Compiler Design Assignment - Syntax Analysis

Venkatesan Subramanian is with IIIT Allahabad Kalaivany Natarajan was with University of South Australia Pallapa Venkataram is 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 Syntax Analysis

Syntax Analysis takes the input as tokens from the lexical phase and produces the syntax tree, which will be used by the semantic phase. The major task of the syntax analyzer is to check whether the syntax presents in the program is part of the programming language or not. To do this check, Syntax Analyzer uses parser with Finite State Machine, Push down Automata, Context Free and Sensitive Grammar. Parser can be top down or bottom up approach and it will be chosen based on the developer requirement. The LL (Left-to-right, Leftmost derivation) and LR (Left-to-right, Rightmost derivation) parsers work only in an unambiguous grammar to parse in linear time. For LL parser, an unambiguous, deterministic and non-left recursive grammar will be taken as input and computes the *first* and *follow*. Using the *first* and *follow*, the parsing table will be constructed to do parsing. In case of the LR parser, the finite state machine based item set for the unambiguous grammar will be generated and parsing table will be constructed from it to do parsing. The universal parsers like Earley's and CYK are available that can take any type of grammar and parse however the complexity of syntax validation will be more. The figure 2 shows the extended concept map of syntax analyser. The industry and research example for each of the core concepts in the extended concept map are as follows.

- 11: ALEX Resume Parser [1] offers unequaled accuracy and speed while converting resumes and CVs into XML or JSON output. It can be used in the medical industry to parse more resume and CVs.
- 1R: An intrusion detection method based on data parsing can be implemented using finite automaton [2].



Figure 2: Concepts of Syntax Analyzer

- 21: A type of formal grammar, which should be a probabilistic/stochastic context-free grammar that is context free grammar with weight for automatic generation of unique social media profiles using Facebook as a test case. The grammar should describe a language to have a valid (semantically relatable) social media profiles using a subset of Facebook (FB) fields [3].
- 2R: To specify the interesting paths, context-free grammars can be used while querying vertex- and edge-labeled graphs.[4].
- 31: Finite State Machine 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 [?].
- 3R: The most part of a network test research is devoted to methods based on graph theory, namely, on the theory of finite state machines (FSM). The network

protocol, such as IPv6, provokes a necessity to carry out the software testing [5]. The FSM can be used to test it.

- 41: In limited network bandwidth environment or resource-constrained computing devices, the overhead caused by Simple Object Access Protocol (SOAP) is disadvantageous. Even though binary representations of XML data were used, due to the special characteristics of SOAP communication most of these approaches are not applicable for web services. A custom pushdown automaton can be generated not only for parsing but also as a compressor [6].
- 4R:Pushdown Automata in Statistical Machine Translation using context-free grammar [7]
- 51: Make a parse tree for the SQL injection query "select * from students where name = 'Alice' or 1 > 0" considering the relational algebra.
- 51: The industry have a valuable business information in the database and this highly motivate attackers. Hence special care needs to be taken to prevent any malicious access. An approach for modeling SQL statements to apply machine learning techniques, such as clustering or outlier detection can be designed, in order to detect malicious behaviour at the database transaction [8]. The approach needs to use parse tree structure of SQL queries as characteristic E.g. for correlating SQL queries with applications and distinguishing benign and malicious queries.
- 61: The professional applications need to have error handling concept to trap unexpected errors. A consistent error handler can make sure that when crashes occur, the user is informed and program exits properly [9].
- 6R: Soft errors known as transient faults may lead to application crashes or silent

data corruption (SDC) that could result in incorrect program outputs. A compilerdirected error recovery scheme can provide fine-grained and guaranteed recovery without excessive performance and hardware overhead [10].

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

References

- [1] HireAbility Smart Parsing Solutions, URL: https://www.hireability.com/products/, [Last Accessed on 14/02/2020].
- [2] D. Qiu, S. Liao, C. Jia and D. Xu, Intrusion detection technology based on data parsing and finite automaton, *Proceedings of the* 5th International Conference on Electronics Information and Emergency Communication, pp.67-70, 2015.
- [3] A. Ade-Ibijola, Synthesis of Social Media Proles Using a Probabilistic Context-Free Grammar, Proceedings of the International Conference on Pattern Recognition Association of South Africa and Robotics and Mechatronics, pp. 104-109, 2017.
- [4] P. Sevon and L. Eronen, Subgraph Queries by Context-free Grammars, *Journal of Integrative Bioinformatics*, Vol.5, No.2, 2008.
- [5] A. Abdel-Karim Helal Abu-Ein, Hazem (Moh'd Said) Abdel Majid Hatamleh and Ahmed A.M. Sharadqeh, Using Finite State Machine at the Testing of Network Protocols, Australian Journal of Basic and Applied Sciences, Vol. 5, No. 10, pp. 956-960, 2011.
- [6] C. Werner, C. Buschmann, Y. Brandt and S. Fischer, Compressing SOAP Messages by using Pushdown Automata, *Proceedings* of the International Conference on Web Services, 2006.

- [7] C. Allauzen, B. Byrne, A. d. Gispert, G. Iglesia and M. Riley, Pushdown Automata in Statistical Machine Translation, *Association for Computational Linguistics*, Vol,20, No.3, pp.687-723, 2014.
- [8] C. Bockermann, M. Apel and M. Meier, Learning SQL for Database Intrusion Detection using Context-Sensitive Modelling, Proceedings of the 6th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp.196-205, 2009.
- [9] L. Chung, Error Handling and Debugging Tips and Techniques for Microsoft Access, VBA, and Visual Basic 6 (VB6), URL: https://www.fmsinc.com/tpapers/vbacode/ Debug.asp [Last accessed on 13/02/2020]
- [10] Q. Liu, C. Jung, D. Lee and D.h Tiwari, Compiler-Directed Lightweight Checkpointing for Fine-Grained Guaranteed Soft Error Recovery, Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp.228-239, 2016.