Research Report INGI 2019

Administrative part

- Researcher name: PHAM Hoang Son
- Date of first research contract in INGI: 5/2/2018
- List of past contracts in INGI (if any): none
- Research advisor name(s): Prof. Kim Mens, Prof. Siegfried Nijssen
- Current funding source: Brussels Region (INTiMALs project)
- Research topic: to research and deploy novel pattern mining algorithms to discover structural regularities in source code repositories.
- PhD admission: not applicable (already postdoc)
- PhD confirmation: not applicable (already postdoc)

Teaching tasks in 2018-2019

- List of courses (1st, 2nd semester): assisted Prof. Siegfried Nijssen in teaching the course of Databases (LINGI2172) in the 2nd semester.
- Estimated time spent on the teaching tasks (in percentage of full-time, averaged over 12 months): approximately 20% of my working time (in the period of teaching time). This amount of time includes practice section (2 hours per week), exercise correction and setting up exercises in the Inginious system.
- Comments, questions and issues related to your teaching tasks: none

Research activities in 2018-2019

- Topic: we focus on researching and deploying novel pattern mining algorithms to source code repositories. In particular, in the first stage of the project, we focused on mining relevant patterns, in software source code, such as programming idioms, coding conventions. These patterns can be used to provide useful recommendations to software developers on how to improve and modernize legacy software systems. In the second stage of the project, we applied frequent graph mining algorithms to discover frequent library usage in software source code. These usage patterns are invaluable in understanding how the API is typically used by developers and help highlight anomalous usage.
- Results of the first stage of the project:
  - In collaboration with other members at VUB and RainCode we explored a framework for discovering syntactic code idioms in source code repositories.
  - We proposed a novel algorithm to mine regularities in software source code. The new algorithm is a constraint-based tree mining algorithm, specifically designed for the analysis of software repositories. It combines two ideas: (i) maximal frequent subtree mining to ensure that a condensed representation of only large patterns is found, (ii) constraint-based data mining, in which additional constraints are imposed on the patterns to be found. Our approach is based on the addition of a number of novel constraints to the FREQT algorithm, combined with a new approach to find maximal subtrees. In collaboration with software engineers we analyzed in detail the quality of the patterns found. The results show (i) a significant reduction of the execution time and number of discovered patterns with respect to the original FREQT algorithm, (ii) that many of the discovered patterns

highlight relevant code regularities, (iii) that some of the patterns found are significantly larger than the simpler coding idioms found in earlier studies.

- Results of the second stage of the project: In this stage we used SUBDUE, a graph mining algorithm, to discover library usage patterns in software source code. SUBDUE is an algorithm that identified a set of patterns based on the MDL principle. Our preliminary experimental results show that SUBDUE was not very scalable to mine usage patterns in source code. Because the size of graphs produced from software source code is often large the cost of computing subgraph isomorphisms is hight.
- Possible difficulties:
  - The proposed algorithm can be applied to search syntactic patterns in source code repositories; however, various constraints have to be configured to find interesting patterns. In practice, choosing appropriate constraints to generate interesting patterns is a difficult task since it requires the users to have a good understanding on the analysis data. Thus, how to efficiently search and evaluate the interesting patterns are the challenging tasks of our current research. Another challenging task is how to efficiently mine library usage patterns in a large corpus.
- Perspectives for 2019-2020:
  - Continue improving our new algorithm for software source code mining.
  - Conduct more empirical analysis to evaluate the efficiency of the new algorithm.
  - Continue researching on graph mining algorithms to mine usage patterns in source code.
  - Write papers for the current research.
- Comments, questions or issues related to your research: none

Publications and research reports (since January 1, 2018):

- Conference papers:
  - A Language-Parametric Modular Framework for Mining Idiomatic. In Pre-proceedings of the 12th Seminar on Advanced Techniques and Tools for Software Evolution (SATToSE), 2019 (accepted but not yet published).
  - Mining Patterns in Source Code using Tree Mining Algorithms. In Proceedings of the 22nd International Conference on Discovery Science (DS), Springer, 2019 (accepted but not yet published).
  - Statistically Significant Discriminative Patterns Searching. In Proceedings of the 21st International Conference on Big Data Analytics and Knowledge Discovery, Springer, 2019 (accepted but not yet published).

- Demo track:
  - A Language-Parametric Toolchain for Mining Idiomatic Code Patterns. Demo in ‹Programming› 2019 Demos Track, 2019 (online).

Regarding future publications, our plain is to conduct more empirical analysis on a large number of software source code repositories to write a journal paper that discusses the specific pattern mining approach for source code repositories. In addition, we plan to write a conference paper that discusses specific techniques used to improve performance of tree mining algorithm for mining regularities in software source code.