HUGH KAUL PRECISION MEDICINE INSTITUTE
The University of Alabama at Birmingham

REVIEW

Of the member of the dissertation council for the dissertation of Peter Lozov on the topic: "Automated Synthesis and Efficient Execution of Relational Programs", submitted for the degree of candidate sciences in a scientific speciality 2.3.5 "Mathematical and software support for computers, complexes and computer networks"

The work presented in this dissertation is motivated by the desire to advance the theory and practice of relational programming. In relational programming, programs model mathematical relations; this can be contrasted with functional programming, in which programs model mathematical functions. Relational programming is appealing for several reasons—for example, relations naturally describe inference rules from standard formal semantics. Relations are also more flexible than corresponding functional programs—arguments to a relation can be replaced with logic variables representing unknown values, which an underlying logic engine will attempt to instantiate through a combination of constraint solving (usually involving unification) and search. The resulting relational programming system can run relations in multiple "directions"—forwards, backwards (function inversion), and in other ways that generalize the original function.

Unfortunately, this flexibility comes at a cost when compared to functional programming. First, it can be difficult, time consuming, and error prone to translate by hand a function into its relational equivalent. Second, without careful hand-optimizations, the resulting translated relation is often too slow to be useful, and may even diverge when called with the same arguments as its functional equivalent. Even worse, since a relation can be called in different "modes", with fresh logic variables in various argument positions, there *may not exist* an ideal ordering of the conjuncts within the translated relation. Together, these problems make relational programming challenging even for experts, and inaccessible to most programmers. The work presented in this dissertation addresses *all* of the above critical problems in relational programming.

The dissertation describes relational programming, and the problem with conjunction being unfair in standard implementations of miniKanren search. The author presents syntax, typing rules, and semantics for both functional programming in OCaml and for relational programming in OCanren. The dissertation presents both denotational and operational semantics, and extends the semantics to angelic/non-deterministic semantics to describe the relational case. The author describes a method for translating well-typed OCaml functions into equivalent OCanren relation, and proves the correctness of this conversion algorithm. The dissertation also shows how dynamic selection of unnesting/conjunct evaluation can be used to optimize the resulting translated relations. Finally, the dissertation presents benchmarks showing that the optimized, translated relations are as efficient as the same relations hand-written by an expert OCanren programmer.

The main results of this dissertation are:

1. A method for converting typed general functional programs to their relational equivalent.
2. Definitions and proofs of static and dynamic correctness of the relational conversion method.
3. A miniKanren semantics that includes a procedure for dynamic control of conjunction ordering.
4. A proof of the fairness of conjunction fairness in the described semantics.

The work in this dissertation makes several important contributions to the theory and practice of relational programming. First, the dissertation describes a practical transformation from functional programs to miniKanren relations. While the dissertation uses OCaml and OCanren as the functional and relational languages, the transformation approach is general, and can be applied to other functional and relational languages. This transformation approach reduces the difficulty of translating a function into a miniKanren relation, which can be time consuming and error prone, especially for beginning relational programmers. The approach also makes it easy to take advantage of the large number of existing functional programs that already exist.

This work proves that the functional-to-relational program transformation preserves correctness. This correctness guarantee is critical, since the transformed relation may be difficult to reason about fully, especially for a beginning relational programmer. Without such a guarantee, a cautious programmer would not dare use the approach given in the dissertation. Additionally, the semantics, proofs, and proof techniques will be useful for future research on miniKanren.

Finally, the work demonstrates that the performance of the automatically transformed relational programs is equivalent to that of their equivalent hand-written relations. Once again, without this evidence, a cautious relational programmer would not be able to trust the approach and system described in this dissertation.

In conclusion, the work presented in this dissertation is an important step towards making OCanren—the first implementation of miniKanren that takes full advantage of the Hindley-Milner type system—usable for relational programming by non-specialists. The described techniques should be especially helpful for those learning miniKanren and relational programming. Even expert miniKanren programmers will benefit from the ease, speed, and correctness of the functional-to-relational transformation. Furthermore, these techniques can, and should, be ported to other implementations of miniKanren, where they should also prove useful.

There are several ways in which the dissertation could be improved:

1. The dissertation could better describe which classes of miniKanren programs can or cannot be used with the presented approach. For example, the most interesting miniKanren programs are relational interpreters and type inferencers. My understanding is that these programs will not work with the well quasi-ordering approach presented in the dissertation. What are examples of programs that *do not* work with this approach? How can a miniKanren programmer easily tell which programs will work with the proposed approach?
2. Jason Hemann has also done work on miniKanren semantics and on transforming functional (Scheme) programs into miniKanren relations. How does his work compare with the approach presented in this dissertation? Also, is there any relevant related work from the Prolog/Mercury/Curry/logic programming literature?
3. In which ways might this work be extended or improved upon in the future? What are the major open problems that remain, or that have been created by the proposed approach?
4. The correctness proof for the functional-to-relational program transformation only proves correctness for when the “input” arguments are all ground, and the “output” argument is a fresh logic variable. Could the proof of correctness be extended to other modes as well, at least in the cases where the “input” arguments *become* ground during the computation?

5. The definition of *appendo* on pages 52—53 appears incorrect. The arguments to the recursive call should be *xs y xys* rather than *t y ty*. Also, I think the recursive call must come earlier in the conjunction to produce the divergence behavior described in the text, if I understand the argument correctly.
6. Definition 9 on page 49 assumes the existence of a semantic variable *q_m*, but the following text only refers to an unintroduced semantic variable *s_m*, which I assume is a typo.
7. Several times mathematical notation is explained long after it is used.
8. The dissertation contains several paragraphs, mathematical descriptions, figure numbers, and bibliography entries in Russian rather than in English. I assume this is unintentional.
9. The dissertation contains minor English grammar, punctuation, and spelling mistakes, such as missing articles and missing spaces. New technical terms should always be in *italic* when first introduced. On page 6 ‘Bird’ should be spelled ‘Byrd’. Several times quoted text has mismatched quotes: ‘foo” or “foo’. On page 73 ‘semitics’ should be ‘semantics’.

The dissertation “Automated Synthesis and Efficient Execution of Relational Programs” meets the basic requirements established by Order No.11181/1 dd. 19.11.2021 “On the procedure for awarding academic degrees at St. Petersburg State University”. The applicant Peter Lozov deserves to be awarded the academic degree of Candidate of Sciences in a scientific speciality 2.3.5 “Mathematical and software support for computers, complexes and computer networks”. No violations of paragraphs 9 and 11 of the specified Order have been detected.

Member of the Dissertation Council,
Scientist, Doctor of Philosophy in Computer Science,
Hugh Kaul Precision Medicine Institute,
Heersink School of Medicine,
University of Alabama at Birmingham,
Birmingham, Alabama, USA

William E. Byrd

Date: October 16, 2022