CS517-Project Proposal

Parisa S. Ataei

ACM Reference Format:
Parisa S. Ataei. 2020. CS517-Project Proposal. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/nnnnnnn.nnnnnnn

First, familiarize yourself with the project guidelines. Then submit a response here with the following information. Please, no more than a few sentences for each question (you will have a chance to write a whole document about this):

(1) What is the NP-hard problem that you will focus on? (input, output, any terminology needed to understand it).
Assume we have the syntax of a query language, given in Figure 1. The problem is: given the program p in this language is there a (smaller) program p' that is semantically equivalent to p?
I want to attempt to solve this using the syntax-driven induction of program p and see if a variational point can be omitted without changing the semantics of the program.

(2) What is the relevance of this problem to an area of CS research? It's quite OK if the relevance is extremely narrow.
Program equivalence is one of the most important problems in formal verification and thus, in CS.

(3) Are there any references (academic articles) that you know will be relevant for writing the project writeup?
Yes. I have to look into the research for program equivalence, however, lots of research has been done on variational programming and variational data which I will be using [4–6, 6]. The provided query language is a simpler version of the language I designed for my ongoing research project [2, 3].

(4) What is your rough plan for implementing the software tool? Please briefly discuss input/output formats, choice of programming language, and any relevant libraries that you might use.
I am planning to reduce the AST of a program to a SAT problem s.t. if it is satisfiable then there exists a (smaller) equivalent program.
The input is a program written in the given language and the output is yes/no.
I am using Haskell and the SBV library for solving the SAT formula.

(5) You may do the project alone or in groups of 2. If you are in a group, what is your plan for collaboration and division of labor?
I will be doing the project on my own.

$$
\begin{array}{rcll}
e \in \mathbf{E} & ::= & \texttt{true} \mid \texttt{false} \mid f \mid \neg f \mid e \wedge e \mid e \vee e & \\
\theta \in \mathbf{\Theta} & ::= & \texttt{true} \mid \texttt{false} \mid a \bullet k \mid a \bullet a \mid \neg\theta \mid \theta \vee \theta & \\
 & \mid & \theta \wedge \theta \mid e\langle\theta, \theta\rangle & \\
A \in \mathbf{A} & ::= & a \mid e\langle a, a\rangle \mid a, A \mid e\langle a, a\rangle, A & \\
q \in \mathbf{Q} & ::= & r & \textit{Relation reference} \\
 & \mid & \pi_A q & \textit{Projection} \\
 & \mid & \sigma_\theta q & \textit{Selection} \\
 & \mid & q \times q & \textit{Join} \\
 & \mid & e\langle q, q\rangle & \textit{Choice} \\
 & \mid & \varepsilon & \textit{Empty relation}
\end{array}
$$

Figure 1: Syntax of variational relational algebra, where $\bullet$ ranges over comparison operators ($<, \leq, =, \neq, >, \geq$), k over cosntant values, a over attribute names, and A over lists of variational attributes. The syntactic category e represents feature expressions, θ is variational conditions, and q is variational relational algebra terms.

REFERENCES

[2] Parisa Ataei, Arash Termehchy, and Eric Walkingshaw. 2017. Variational Databases. In *Int. Symp. on Database Programming Languages (DBPL).* ACM, 11:1–11:4.
[3] Parisa Ataei, Arash Termehchy, and Eric Walkingshaw. 2018. Managing Structurally Heterogeneous Databases in Software Product Lines. In *VLDB Workshop: Polystores and Other Systems for Heterogeneous Data (Poly).*
[4] Martin Erwig, Eric Walkingshaw, and Sheng Chen. 2013. An Abstract Representation of Variational Graphs. In *Int. Work. on Feature-Oriented Software Development (FOSD).* ACM, 25–32.
[5] Meng Meng, Jens Meinicke, Chu-Pan Wong, Eric Walkingshaw, and Christian Kästner. 2017. A Choice of Variational Stacks: Exploring Variational Data Structures. In *Int. Work. on Variability Modelling of Software-Intensive Systems (VaMoS).* ACM, 28–35.
[6] Eric Walkingshaw, Christian Kästner, Martin Erwig, Sven Apel, and Eric Bodden. 2014. Variational Data Structures: Exploring Trade-Offs in Computing with Variability. In *ACM SIGPLAN Symp. on New Ideas in Programming and Reflections on Software (Onward!).* 213–226.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn