



Review report of a final thesis

Reviewer: Ing. Tomáš Pecka
Student: Aleksandr Levin
Thesis title: Parsing of General Expressions and Describing Programming Language Syntax using Expressions
Branch / specialization: Computer Science 2021
Created on: 11 June 2024

Evaluation criteria

1. Fulfillment of the assignment

- ▶ [1] assignment fulfilled
- [2] assignment fulfilled with minor objections
- [3] assignment fulfilled with major objections
- [4] assignment not fulfilled

Assignment is fulfilled.

2. Main written part

52 / 100 (E)

The submitted thesis addresses the design of a parser for general expressions based on an operator precedence table. Some parts of the thesis are unnecessary, and several basic concepts are not adequately defined (see details below).

The language used in the thesis is generally good. I have identified only a few language errors. The typographic aspect is satisfactory, there are multiple oversights: code listings are usually not referenced, the string literal in code listing 4.3 is incorrectly colored, references to table 5.1 are often connected to the preceding word, and table 5.1 itself should likely have been placed earlier in the chapter, among other issues.

In section 1, I would expect a review of the existing literature on the topics discussed in the thesis. Most parts of Chapter 2 are not used later in the thesis. It seems unnecessary to define DFAs, Turing Machines, etc., when these concepts are not utilized throughout the thesis. While the author provides formal definitions for various types of automata, he does not define other related concepts (e.g., run on the word). The same applies to the mention of Backus-Naur Form. On the other hand, some important definitions are missing. For instance, while the thesis deals with grammars, which are defined, there is no definition of a (leftmost) derivation of a word in a grammar. I also did not find definitions for an LL grammar/language, LL parser and its construction (both nondeterministic and using parsing table), or the LL parsing process. Recursive descent

parsing is mentioned but insufficiently defined. Abstract syntax trees are mentioned but not defined, nor is an example provided. This chapter does not provide any examples to aid the reader's understanding.

Section 2.9 closely resembles material from <https://courses.fit.cvut.cz/BI-PJP/materials/expressions.html> (wording is almost identical) but there is no citation and it is unattributed in the bibliography! I think that this might violate the citation ethics.

The example in section 3.5.2.1 highlights the importance of specifying the grammar's quadruple precisely. It is unclear from the example whether B1 is single symbol `B_1` or the symbol B followed by the symbol 1. Section 4.1 is not very clear to me. However, the conclusion of the second part of the thesis in section 5 appears logical to me but I would expect that the thesis digs a little bit more into the problem. Some parts of section 6 would be better as an attachment, and in the README file of the project. I do not see relevance of section 7.2 in the text (CMake was not mentioned before).

Overall, it seems the written part was completed in haste and there is considerable room for improvement of the written part.

3. Non-written part, attachments

57 /100 (E)

Unfortunately, the code lacks comments, class documentation, and a README file. Despite the project's small size, it took me quite a while to familiarize myself with it. I am not sure about the usage of the project. Should it be used as a library? Or is it just a standalone program? I think it would make sense to distribute it as a library, but currently it is not. There is a room for improvement for being more user-friendly (for instance, loading operator precedence table from a file, ...). The code quality seems to vary, as well as the code formatting.

The testing approach is not very sophisticated. The author provides only a main.cpp file with a few asserts, some of which are commented out, leaving me wondering why. Since this project is meant to test both a general expressions grammar parser and a Pascal language parser, I would expect dozens of tests.

The directory hierarchy is confusing, with each class having its own directory. This seems excessive for a project with only seven .cpp files. The use of CMake is very basic; I would have expected at least CTest integration to automatically run tests. According to the attached .git directory, the author started working very late in the semester. The git commit messages lack details about the work done in each commit ("Some errors are fixed, testing continues"—what errors? How and where did they appear? "Currently doing some things with atomic operations"—what things?), making it difficult to understand the motivation behind each commit and might make future contributions to the project more difficult.

Additionally, the thesis attachment includes not only the source code but also object files and IDE settings from the author's development machine, which should not have been distributed.

4. Evaluation of results, publication outputs and awards

55 /100 (E)

I am uncertain if the results could be deployed in practice. The code does not seem ready for production use, and no current use cases are apparent. Additionally, the code quality does not meet the standards for educational use.

Had the software for generating expression parsers be more configurable and more user-friendly I think there would be a merit.

The overall evaluation

52 /100 (E)

I think the thesis needs a little bit more time to be fulfilled better. The topic is indeed interesting and it is unfortunate that the author did not invest more time to look into it. I evaluate the thesis with grade E and 52 points.

Please note the comment regarding section 2.9 in part 2 of the review concerning the citation. I am not in a position to evaluate the severity of this issue.

Questions for the defense

1. Is there any literature on the same topics you were trying to solve? I.e., the general parsers for expressions and the possibility of expressing the whole language using the expressions?
2. Did you consider other languages than Pascal? Could they be expressed using general expressions parser?

Instructions

Fulfillment of the assignment

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

Main written part

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 52/2021, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

Non-written part, attachments

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

Evaluation of results, publication outputs and awards

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

The overall evaluation

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.