CS301P Compiler Design Laboratory Exercises Week-1

Date: Aug 05 2024

Objectives

- To learn a tool *Lex: lexical analyzer generator* through simple exercises
- To learn writing make files

Exercise Problems

1. Construct a Lexical Analyzer accepting each of the following languages.

   (a) $\{w \in \{0,1\}^* \mid w$ contains 111 as a substring and does not contain 00 as a substring$\}$

   (b) $\{w \in \{x,y,z\}^* \mid$ in w the number of xs modulo 2 is equal to the number of ys modulo 3$\}$

2. Construct a Lexical Analyzer for the following. Given an alphabet $\sum = \{a,b,c,d\}$, L is the set of strings $xyzwy$, where x and w are strings of one or more characters in $\sum$, y is any single character in $\sum$ and z is the character z, taken from outside the alphabet. (Each string xyzwy contains two words xy and wy built from letters in $\sum$. The words end in the same letter, y. They are separated by z.)

3. Construct a Lexical Analyzer that tokenizes various components of a URL. The following shows the general structure of URL.

   (a) Scheme: The scheme indicates the protocol or method used to access the resource. Common schemes include http, https, ftp, mailto, and file.

   (b) Username and Password: These components are optional and provide authentication credentials for accessing the resource. The format is username:password@.

   (c) Host: The host specifies the domain name or IP address of the server where the resource is located.

   (d) Port: The port number is optional and specifies the network port on the host to connect to.

   (e) Path: The path represents the location of the resource on the server's file system. It is typically a hierarchical structure of directories and filenames. This is an optional one.

   (f) Query Parameters: Query parameters are optional and are used to pass data to the server in the form of key-value pairs. They are separated from the path by a ? and from each other by &.

   (g) Fragment: The fragment is optional and represents a specific section or anchor within the resource, often used in web pages to scroll to a specific part of the page.

   Here are a few examples to illustrate the structure:

http://www.example.com:8080/path/to/resource?param1=value1&param2=value2

https://secure.example.com/path/file.html

ftp://ftp.example.com/files/file.txt

mailto:contact@example.com

file:///path/to/local/file.html

References

1. Flex Manual https://westes.github.io/flex/manual/
2. GNU Make Manual https://www.gnu.org/software/make/manual/html_node/index.html

Submission Guidelines

The assignment should be submitted in the following format.

1. All files and folders should be lowercase letters
2. Create a folder with name "yourrollnumber_lab1" (roll no CS21B001, the directory name should be cs21b001_lab1).
3. Prepare separate lex file for each problem.
4. The lex files and makefile should be named as prob1.l, prob2.l, prob3.l and Makefile
5. The final target for each problem should be be named as **lexer**. That is, makefile should generate final executable file named **lexer**.
6. The input must be given through a file and the file name should be taken through command line arguments.
7. Inorder to test your lexer, submit four input files as well. The names of these input files should be inp1, inp2, and inp3. No extension is needed. These test files should verify different aspects of your lexer.
8. Copy all the files (including self declaration form) into the folder created in Step 2.
9. Don't submit any unrelated or executable files.
10. Finally tar and compress yourrollnumber_lab1 directory as yourrollnumber_lab1.tar.gz and upload the same to the course page at Google Classroom before the due date.

Evaluation Guidelines

- Complaince - adherance to the instructions, naming conventions, code readability, comments, beautification, and quality of content in readme file. (10marks)
- Correctness - logic/approach, working makefile, compilation with no errors and warnings. (10marks)
- Completeness - ability to work for different testcases and presentation of the output (i.e., output beautification). (10marks)

Academic Honesty

Any kind of copying, sharing code with others, and malpractices attract high penalties to the extent of referring to the **Institute Level Disciplinary committee**. To this end, your submission must include a signed self declaration in the following format.

Declaration of Academic Honesty

I declare that i) the assignment here submitted is original except for source material explicitly acknowledged in the **readme** file; ii) I have not distributed or shared the assignment either wholly or partly to the fellow students of the course - Compiler Design; iii) I have not copied the assignment either partly or wholly from the fellow students of the course.

I am aware that I will be held responsible and liable to disciplinary actions if I am found guilty.

Full Name:

Roll No:

Date:

Signature of the student