CPSC 323 Project 1

Names:

You are allowed to write your project in C/C++/Java/Python but you ARE NOT allowed to use **Yacc, Bison, or any other items similar** that assists in the creation of compilers. This includes **NO Regular Expression libraries**, The scope of this project is to understand how lexical analysis works and design the transition table specific to a programming language. This is directly **applicable to your exams**, so again, **know how it works**. You will build and make use of **Deterministic Finite State Automata** to define items needed for programming. Although the lecture focuses on the **academic** side, the project is designed to give you a **practical** hands-on approach to the subject of compilers.

Project 1 consists of a number of questions to be answered in writing along with three programs to be submitted online. The answers need to be uploaded on Canvas before the deadline. Each program needs to have the specified name in the problem description. All three programs need to be zipped into a single file called Project1.zip and upload it on TITANium under **Project 1**. The files will be decompressed, compiled and executed in the same directory. Maximum 80 points.

1. [6 points] True or False (1 point each)

|  |  |
| --- | --- |
| babbaa is in a\*(ba)\* | a(ab)\* = a(ba)\*b |
| b\*a\*b\*= (bab)\* | (a | b)\* = a\*b\* |
| bab is in a\*(ba)\* | (aa\*)(a **|** ɛ ) = a\* |

1. [28 points] Given the language L= a (a | b\*) ab\* over the alphabet { a, b } :
2. Construct an FSM for L (can be NFSM or a DFSM). You may either draw the FSM or describe it formally, but the states Q, the start state q0, the accepting states F, and the transition function δ must be clearly specified. [3 points]
3. Construct the DFSM that accepts L. If you wrote a NFSM, then convert it to a DFSM. You may either draw the DFSM or describe it formally, but the states Q, the start state q0, the accepting states F, and the transition function δ must be clearly specified. [6 points]
4. Use the transition function δ as part of a program that determines whether a given string is accepted or not by the DFA. Write a program to read from the user a string of characters ending with $, for the DFA of the language L to determine whether the string is accepted or rejected by L.   
   Example:  
   input = aaababa$ output=NO   
   input = **a**babb$ output = YES.   
   Save it as Prog1and upload it on Canvas. [14 points]
5. Include a snapshot of the lines of code in the program implementing the transition table [5 points]
6. Convert the non-deterministic FA (NDFA) to a deterministic FA (DFA) [6 points]
7. Write a program to read from the user a string of characters, ending with $, for the FA of the language L to determine whether the following words are accepted or rejected by L. For example, for the input abbbaba$ the output should be YES while for the input baabba$ the output should be NO. Save it as **Prog1** and upload to zip file. [14 points]
8. [10 points] Let L be the set of all strings in {0, 1}\* that contain at least two occurrences of the substring 101. Hint: The shortest string in L has the length 5; for example, the substring "10101" contains two occurrences of the substring 101.

(a) Give a regular expression that describes that language, and briefly argue why your regular expression is correct. [4 points]

(b) Describe a DFSM over the alphabet Σ = {0, 1} that accepts the language L. You may either draw the DFSM or describe it formally, but the states Q, the start state q0, the accepting states F, and the transition function δ must be clearly specified. [6 points]

1. [12 points] Given a text file “file.txt”, write a program to copy the file into file “clean.txt” by removing all comment lines (lines that begin with //), blank lines and extra white spaces. Place one space before and space after each token. Note that the cin>> is one lexeme. Save it as Prog2 and upload it on Canvas.

|  |  |
| --- | --- |
| Sample input “file.txt” file | Correct output “clean.txt” |
| //declare variables  int num1, nume2, num3 ,t ;  //initialize variables  num1 =13; num2 = 55 ;  num3 = 17 ;  cin>> t ;  //compute and display their total  num1= num2 + num3 +t; //compute sum  t = t+num1; //compute new value for tl | int num1 , nume2 , num3 , t ;  num1 = 13 ;  num2 = 55 ;  num3 = 17 ;  cin>> t ;  num1 = num2 + num3 + t ;  t = t + num1 ; |

1. [24 points] Write a program to read from the user a single line (up to 255 characters) at a time and splits the line into lexemes, if possible. The program will determine whether each lexeme is a reserved word, an arithmetic operator, a special symbol, a number, or an identifier. The following arrays are already given:

char reservedWords[4][10]={ “cin>>”, “for”, “int”, “while”};

char operator[6][2]={ “+”, “-”, “\*”, “/”, “++”, “--”}

char special[7][3]={“>”, “=” , “;” , “(“ , “)” , “>=” ,“,”};

A number is an unsigned integer or a signed integer.

An identifier starts with a letter or \_ followed by any number of letters, \_, and digits, and excludes any reserved word.

For each lexeme check whether it is a reserved word. If not, check whether it is a special symbol. If not, check whether it is a number. If not, check whether it is an identifier. If none of these, display “invalid”. The program must be case insensitive. Save it as Prog3 and upload it on Canvas.

**Sample I/O:**

Enter a statement:

cin>> 2i ; For ( int j = 100 ; j >= i ; j = **--** j - 1 )

cin>> reserved word

2i invalid

; special symbol

For reserved word

( special symbol

int reserved word

j identifier

= special symbol

100 number

; special symbol

j identifier

>= special symbol

i identifier

; special symbol

j identifier

= special symbol

-- operator

j identifier

- operator

1 number

) special symbol

CONTINUE(y/n)? y

Enter a statement: int j = -5 ; while ( j < 10 ) cin>> myTime ;

int reserved word

j identifier

= special symbol

-5 number

; special symbol

while reserved word

( special symbol

j identifier

< invalid

10 number

) special symbol

cin>> reserved word

myTime identifier

; special symbol

CONTINUE(y/n)? n