Universidade do Minho
Escola de Engenharia

Semana da Escola de Engenharia
October 24 - 27, 2011

BIDIRECTIONAL DATA TRANSFORMATION BY CALCULATION

Hugo Pacheco
Department of Informatics
E-mail: hpacheco@di.uminho.pt

KEYWORDS
Bidirectional Transformation, Data Refinement, Lenses, Program Calculation, Point-free Programming

ABSTRACT

Computing is full of situations where transforming a data format into a different one is essential to "bridge the gap" between technology layers, and these transformations are often desired to be bidirectional. The 2LT framework provides a bidirectional approach to the specification of general-purpose transformations between arbitrary data formats. This document outlines the main objectives and contributions of the under-development PhD thesis of the author, with the goal of extending the functionality of the 2LT in three ways: 1) propose a formal language and calculus for bidirectional lenses 2) discuss solutions for the alignment problem 3) implement the techniques into a renewed 2LT framework for the specification and optimization of two-level bidirectional lens transformations.

INTRODUCTION

Data transformation frameworks play an essential role in facilitating the sharing of information among software applications. Modifications to the data often break consistency constraints, and a key problem with most of these frameworks is the lack of proper mechanisms to synchronise them. Typically, we end up with *ad-hoc* solutions like manually specifying two unidirectional transformations so that they are coherent This introduces a severe maintenance problem: any change in one of the data models implies a redefinition of both transformations and a new coherence proof.

In response to this problem, intrinsic *bidirectional transformation* (Czarnecki et al. 2009) frameworks have become increasingly popular in various computer science domains. Most of these approaches encompass the design of domain-specific bidirectional languages in which one expression denotes a connected pair of transformations, whose consistency is guaranteed in the respective semantic space.

A formal solution to this problem is provided by the 2LT framework. This framework currently allows users to specify a transformation using a set of bidirectional *Haskell* combinators between a source format A and a target format B, getting for free forward and backward transformations between values of A and B Two unique features of this framework are that: it supports two-level transformations, i.e., transformations (such as drop all Birthday elements that appear inside the format) that are type-level functions executing over many possible formats; and performs automatic optimization of the underlying value transformations (that would search the whole, possibly large format for Birthday elements).

A well-behaved bidirectional transformation scenario is data *refinement* (Berdaguer et al. 2007), encompassing the transformation of a source (abstract) format into a target (concrete) format that contains more information. Only refinement transformations are currently supported by the 2LT framework. Another well-behaved scenario are *lenses* (Foster et al. 2007), that dually allow transformations that abstract some details from a source format. Lenses are more complex than refinements, since their backward transformation must recover the source information that was dropped by the forward transformation, for which there is no unique way.

GOALS AND CONTRIBUTIONS

We now outline the contributions of the thesis.

Language & Calculus

The first contribution is the design of a bidirectional lens language over algebraic data types (Pacheco and Cunha 2010), used in modern functional languages like *Haskell, ML* or *F#.* Following a natural embedding of our lens language in *Haskell*, the type system can then reason statically if a given lens program is type correct. To guarantee semantic correctness, we write each value-level transformation in a point-free style that unveils a powerful point-free calculus and allows proving bidirectional properties in a purely equational and often mechanizable way. Another contribution is an algebraic calculus for lenses, such that we can perform proofs about bidirectional programs as if regular functional programs (Pacheco and Cunha 2011). We establish that most of the standard point-free combinators can be lifted to lens combinators in our language and preserve their characterizing set of algebraic laws. As a result, we can show that optimization at the lens level subsumes the independent optimization of all the value- level transformations, opening interesting perspectives towards agile automatic lens optimization tools.

Alignment

A transversal issue to bidirectional programing languages is that of *alignment.* In lenses, the forward transformation might discard information, while the backward transformation shall recombine parts of the updated target with parts of the original source to produce an updated source. For reasonable behavior, lenses must be able to identify the modifications and establish correspondences between source and target models. Thus, we extend our lens language into a set of alignment-aware combinators that compute, propagate and use these correspondences alongside value-level transformations, and formalize this interaction and additional laws that guarantee that correspondences are propagated correctly using dependent type theory.

Implementation

The last contribution is the implementation of the above-mentioned lens language under a new version of 2LT framework providing strong, inferable types as well as two-level combinators for lenses. An example of an application scenario for such two-level programs is the automated abstraction of implementation details at particular positions of a specification (Cunha and Pacheco 2011). In particular, we lift the two-level combinator suite from refinements to lenses, together with a rewrite system for the automatic optimization of point-free programs (Pacheco and Cunha 2011).

CONCLUSIONS

The field of bidirectional programming is vast, with many competing bidirectional languages based on refinements, lenses and more general synchronization patterns. However, there exists no approach that we are aware of that tackles the issues of two-level transformation and optimization for lenses. As future work, we plan to integrate other transformation scenarios, and to consider data invariants to extend the sets of supported formats and transformations.

REFERENCES

P. Berdaguer, A. Cunha, H. Pacheco, and J. Visser. "Coupled schema transformation and data: Conversion for XML and SQL." In *Proceedings of the 9th International Symposium on Practical Aspects of Declarative Languages (PADL 2007),* volume 4354 of *Lecture Notes in Computer Science,* pages 290–304. Springer Berlin / Heidelberg, 2007.

A. Cunha and H. Pacheco. "Algebraic specialization of generic functions for recursive types." *Electronic Notes in Theoretical Computer Science,* 229:57–74, March 2011.

K. Czarnecki, J. N. Foster, Z. Hu, R. Lämmel, A. Schürr, and J. Terwilliger. "Bidirectional transformations: A cross-discipline perspective." In *Proceedings of the 2nd International Conference on Theory and Practice of Model Transformations (ICMT 2009),* volume 5563 of *Lecture Notes in Computer Science,* pages 260–283. Springer Berlin / Heidelberg, 2009.

J. N. Foster, M. B. Greenwald, J. T. Moore, B. C. Pierce, and A. Schmitt. "Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem." *ACM Transactions on Programming Languages and Systems (TOPLAS),* 29(3):17, 2007.

H. Pacheco and A. Cunha. "Generic Point-free Lenses." In *Proceedings of the 10th International Conference on Mathematics of Program Construction (MPC 2010),* volume 6120 of *Lecture Notes in Computer Science,* pages 331–352. Springer Berlin / Heidelberg, 2010.

H. Pacheco and A. Cunha. "Calculating with lenses: optimising bidirectional transformations." In *Proceedings of the 20th ACM SIGPLAN workshop on Partial evaluation and Program Manipulation (PEPM 2011),* pages 91–100, New York, NY, USA, 2011. ACM.

AUTHORS' BIOGRAPHY

HUGO PACHECO is a 4th year PhD student under the MAP-i Doctoral Programme in Computer Science, and a researcher of the CCTC group, in the area of Theory and Formal Methods. Previously, he has graduated in Licenciatura em Engenharia de Sistemas e Informática (LESI) at the University of Minho, in 2007. For more information, email him at hpacheco@di.uminho.pt or check his personal address at www.di.uminho.pt/~hpacheco.