FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Supervisor's statement of a final thesis

Supervisor:	Ing. Jan Trávníček, Ph.D.
Student:	Jan Chybík
Thesis title:	Dataflow analysis in Azure Data Factory
Branch / specialization:	Computer Science
Created on:	8 June 2022

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

The thesis goals were a) to analyze the Azure Data Factory tool, more specifically, how it specifies dataflow, and to analyze Manta as a tool that aggregates dataflow from multiple systems in the form of a complete lineage across a data warehouse, b) to design ways to automatically retrieve this dataflow, and c) to implement a proof of concept implementation that is going to retrieve the dataflow specified by Azure Data Factory configurations/scripts. All those requirements were fulfilled.

2. Main written part

82 /100 (B)

The text is written in the English language on a good to a very good level. There are occasional typos or grammar incorrectnesses but not in an amount that would prevent understanding the text. The text could be a bit more formal in some places.

Factual issues:
- "A directed edge (u, v ∈ E)" seems incorrect, should be "A directed edge (u, v) ∈ E".
- I'm missing the definition of union, intersection, and complement of a language.
- In the definition of derivation in multiple steps, there should be α0, α1, α2, . . . , αk; α0 is missing in the text.
- "(q, ε) is called accepting configuration of automaton M iff q ∈ Q"; should be q ∈ F.
- There is a definition of a parse tree and abstract syntax tree, however, there is no relation between the two specified.
- "LL(k) stands for the number of look-ahead characters"; should be look-ahead tokens or lexemes. At this point, the lexer has already processed the input.
- In the context of LL(k): "The parse table for grammar G = (N, Σ, P, S) is a mapping M : (N ×

(\Sigma ^ k ∪ {ε})) ⟶ n"; actually it is (N × \Sigma ^ l) for all l <= k, close to the end of the analyzed sentence there simply is not enough tokens/lexemes to provide all k.
- "This work will further focus only on ANTLR because it is used to implement a proof of concept scanner." There is no reasoning behind this decision. Later in the text, there are mentions of someone else starting the implementation, which would be a valid reason to use ANTLR3, another could be that it is the strongest LL analyzer with, for instance, simpler error recovery and reporting of syntax errors than LR analyzers.

Typography issues:
- a multitude of missing articles,
- some character swaps: ANTRL, "... it means that ,for instance, filter ...",
- sometimes a singular word form is used instead of a plural, and vice versa,
- citation references are sometimes at the very end but within a sentence, sometimes right after the sentence,
- azure data factory vs. Azure Data Factory, manta vs. Manta,
- there are some sentence duplications, primarily in Chapter 1.

3. Non-written part, attachments 82 /100 (B)

The attachment has the form of a Java code (mostly) and grammar definition files of ANTLR. The code is well written and it satisfies the criteria to be considered a proof of concept.

Nevertheless, there are some issues with the code, most of which don't have any impact on the functionality:
- The parser has issues with a unary minus; the in-parser definitions of AST could be a bit simpler in some places (minor); the parser compilation reported a few warnings which didn't have any effect on the resulting parser.
- The AST interfaces could be simpler as some methods are not needed in artefacts that actually use the AST nodes.
- I would like to see the interesting transformation attributes explicitly represented in AST.
- Processing of AstParameter in cooperation with DataFlowScript could be simplified if interfaces are redesigned.
- Name normalization implemented in the resolver could statically enforce that normalization actually happened and happened exactly once.
- At some places, there are notes like "TODO check if this works".
- The dataflow generator does not generate filter flows.

Positives:
- I like the use of Optional to avoid naked nulls representing that a value is not present, but I would recommend using it more broadly, at least with all AST retrieval methods.

4. Evaluation of results, publication outputs and awards 85 /100 (B)

The code is tested with unit and integration tests which cover a substantial part of the implementation. I would still like to see more of them. Also, especially in the dataflow generator, the tests could be more focused on just a single (or a handful of) thing(s). That way they would be more understandable and their expected results smaller.

On the other hand, the code is already a part of the regular release of Manta and as such, the testers team has already verified the functionality of the code.

5. Activity of the student

▸ [1] **excellent activity**
[2] very good activity
[3] average activity
[4] weaker, but still sufficient activity
[5] insufficient activity

The student actively and diligently worked on the assignment.

6. Self-reliance of the student

▸ [1] **excellent self-reliance**
[2] very good self-reliance
[3] average self-reliance
[4] weaker, but still sufficient self-reliance
[5] insufficient self-reliance

We had a few consultations but besides those, the student was able to solely design and implement most of the code.

The overall evaluation 83 /100 (B)

Overall, the text of the thesis would benefit from another round of proofreading. Most of the issues are in the general introduction section like in definitions. The implementation, in the form of the attachment, is working and is able to extract important data flow from the provided scripts, still, there are some constructs and design choices that I find not ideal and that I would like to see fixed in the future. Nevertheless, the result is a solid proof of concept implementation and that counts. All in all the quality of the thesis and the code is high, thus, I recommend the thesis for defence and I recommend evaluating it with 82 points, i.e grade B (very good).

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.

Activity of the student

From your experience with the course of the work on the thesis and its outcome, review the student's activity while working on the thesis, his/her punctuality when meeting the deadlines and whether he/she consulted you as he/she went along and also, whether he/she was well prepared for these consultations.

Self-reliance of the student

From your experience with the course of the work on the thesis and its outcome, assess the student's ability to develop independent creative work.

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.