

BIDIRECTIONAL DATA TRANSFORMATION BY CALCULATION

Hugo Pacheco*

Supervisor: Manuel Alcino Cunha

Co-supervisor: José Nuno Oliveira

University of Minho
School of Engineering
Department of Informatics / CCTC

Bidirectional Data Transformation

- Data transformations are frequent in software engineering and are essential to “bridge the gap” between technology layers. Moreover, users generally expect transformations to be bidirectional, in the sense that changes made to one model can be safely propagated to its connected pair.

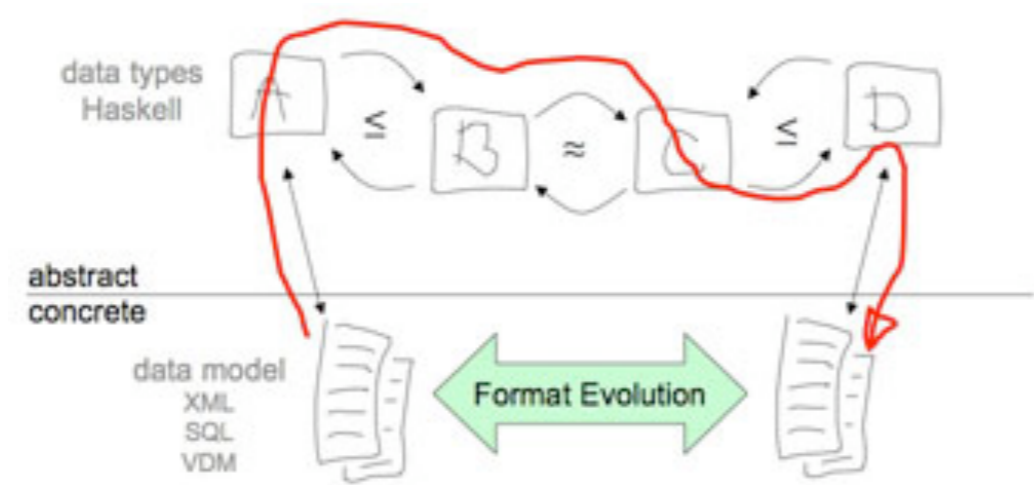
- The key issue in bidirectional transformations is what is meant by “safely propagated” and how to guarantee that property.

- 2 unidirectional transformations / bidirectional transformation



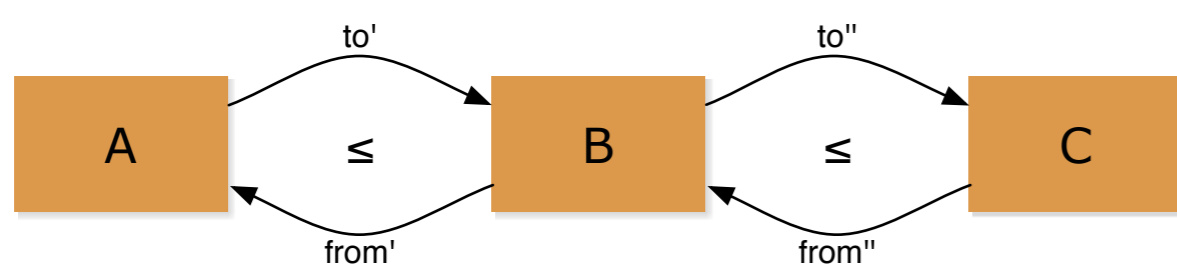
(manually prove consistency) (expensive, error-prone) (gen. reverse transformation) (strong behavioral laws)

2LT Framework

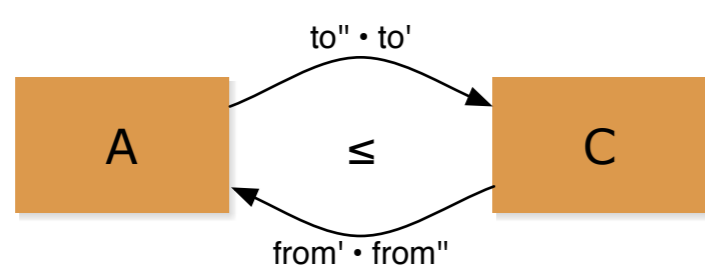


- <http://2lt.googlecode.com>

- Type-safe implementation of data refinements in Haskell such that transformations are well-typed, consistent and composable.



- Transformations are specified as point-free functions and subject to point-free simplification laws.



- An example application is the automatic mapping between XML schemas and SQL databases.

Results

- Alcino Cunha and Hugo Pacheco. Algebraic Specialization of Generic Functions for Recursive Types, MSFP 08.

Well-behaved Bidirectional Transformations

- Refinements

transform abstract models into more concrete ones, that have more information



- Lenses

transform concrete models into more abstract ones, that have less information



- Isomorphisms

transform models into equivalent ones, that have the same information



Goals / Open Questions

- Extend the 2LT framework to recursive types and investigate how to generically express and represent transformations over mutually recursive types.

- Investigate how the relational calculus, inherently more bidirectional than the currently expressed functional setting, can be mechanized and how that additional power will enhance the functionalities of the 2LT framework.

- Study how can lenses be integrated into the 2LT framework: which transformation scenarios exist and how to extend the lens theory to recursive types.

- Consider transformations that may add and delete information and, thus, are more general than refinements and lenses: which formal properties hold for such relations between models?



*hpacheco@di.uminho.pt