INGI research Report 2018

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 2017-2018

- 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 2017-2018

- Topic: To research and deploy novel pattern mining algorithms to source code repositories. In particular, we focus on mining relevant patterns, in software source code, such as programming idioms, coding conventions and library usage protocols. These patterns can be used to provide useful recommendations to software developers on how to improve and modernise legacy software systems.
- Results:
  - At the first stage of research we collaborated with other members of the project at VUB and RainCode to explore a framework for discovering syntactic code idioms in source code repositories.
  - In this period of time our main research was to explore and improve available frequent tree mining techniques for discovering programming idioms. In particular, we are applying Freqt to mine frequent subtrees in software source code. Freqt is an efficient algorithm for discovering frequent subtrees in labeled ordered tree data. It is easy to use and to improve. According to various experiments, it is shown that Freqt can be used for programming idiom mining, but it has some limitations such as being highly time consuming, generating a large amount of patterns and redundant patterns. To tackle these problems various strategies are being exploited to improve the Freqt algorithm.
    - Strategies to reduce the number of output patterns: filtering the output patterns based on size constraints such as size of pattern (i.e. minimum and maximum

number of nodes in a pattern) and size of leaf nodes (number of leaf nodes in a pattern). In addition, a post-processing is also used to only output maximal patterns.

- Strategies to prune the search space: with the use of size constraints the number of patterns decreases, but it is still high for analysis. In addition, these patterns are not very interesting according to software developers' expectation. To reduce the execution time, the number of output patterns and to discover patterns which are more interesting domain knowledge is integrated into the mining process. In particular, different constraints which based on programming language definitions are proposed to drive the pattern searching. For example, the constraints based on the definitions of ordered or unordered node, obligatory or optional child; the constraints based on user-defined rules such as blacklist children, root labels; and other constraints based on leaf nodes in a subtree . As a result, with the use of these constraints the execution time of Freqt significantly reduces and the number of discovered patterns is limited. In addition, many discovered patterns are useful to explain a structure of a program source code.

- Possible difficulties:
  - Currently, the proposed method can be applied to search syntactic code idioms in source code repositories; however, it has to use various constraints 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.
- Perspectives for 2018-2019:
  - Continue exploring new pattern mining techniques for software source code mining.
  - Focus on applying statistical techniques to evaluate the interestingness of the patterns.
  - Write a paper for the current research.
- Comments, questions or issues related to your research: none

Publications and research reports (since January 1, 2017):

- We are going to submit a paper that discusses the discriminative pattern mining algorithm for genomic data. This paper is the result of my PhD.
- We are proposed to write the first research report for the current project.
- We plan to submit a joint paper about the project in this autumn, and a paper that discusses the specific pattern mining approach for source code repositories in the winter or spring.