Generic and Indexed Programming (Project Paper)

Jeremy Gibbons, Meng Wang, and Bruno C. d. S. Oliveira

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD, UK
{jg,menw,bruno}@comlab.ox.ac.uk

Abstract

The EPSRC-funded *Generic and Indexed Programming* project will explore the interaction between *datatype-generic programming* (DGP) — programs parametrized by the shape of their data — and *indexed programming* (IP) — lightweight dependently-typed programming, with programs indexed by type-level representations of properties. Integrating these two notions will provide new ways for programmers to *capture abstractions*.

The project builds on insights from our recent work in DGP, which has investigated both *programming techniques* (including reasoning about generic programs, and using them to capture design patterns precisely), and *language mechanisms* (particularly lightweight approaches: patterns for simulating highly-expressive techniques in familiar but apparently less-expressive languages). Firstly, these lightweight techniques, which we have been embedding in Haskell's still relatively expressive type system, are in fact applicable to even less expressive but more *popular mainstream languages* such as Java and C#. Secondly, the techniques are more applicable than we first thought; in particular, they offer a solution to the so-called 'expression problem': safe combination of independent extensions along multiple dimensions. Thirdly, there is a synergy between DGP and IP: DGP makes IP more appealing, because the effort of stating properties can be amortized over more programs; IP provides a mechanism for DGP, because the indices can be representations of data's shape.