Type-directed defunctionalization in the presence of type classes and monomorphization

Ayaz Hafiz
ayaz.hafiz.1@gmail.com

Abstract

We present the design and implementation of type-directed defunctionalization in Roc, a pure functional language with a continuation-based effect system and type classes. We have developed a compiler for Roc that performs monomorphization and specialization of higher-order functions over the function values they consume, via the lambda set specialization introduced in [1]. We describe our implementation effort to efficiently infer lambda sets, which can be modeled as anonymous sums. We also present our strategies to compile lambda sets with closure data to optimal runtime representations, which avoid boxing closure data except in the case of recursive closures. Finally, we present a single-pass algorithm we have developed to infer and specialize lambda sets in the presence of typeclasses compiled via type specialization. We believe this algorithm to be novel, and hope that it can inspire efforts to specialize higher-order functions over closure values in other languages.

ACM Reference Format:
Ayaz Hafiz. 2024. Type-directed defunctionalization in the presence of type classes and monomorphization. In *Proceedings of Workshop on the Implementation of Type Systems (Conference acronym 'XX)*. ACM, New York, NY, USA, 1 page. https://doi.org/XXXXXXX.XXXXXXX

References

[1] William Brandon, Benjamin Driscoll, Frank Dai, Wilson Berkow, and Mae Milano. 2023. Better Defunctionalization through Lambda Set Specialization. *Proc. ACM Program. Lang.* 7, PLDI, Article 146 (jun 2023), 24 pages. https://doi.org/10.1145/3591260

Conference acronym 'XX, January 20, 2024, London, United Kingdom

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX