Assisting Users for Repetitive Actions: Pattern Mining Meets Program Synthesis

Peggy Cellier, IRISA - SemLIS, INSA Rennes, peggy.cellier@irisa.fr
Alexandre Termier, IRISA - Lacodam, Univ. de Rennes 1 Alexandre.Termier@irisa.fr

Location: IRISA, team Lacodam/SemLIS, Rennes

Keywords: data science, data mining, pattern mining, program synthesis

Context and description of the internship:

Refactoring of structured or partially structured documents (e.g., program codes, markdown notes) is a tedious and time consuming task. It goes beyond a mere syntactic correction, and affects the structure of the document. An example of a refactoring task in markdown notes is the transformation of a sequence of headers into a bullet list. Usually this task requires making manually the same kind of changes at many places of the document. There is a risk of error if the author forgets one or more occurrences, or makes a typing mistake in some of them. To address this issue, program editors have some "pre-built" refactoring recipes, but they only apply to well-defined situations, and do not consider documents other than program codes.

An exciting development in research is the field of *Program Synthesis* [GPS17] and especially *Programming by Example* (PBE). The idea is to observe a human user performing a repetitive task, and generate automatically a program than can perform this task instead of the user on all the remaining occurrences. One of the most well-known program synthesis approach is FlashFill [Gu11], that is available in Excel. An example of a FlashFill use case is the automatic filling of a cell from the values of other cells. For instance, the user creates a new column named "Full name" and then manually fills the cells of the new column by concatenating the strings of two input columns: "firstname" and "lastname". In that setting, after a few examples FlashFill will detect the concatenation operation and propose the user to fill all the remaining rows with the same operation. Recently, the research group which developed Flashfill, proposed BluePencil [MGL+19], a PBE approach that detects on the fly user-specific refactorings in any kinds of structured documents, and automates them. BluePencil is available as an option in Visual Studio Code.

The goal of this internship is to analyze and improve the BluePencil approach. In their paper, the authors show that the main issue is the generation of false positives, i.e. wrong refactorings. Such false positives often originate from the noise of the data: the users are making many edits all over the document, and it may be difficult to precisely isolate which parts of these edits are relevant to the refactoring. Techniques from data mining can help here: pattern mining methods [AH14] are designed to discover possibly complex repetitions in data while resisting to noise. The shortcoming of pattern mining approaches is that they may output a huge number of mostly redundant results, and that they can require a large amount of computation time (which would not be compatible with an "on the fly" behavior). The challenge of the internship is thus to determine what pattern mining approaches would be suited to reduce the number false positives of BluePencil, and then how should these approaches be adapted in order to be used successfully in BluePencil.

The internship work will be organized as follows:

- Literature review on program synthesis and pattern mining;

- Experiments with BluePencil in order to characterize the reasons for false positive;
- From such characterization of false positives, proposition of a pattern method adapted for usage as the BluePencil repetition inference engine;
- Experiments to validate the interest of the proposed contribution.

The candidate should have a strong interest for algorithms, programming and possibly data science. This internship can lead to a PhD proposition.

References

[AH14] Charu C. Aggarwal, Jiawei Han*: Frequent Pattern Mining*. Springer 2014, ISBN 978-3-319-07820-5

[Gu11] Sumit Gulwani: *Automating string processing in spreadsheets using input-output examples*. POPL 2011: 317-330

[GPS17] Sumit Gulwani, Oleksandr Polozov, Rishabh Singh: *Program Synthesis*. Found. Trends Program. Lang. 4(1-2): 1-119 (2017)

[MGL+19] Anders Miltner, Sumit Gulwani, Vu Le, Alan Leung, Arjun Radhakrishna, Gustavo Soares, Ashish Tiwari, Abhishek Udupa: *On the fly synthesis of edit suggestions*. Proc. ACM Program. Lang. 3(OOPSLA): 143:1-143:29 (2019)