The essence of functional programming on semantic data

Martin Leinberger[1], Ralf Lämmel[2], Steffen Staab[1,3]

[1]Institute for Web Science and Technologies, University of Koblenz-Landau, Germany
[2] The Software Languages Team, University of Koblenz-Landau, Germany
[3] Web and Internet Science Research Group, University of Southampton, England

Programming with knowledge represented in description logics (DL) is error-prone. Untyped access, e.g., provided by the OWL API [1], does not leverage static typing which allows for proving the absence of runtime-errors. Mapping approaches, e.g., described by [2] cannot fully capture the conceptualization of semantic data. In [3], we present λ_{DL}, a typed λ-calculus with constructs for operating on semantic data. This is achieved by the integration of description logics into the λ-calculus for both typing and data access.

We rely on ALCOI as a basic description logic. We assume the possibility of checking whether axioms follow logically from a knowledge base as well as class expression queries. Furthermore, we only allow named individuals as query results to avoid infinitely large result sets.

Key design principles of λ_{DL} are to treat (1) *concept expressions as types* in the programming language (2) *subtype inference* by forwarding typing judgments to the knowledge system as axioms during type checking, (3) *typing of queries* to ensure satisfiability and proper result processing, (4) *class expression queries* as well as (5) *open-world querying* in which we treat axioms only as true if they are true in all models of the knowledge system.

Runtime semantics are modeled as a small-step operational semantics. As usual, runtime errors are modeled through stuck states during evaluation. Using the key design principles, we show that a straightforward extension of a basic λ-calculus, is sufficient to achieve a type-safe integration. The only exception comprises of accessing empty query results.

While the presented approach shows how a basic integration supporting subtyping and queries can be achieved, future work focuses on an extended query language, in particular, a subset of SPARQL, as well as type inference for all types of λ_{DL} and polymorphism.

References

1. M. Horridge and S. Bechhofer. The OWL API: A Java API for OWL ontologies. *Semantic Web*, 2(1):11–21, 2011.
2. A. Kalyanpur, D. J. Pastor, S. Battle, and J. A. Padget. Automatic Mapping of OWL Ontologies into Java. In *Proc. SEKE 2004*, pages 98–103, 2004.
3. M. Leinberger, R. Lämmel, and S. Staab. The essence of functional programming on semantic data. In *Proc. ESOP 2017*, LNCS. Springer.