Compiler Design Assignment - Lexical Analysis

Venkatesan Subramanian is with IIIT Allahabad
Kalaivany Natarajan was with University of South Australia
Pallapa Venkataram with IISc Bangalore

1 COMPILER DESIGN CONCEPT MAP

The compiler design concept map in shown figure 1, which includes the pre-processor and pre-requisite for this subject such as instruction set, Context Free Grammar (CFG) and Context Sensitive Grammar (CSG), Regular Expression (RE), Finite State Machine (FSM) and Push Down Automata (PDA). The concept map gives the complete overview of compiler design course and relationship among the concepts. This makes a student to understand the importance of Theory of Computation (ToC) subject since they study RE, FSM, PDA and CFG/CSG.

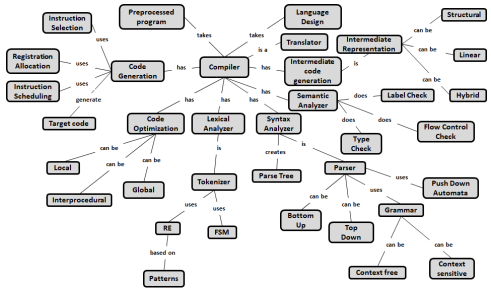

Figure 1: Basic concept map of Compiler Design

1.1 Lexical Analysis

The figure 2 shows the extended concept map of lexical analyser [1][9]. The industry and research example for the core concepts in the extended concept map are as follows.

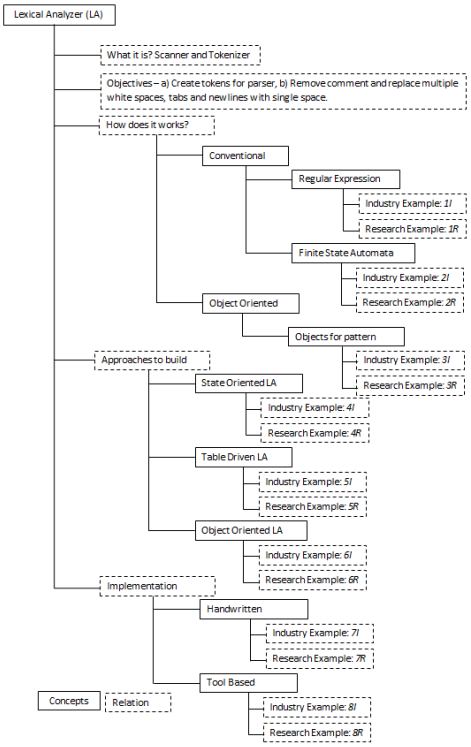

Figure 2: Concepts of Lexical Analyzer

- *1I*: Use the regular expression for a medical industry to identify the external analytic vendor whoever is different from the approved list.

- *1R*: Use text manipulation (update, delete, search) in the text file database using the regular expression [2]. The extensive facilities provided by the databases such as MySQL, Oracle, SQL server, etc are not required and we can free ourselves from the additional work of database soft-

ware installation, knowing a query language etc. In that case flat files may play an useful role as a database.

- *2I*: Finite State Machine (FSM) can be used to execute sequence of tests for a flow measurement testing. This solution can be applied to automate other serial and batch processes [3].

- *2R*: We can design a process flow for managing enterprise documents and identify all of the possible states that a document can be in; and also, to identify the corresponding actions which allow the documents to transition between states. Software usually allows auto deleting documents once they have been in a final state for a given number of days, months or years. This is useful for compliance scenarios in which some documents must be kept, archived and later deleted following strict rules in accordance with legal requirements. [4]

- *3I & 6I*: An object-oriented lexical analyzer for the compiler construction, which can simplify the design effort and it permits code re-use [5].

- *3R & 6R*: An industry handling the big data and need to analyse the input/data can use the object based scanner to recognize.

- *4I*: The state oriented lexical analyser can be designed to detect the SQL injection attack.[6]

- *4R*: Use the state oriented lexical analyser for sentimental analysis of an entertainment industry for the prediction of movies. Identify the states and transition for the prediction of the movie [7].

- *5I*: Design the table driven lexical analyser tool to identify the patterns of a Go language.

- *5R*: Design the table driven Lexical analyser to analyse the log file of an online product selling industry.

- *7I*: Design the handwritten scanner using the state oriented/table driven/object oriented to extract the useful information from an email message to categorize the emails.

- *7R*: Design the handwritten tokenizer using the state oriented, table driven or object oriented for the analysis of natural language data.

- *8I*: Design the automated tokenization tool for an online product selling industry that can be reused for the cyber security industry which need to scan the packets datagram to identify the attack if any.

- *8R*: Design the automated lexical analysis tool for sentiment, social cognition,and social-order analysis [8].

Like above, for all concepts in the lexical analyzer, your team has to prepare the assignments considering the recent research and industry requirement

References

[1] A. V. Aho, M. S. Lam, R. Sethi and J. D. Ullman, Compilers: Principles, Techniques and Tools, Second Edition Book, 2006.

[2] S. Biswas, D.Sengupta, R. Bhattacharjee and M. Handique, Text Manipulation Using Regular Expressions, *Proceedings of the 6th International Conference on Advanced Computing*, pp.62-67, 2016.

[3] A. Jamhour and C. Garcia, Automation of industrial serial processes based on finite state machines, *Proceedings of the 20th International Congress of Chemical and Process Engineering*, pp.186196, 2012.

[4] J. Ramirez, Finite-state machines: Better than flowcharts, URL:

https://www.itproportal.com/features/finite-state-machines-better-than-flowcharts/ [Last accessed on 11/02/2020]

[5] A. Schreiner and B. Kühl, Object-oriented Compiler Construction URL: http://citeseerx.ist.psu.edu/ viewdoc/download?doi=10.1.1.201.4043&rep=rep1&type=pdf [Last accessed on 11/02/2020]

[6] L. Ntagwabira and S. L. Kang, Use of query tokenization to detect and prevent sql injection attacks, *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, Vol. 2, pp. 438-440, 2010.

[7] M. P. M. Raj and S. Aditya, Predictive model for movies success and sentiment analysis, *Research Journal of Management Sciences*, Vol.6, pp.1-19, 2017.

[8] S. A. Crossley, K. Kyle and D. S. McNamara, Sentiment Analysis and Social Cognition Engine (SEANCE):An automatic tool for sentiment, social cognition,and social-order analysis, *Journal of Behavior Research Methods*, Vol.49, pp.803-821, 2017.

[9] PART I: Lexical Analysis, URL: http://infolab.stanford.edu/ ullman/dragon/slides1.pdf, [Last accessed on 11/02/2020]