

# Type inference for functional interfaces in Java 8

– Abstract –

Martin Plümicke

April 14, 2013

Java will be extended in version eight by closures and functional interfaces, whereupon functional interfaces are interfaces with one method [Goe11]. Functional interfaces represent the types of closures, which are also called lambda expressions. The type inference mechanism will be extended, such that the types of the parameters of lambda expressions could be inferred. But types of complete lambda expressions will still not be inferable.

In [Plü11] we presented a type inference mechanism for Java lambda expressions with function types. In this approach we change the algorithm such that functional interfaces are inferred. Furthermore we replace the approach of well-typings, which was introduced by [FM88] by our type unification [Plü07].

## References

- [FM88] You-Chin Fuh and Prateek Mishra. Type inference with subtypes. *Proceedings 2nd European Symposium on Programming (ESOP '88)*, pages 94–114, 1988.
- [Goe11] Brian Goetz. State of the lambda. December 2011.
- [Plü07] Martin Plümicke. Java type unification with wildcards. In Dietmar Seipel, Michael Hanus, Armin Wolf, and Joachim Baumeister, editors, *Proceedings of 17th International Conference on Applications of Declarative Programming and Knowledge Management (INAP 2007) and 21th Workshop on (Constraint) Logic Programming (WLP 2007)*, number 434 in Technical Report, pages 234–245. Bayerische Julius–Maximillian–Universität Würzburg, Institut für Informatik, October 2007.
- [Plü11] Martin Plümicke. Well-typings for  $\text{Java}_\lambda$ . In *Proceedings of the 9th International Conference on Principles and Practice of Programming in Java*, PPPJ '11, pages 91–100, New York, NY, USA, 2011. ACM.