Complier Lab(CSE4112)

Assignment#1

Constructing a Lexical Analyzer

Assignment details

Construct a lexical analyzer generator `lexergen.exe` or `lexergen.o` that takes a lexer specification file `mylex.l` as input. The file `mylex.l` contains a set of regular expressions and the corresponding tokens where the input alphabet is $\sum$ = {a,b}. `lexergen.exe` or `lexergen.o` will produce a lexical analyzer `mylex.exe` or `mylex.o` according to the input specifications file `mylex.l`. `mylex.exe` or `mylex.o` should be able to take a input string and return the lexemes that matches with the regular expressions and the corresponding tokens.

ALTERNATIVELY, your program lexergen.exe can take as input the mylex.l and the input string and produce the list of matched lexemes and tokens output.

The regular expressions may contain the following operators: '.', '|', '*', '+' (positive closure), ?, and symbols: () and epsilon (we will use e instead of ε) .

We will follow lex/flex's conventions for matching

1. If it finds more than one match, it takes the one matching the most text
2. If it finds two or more matches of the same length, the rule listed first in the `mylex.l` input file is chosen.

Also assume the input string doesn't contain any illegal characters and the whole input string can be divided into the token to be recognized. Extra credit will be awarded if your lexer can handle error case –i.e, skip illegal characters and portions of the input string that doesn't match with any of the regular expressions.

Implementation

Construct DFAs from the given regular expressions using the direct DFA construction algorithm that uses the firstpos, lastpos, and followpos functions. You may want to convert the regular expressions to postfix notation first and rather than creating a tree use a stack to find the firstpos, lastpos, and followpos for each 'node'. You can use C/C++ or Java for implementation.

Sample input and output are given below:

Sample mylex.l file

```
a+ TOKEN1
a|b|e).a.b* TOKEN2
(aa)+(bb)+ TOKEN3
```

Sample input for mylex.exe

```
aabbbbaaaa
```

Sample output for mylex.exe

```
aabbbb TOKEN2
aaaa TOKEN1
```