

TypeSearch: Type-Directed API Search For All

Graduate



Marc Etter

Introduction: Software developers spend a lot of their time finding and composing pre-existing functions from various libraries. Almost all developers today use general-purpose search engines for this search. Specialized search engines such as Hoogle additionally use type information to improve this search, and have been successful for some typed functional programming languages. The options currently available for type-directed search for mainstream object-oriented languages is limited. Existing approaches for these languages do not have first-class support for subtyping or parametric polymorphism. The splitting and composition of a desired functionality into and from a number of pre-existing functions is also a task that needs to be done manually. In this Master's Thesis we present a proof-search-based approach to type-directed search with first-class support for subtyping, parametric polymorphism, splitting, and composition. The approach is language agnostic, and can be specialized to simultaneously support multiple typed object-oriented languages. Given that most mainstream languages fall under this category, this approach would extend the benefits of type-directed search to the majority of programmers. As a proof of concept, we provide a running implementation of the core language-agnostic approach and extend it to support the Java programming language. Further extensions would allow the tool to simultaneously support multiple programming languages using the same query syntax.

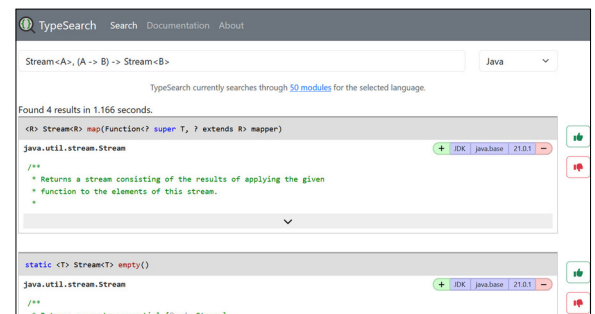
Approach: This Master's Thesis presents two chief contributions:

- A proof-search-based approach, using the Curry-Howard Correspondence, for finding API functions based on a user-supplied approximate type signature, with full support for parametric polymorphism and subtype polymorphism. We name this approach TSaPS (Type Search as Proof Search).
- TypeSearch, a proof-of-concept implementation of TSaPS. TypeSearch is designed with a modular plugin system so that support for new programming languages and different storage backends can be added easily. TypeSearch ships with a plugin implementation for Java, and a simple in-memory storage backend. TypeSearch is publicly accessible at <https://typesearch.dev>.

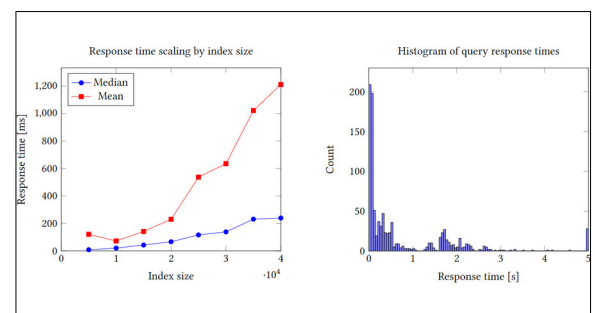
Conclusion: A type-directed API search and code synthesis approach based on the Curry-Howard Correspondence and proof search is certainly viable and flexible enough to be used for typed mainstream programming languages. With our proof-of-concept implementation, TypeSearch, we show that the approach performs well for medium to large sized libraries. We are looking forward to further

development of TypeSearch and are eager for feedback, which can be given on <https://typesearch.dev>.

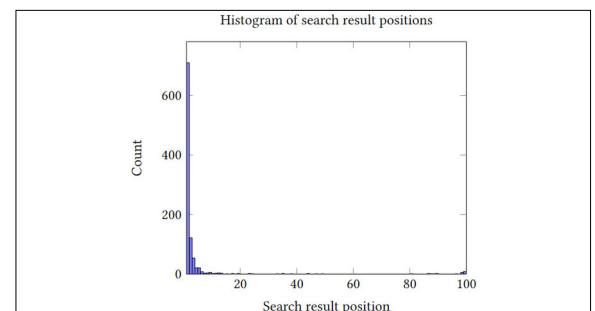
Screenshot of the TypeSearch UI at <https://typesearch.dev>. Own presentation



Response time measurements of TypeSearch. Histogram (right) is measured with index size 40 000. Own presentation



Histogram of search result ranking measurements for exactly matching type query. Own presentation



Advisor
Prof. Dr. Farhad D. Mehta

Co-Examiner
Joachim Breitner

Subject Area
Computer Science,
Software and Systems