Live Functional Programming with Typed Holes

Extended Version*

CYRUS OMAR, University of Chicago
IAN VOYSEY, Carnegie Mellon University
RAVI CHUGH, University of Chicago
MATTHEW A. HAMMER, University of Colorado Boulder

abstract stuff

CCS Concepts: • **Software and its engineering → Functional languages**;

Additional Key Words and Phrases: live programming, gradual typing, contextual modal type theory, typed holes, structured editing

*The original version of this article was published in the POPL 2019 edition of PACMPL [**?**]. This extended version includes an additional appendix.

Authors' addresses: Cyrus Omar, University of Chicago, comar@cs.uchicago.edu; Ian Voysey, Carnegie Mellon University, iev@cs.cmu.edu; Ravi Chugh, University of Chicago, rchugh@cs.uchicago.edu; Matthew A. Hammer, University of Colorado Boulder, matthew.hammer@colorado.edu.

1 INTRODUCTION

stuff

Paper Outline. outline stuff

2 RELATED AND FUTURE WORK

- OCaml debugger: https://caml.inria.fr/pub/docs/manual-ocaml/debugger.html
- Tolmach's debugger for SML: https://www.cs.tufts.edu/~nr/cs257/archive/andrew-tolmach/jfp95-debugger.pdf
- Recent work by Whitington & Ridge on stepping through OCaml programs: https://github.com/johnwhitington/ocamli
- DrRacket debugger: https://docs.racket-lang.org/drracket/debugger.html
- GHCi debugger: https://downloads.haskell.org/~ghc/7.0.1/docs/html/users_guide/ghci-debugger.html
- LambdaLab: https://www.cs.cornell.edu/~asampson/media/papers/lambdalab-splashe2018-preprint.pdf
- Elsa, lambda calculus evaluator: https://github.com/ucsd-progsys/elsa
- Look into Pyret debugger
- Hoed algorithmic debugger for Haskell and associated papers on computation tree tracing for lazy functional programs:
  - http://hackage.haskell.org/package/Hoed
  - https://kar.kent.ac.uk/49003/1/AlgDebugHaskellCostCentreStack.pdf
  - https://dl.acm.org/citation.cfm?id=2908104