MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Prof. Dr. Derek Dreyer
Max Planck Institute for Software Systems
Campus E1.5, 66123 Saarbrücken, Germany
Tel: +49 681 9303 8701
E-mail: dreyer@mpi-sws.org

August 30, 2020

Review of Piotr Polesiuk's doctoral dissertation

To whom it may concern:

Piotr Polesiuk's dissertation presents a significant number of interesting contributions to the semantics of delimited continuations and algebraic effects, with an emphasis on developing sound (and in some cases complete) methods for relational reasoning about programs. The thesis comprises 9 papers published in prestigious and highly competitive conferences and journals, including 3 papers in the top international programming languages conference POPL and 3 in the top logic and semantics journal LMCS. Although much of this work is collaborative with several authors, it is in any case an impressive publication record for a graduate student. Moreover, the thesis is notable for the wide array of problems considered and techniques developed, many of which require highly specialized expertise but are nonetheless presented clearly. It feels like several dissertations in one!

Summary of the dissertation:

The contributions of the thesis are divided in two parts.

The first part (papers [U1]-[U4]) concerns the development of *bisimulations* for untyped languages with *delimited continuations* and state. Delimited continuations are a very old topic of study in programming language semantics. However, they are notoriously tricky to reason about because they provide so much expressive power: since they allow the programmer to capture the continuation (the current program context) as well as to control how much of the continuation can be captured using delimiters, they can be used to encode a wide range of programming features, including mutable state, exceptions, and backtracking.

The second part (papers [T2]-[T5]) primarily concerns the development of *logical relations* and *type systems* for languages with *effect handlers*. ([T1] concerns logical relations and delimited control, but not effect handlers.) Effect handlers are in some sense a more user-friendly and easily programmable form of delimited continuations, where the code that has access to captured continuations is encapsulated neatly in the form of the effect handler. The study of their semantics,

as well as figuring out how best to integrate effect handlers into a practical programming language, is a highly active topic of current study.

The main novel contribution of Piotr's work in the first part of the thesis is in the development of *diacritical bisimulations*, which include two relations: one for *active transitions*, where the program takes at least one step, and one for *passive transitions*, where play an administrative role but do not correspond to an actual program step. This distinction between active and passive makes it possible to define a sound and complete bisimulations (both environmental and normal-form) for languages with delimited control and/or state that support extensional reasoning (*i.e.,* admit η-equivalence) and also support useful "up-to" techniques (which previous approaches did not). Piotr also shows how the theory of diacritical bisimulations generalizes to any complete lattice, and connects this technique with Pous' idea of a "companion" (a kind of largest up-to technique) to develop the notion of a diacritical companion.

At the heart of the second part of Piotr's thesis is the groundbreaking development of type systems and Kripke logical-relations models for languages with effect handlers, represented by three consecutive POPL papers [T2, T3, T4]. In [T2], Piotr studies a calculus with row-polymorphic algebraic effects based on prior effect-handler languages like Koka, and introduces the "lift" operation into this calculus in order to enable modular reasoning (in particular, parametricity). As far as I know, this is the first attempt to study parametricity and logical relations in the context of algebraic effects. In [T3], Piotr presents a type system for the novel features of *existential* and *local* effects, which serve to protect the user of an effect from its implementation details and vice versa, respectively. Piotr has realized this type system in an experimental language design (inspired by ML-style module systems) called Helium, although the thesis does not really provide details about that. In [T4], Piotr explores the problem of how to combine different uses of the *same* effect in a program (*e.g.,* to program with multiple mutable cells as opposed to a single one in the same context). He proposes to address this problem with the idea of *lexically-scoped effect instances* and explores two types of operational semantics for this language (open vs. generative). He also develops a novel Kripke logical relation for this calculus (equipped with the ability to reason about open terms), which he uses to prove equivalence of the open and generative semantics under some restrictions on type and effect polymorphism.

The second part of the thesis also presents some additional results concerning logical relations and type systems for delimited control and effect handlers. [T1] uses logical relations to establish coherence of type-directed, selective CPS-translation. [T5] builds on prior work by Forster *et al.* on establishing that delimited control operators and effect handlers can macro-express each other. Forster *et al.*'s work was done in an untyped setting, and Piotr generalizes this to the typed setting, showing that polymorphism enables type-preserving macro translation.

