Type inference for Java(X)
[Abstract]

Markus Degen, Peter Thiemann, and Stefan Wehr

Institut für Informatik, Universität Freiburg
{degen,thiemann,wehr}@informatik.uni-freiburg.de

1 Abstract

JAVA(X) is a framework for type refinement. It extends Java's type language with annotations drawn from an algebra X and structural subtyping in terms of the annotations. Each instantiation of X yields a different refinement type system with guaranteed soundness [1].

JAVA(X) has a concept of activity annotations paired with the notion of droppability. An activity annotation is a capability which can grant exclusive write permission for a field in an object and thus facilitates a typestate change (strong update). Propagation of capabilities is either linear or affine (if they are droppable). Thus, JAVA(X) can perform protocol checking as well as refinement typing.

To enable a type inference algorithm for JAVA(X) we setup a constraint type system and a constraint solver. The main concerns were addressed to the behavior of the ternary splitting relation and its impact on the complexity of the constraint solver.

Luckily, against the first intuition, the splitting relation for alias handling does not increase the complexity of the constraint solver. Since the splitting may completely be forced by one of the components of the splitting relation, the corresponding constraint may be solved directly without additional guessing or further, potentially exponential many, constraints.

As prove of concept we implemented the provided type inference for JAVA(X) and gained a running system with useful error messages for the programmer.

References

1. M. Degen, P. Thiemann, and S. Wehr. Tracking linear and affine resources with java(x). In *21st ECOOP*, LNCS, pages 550–574, Berlin, Germany, July 2007. Springer.