

Combining Syntactic and Semantic Bidirectionalization

Janis Voigtländer, Universität Bonn, jv@iai.uni-bonn.de

joint work with Zhenjiang Hu, Kazutaka Matsuda, and Meng Wang

Abstract

Matsuda et al. [1] (syntactically) and Voigtländer [2] (semantically) introduced two techniques that given a source-to-view function $get :: \tau_1 \rightarrow \tau_2$ provide an update propagation function $put :: \tau_1 \rightarrow \tau_2 \rightarrow \tau_1$ (partially) mapping an original source and an updated view back to an updated source, subject to standard consistency conditions, in particular: $put\ s\ (get\ s) = s$ and if $put\ s\ v'$ is defined, $get\ (put\ s\ v') = v'$. The aim is to make put maximally defined.

Being fundamentally different in approach, both techniques have their respective strengths and weaknesses. We present a synthesis of the two techniques that improves on both. On the intersection of their applicability domains, the new combined technique achieves more than what a simple union of applying the original techniques side by side delivers. A focus is on enabling shape-changing updates. For example, if $get :: [\alpha] \rightarrow [\alpha]$, and thus $put :: [\alpha] \rightarrow [\alpha] \rightarrow [\alpha]$, we want $put\ s\ v'$ to be defined even when $length\ v' \neq length\ (get\ s)$.

The combination strategy pursued is essentially motivated by combining the specialties of the two approaches. Semantic bidirectionalization's specialty is to employ polymorphism to deal with the content elements of data structures in a very lightweight way. In fact, the shape and content aspects are completely separated, arbitrary updates to content elements are simply absorbed, but updates affecting the shape are completely outlawed. Syntactic bidirectionalization's specialty is to have a more refined notion of what updates can be permitted. But content elements often get in the way: by having to deal with both shape and content in the key step ("view complement derivation"), updatability is hampered. The combined approach divides the labor: semantic bidirectionalization deals with content only, syntactic bidirectionalization with shape only. As a result, the reach of semantic bidirectionalization is expanded beyond shape-preserving updates, and syntactic bidirectionalization is invoked on a more specialized kind of programs, on which it can yield better results, benefitting both.

References

1. K. Matsuda, Z. Hu, K. Nakano, M. Hamana, and M. Takeichi. Bidirectionalization transformation based on automatic derivation of view complement functions. *ICFP'07*.
2. J. Voigtländer. Bidirectionalization for free! *POPL'09*.