length

f Binary strings that starts with 0 and has odd length, or starts with 1 and has even length

g Binary strings whose length is at least 1 and at most 3

Solution: Your solutions go here

a consonant $\longrightarrow ([bcd][fgh][j-n][p-t][v-z])^*a([bcd][fgh][j-n][p-t][v-z])$
answer $\longrightarrow$ (consonant)$^*u^+$(consonant)$^*o^+$(consonant)$^*i^+$(consonant)$^*e^+$(consonant)$^*a^+$

b $(0|1)(0|1)0(0|1)^*$

c $(1^*01^*01^*01^*)^*$

d $(0((0|1)^*0)?)|(1((0|1)^*1)?)$

e $(1|0)((0|1)(0|1))^*$

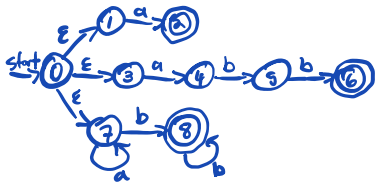
f $(0|(1(0|1)))((0|1)(0|1))^*$

g $(0|1)(0|1)?(0|1)?$

Problem 3 - 10 points
Provide transition diagram as an NFA to recognize the language represented by $a|abb|a^*b^+$. Convert this NFA to a DFA and show all steps.

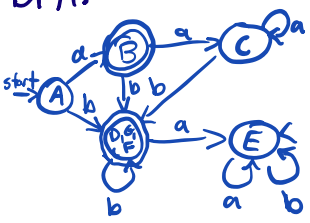
Solution:

NFA:



ε-closure({0}) = {1, 3, 7} → A
move({1, 3, 7}, a) = {2, 4, 7}
ε-closure({2, 4, 7}) = {2, 4, 7} → B
move({2, 4, 7}, a) = {7}
ε-closure({7}) = {7} → C
move({7}, a) = {7}
ε-closure({7}) = {7}
move({7}, b) = {8}
ε-closure({8}) = {8} → D
move({8}, a) = {}
ε-closure({}) = {} → E
move({8}, b) = {8}
ε-closure({8}) = {8}
move({2, 4, 7}, b) = {5, 8}
ε-closure({5, 8}) = {5, 8} → F
move({5, 8}, a) = {}
ε-closure({}) = {}
move({5, 8}, b) = {6, 8}
ε-closure({6, 8}) = {6, 8} → G
move({6, 8}, a) = {}
ε-closure({}) = {}
move({6, 8}, b) = {8}
ε-closure({8}) = {8}
move({1, 3, 7}, b) = {8}
ε-closure({8}) = {8}

DFA:



NFA State	DFA State	a	b
{0, 1, 3, 7}	A	B	D
{2, 4, 7}	B	C	F
{7}	C	C	D
{8}	D	E	D
{}	E	E	E
{5, 8}	F	E	G
{6, 8}	G	E	D

Figure 1: The NFA and DFA, with explanation.

Problem 4 - 50 points
Write a flex program which does the following:

- reads multiple input files
- for each file:
  - it prints the number of characters, number words and number of lines
  - it replaces more than one contiguous space by a single space
  - it prints the number of single line C comments
  - it prints the number of multiple line C comments
  - it prints the number of occurrences of each of these keywords: for, do, and $while$
- all the above counts are printed for each file in order and a cumulative number for all the files is also printed at the end
- the entire output is printed to a file named "problem4output"
- the output should clearly indicate what each of the count indicates
- the flex file should be named "problem4lex.l"