THE NEXT 700 FAILED STEP-INDEX-FREE LOGICAL RELATIONS

DANIEL GRATZER

ABSTRACT. An important question in programming languages is the study of program equivalence. This is done typically with the construction of a relation denotational model or a syntactic analogue, called a logical relation. Logical relations have been shown to be an effective tool for analyzing properties and lending formal weight to ideas like data abstraction and information hiding. A central difficulty with logical relations is their fragility; it has proven to be a difficult challenge to scale logical relations to more realistic languages. In this thesis, we discuss a number of methods for the construction of a logical relation for a language with general references. None of these methods are sufficient to construct a logical relation without resorting to step-indexing, demonstrating the difficulty of expressing the recursive structure of higher-order references.

In the study of programming languages a great number of important questions can be reduced to questions about the equality of programs. The verification of a compiler pass is nothing but a question of equality of the naive program and its optimized version, an optimized data structure may be shown to be correct relative to a much simpler reference solution, a type theory hinges on the construction of definitional equality and so on. Given the role that equality plays in programming languages, a great variety of mathematical tools have been developed to analyze the various notions of equality possible in a programming language.

Broadly speaking, there are two classes of these tools. One may consider denotational approaches, where in the equality of programs is reduced to the question of the equality of normal mathematical objects. This line of study begins with Dana Scott's investigations of the lambda calculus [**?**]. These tools have been immensely effective when they may be applied but they are difficult to use. Complex programming languages often make use of extremely sophisticated mathematical objects and using the model requires understanding them. This has meant that denotational semantics is traditionally out of reach for the verification of programs by an average programmer or anyone besides a domain expert.

On the other hand, there are syntactic approaches to equality. Some of these date back to the original study of the lambda calculus and its reduction properties. Syntactic tools tend to be simpler to use, relying only one elementary mathematics and an understanding of the syntax itself. For equality, the tool of choice when working syntactically is a logical relation [**?**]. Dating back to **?** logical relations crucially define a type-indexed notion of equality. Logical relations have proven important for validating certain obviously desirable properties such as function extensionality. They have also given rise to a powerful theory of data abstraction called parametricity [**?**].

Date: April 8, 2018.

1. An ML-like Language with General References

2. A Step-Indexed Logical Relation

3. Tying the Knot Using Domains

4. Understanding State through Guarded Recursion

4.1. **A Logical Relation in Ultrametric Spaces.**

5. Conclusions

References