Assessment of the dissertation:

Spread across 9 papers on a range of different topics and utilizing a range of advanced techniques in programming language semantics and type systems, Piotr Polesiuk's dissertation represents an **outstanding** effort. Given how sprawling and expansive a dissertation this is, I cannot claim to

understand all of it, but I understood enough to see that Piotr is richly deserving of being awarded the PhD degree!

To my mind, the most significant part of the dissertation lies in the 3 POPL papers on type systems and logical relations for effect handlers. Developing effective language designs for algebraic effects (no pun intended) is a very hot topic, with papers published every year at conferences like POPL and ICFP. Although I am not an expert on effect handlers, I am very pleased to see work like Piotr's because it takes what you might call a "reasoning-centric" approach. Rather than just exploring the design space of effect handlers in terms of expressivity (a worthy criterion though that is), Piotr's work in this space is equally motivated by the concern of ensuring that the resulting language design enjoys useful reasoning principles (*e.g.,* parametricity) for establishing equivalences between programs. In much research on language design, reasoning about program equivalence has been done *post-hoc* because the field lacked sufficiently powerful methods for reasoning about features like continuations and higher-order state at the time when those features were introduced. But with the present exploration of the design space for algebraic effects, we are in a good position to explore the impact of different design choices on relational reasoning in "real time", so to speak. I am thus very happy to see Piotr's much-needed work on this topic, and I am happy to see it published regularly in POPL to ensure the widest visibility.

What's more, I am happy to see several of Piotr's papers ([T1, T2, T4]) mechanized formally in the Coq proof assistant. This is actually a non-trivial effort, given that Coq was not designed to support the types of proofs that Piotr is mechanizing. When Piotr began his work in this direction, he took inspiration from work I had previously done on the step-indexed LSLR logic, and mechanized a version of this logic in Coq in his IxFree library. (More recently, we have developed strong support for step-indexed logical relations in Coq using the Iris framework, but this was not available when Piotr started out.) I am extremely pleased to see Piotr formalizing his work on algebraic effects in IxFree: I view this kind of mechanization as essential for more rapid, stable, and trustworthy development of programming language metatheory, especially as our languages and semantic models grow ever more complex. I am curious whether these models could be mechanized in Iris.

The first part of Piotr's dissertation also represents a strong contribution. Developing any kind of relational reasoning models for languages with delimited control and state is a notoriously difficult problem, with a long and difficult literature. Compared to Piotr's work on logical relations and effect handlers, I would say that his work on bisimulations is more incremental and "technical" compared to prior work—*i.e.,* its practical relevance is less immediately clear—but it is still very interesting. As one who spent a number of years developing relational models (both logical relations and bisimulations) for languages with higher-order state, I was particularly intrigued by Piotr's paper on a complete normal-form bisimilarity for state [U3]. In that paper, he avoids some of the complexities of my work on transition-system-based models in favor of a more direct handling of certain anomalous examples (*e.g.,* the "deferred divergence" example). I'm not sure I completely grokked the proof of this example in the paper, but it appears simpler than other bisimulation proofs I have seen before. I would encourage Piotr to consider mechanizing his bisimulation-based methods as well as his logical-relations methods, as that would (a) breed confidence in the results, and (b) showcase the relative merits of the two styles of relational reasoning more clearly.

In summary, Piotr's Polesiuk represents an outstanding contribution in the area of relational reasoning and language design for languages with delimited control and algebraic effects. There is more than enough work presented here to fill multiple dissertations, but despite its somewhat sprawling content, there is nevertheless a strong degree of thematic unity, and the work itself clearly demonstrates Piotr's proficiency with the most advanced "tools of the trade". Furthermore, I expect that his work on type systems and logical relations for effect handlers, taking as it does a reasoning-centric perspective, will have impact on the direction of this rapidly evolving field of study. I therefore strongly recommend that Piotr be awarded the PhD with distinction.

Sincerely,

Prof. Dr. Derek Dreyer
Tenured Faculty
Max Planck Institute for Software Systems