

Pure Bondi with datatypes and patterns

Thomas Given-Wilson
University of Technology, Sydney
tgwilson@it.uts.edu.au

Abstract

The work presented here is an extension of the Bondi interpreter, for the typed pattern calculus, realized by Jay (2004). The current work focuses on three untyped calculi that explore the role of data structures and patterns.

1. The pure compound calculus, that is able to act on data structures in the manner of generalized algebraic data types. This is done through an extension of the lambda-calculus with special terms for the treatment of data structures.
2. The introduction of pattern matching with the pure static pattern calculus. The patterns are restricted to be any non-reducible term where all variables are binding.
3. The pure pattern calculus removes the restrictions on the patterns in the static calculus and allows any term to be a pattern. The treatment of patterns as “first-class citizens” allows for generic queries, such as the generic update.

The goals of this project are to improve the understanding of the calculi and to provide the foundations for future work. Implementing the algorithms and machinery of the theory aid in the understanding and testing of the theoretical underpinnings. The interpreters also provide a template for future compilers, including solving some of the potentially difficult problems in translating the theory into practice.

Jay, C. B. 2004, The pattern calculus, *ACM Transactions on Programming Languages and Systems (TOPLAS)* **26**(6), 911–937.