

# Reimplementation and Extension of Java Type Unification

Florian Steurer

DHBW Stuttgart Campus Horb, Informatik,  
Florianstraße 15, 72160 Horb, Germany

**Background** Java is a programming language where it is necessary to declare types explicitly. With type inference, type declarations can be omitted, making programs shorter and easier to write. In [1], a type inference algorithm for Java is introduced. This algorithm uses a constraint-based type unification, which, for a given set of type constraints, computes all possible typings of a program [2]. The type unification algorithm has been implemented as part of a compiler [3], which supports implicitly typed Java programs.

**Extension** The Java type unification is extended by real function types [4]. Additionally, some cases concerning wildcard types are added, that were not considered in [2]. This leads to a simplification of the algorithm.

**Reimplementation** The old implementation is too inefficient to handle inputs of relevant size. A complete reimplementation aims to improve efficiency by using immutable data structures.

## References

- [1] Martin Plümicke. “More Type Inference in Java 8”. In: *Perspectives of System Informatics - 9th International Ershov Informatics Conference, PSI 2014, St. Petersburg, Russia, June 24-27, 2014. Revised Selected Papers*. Ed. by Andrei Voronkov and Irina Virbitskaite. Vol. 8974. Lecture Notes in Computer Science. Springer, 2015, pp. 248–256.
- [2] Martin Plümicke. “Java Type Unification with Wildcards”. In: *17th International Conference, INAP 2007, and 21st Workshop on Logic Programming, WLP 2007, Würzburg, Germany, October 4-6, 2007, Revised Selected Papers*. 2007, pp. 223–240.
- [3] Martin Plümicke and Jörg Bäuerle. “Typeless programming in Java 5.0”. In: *4th International Conference on Principles and Practices of Programming in Java*. Ed. by Ralf Gitzel et al. Vol. 178. ACM International Conference Proceeding Series. Mannheim University Press, Aug. 2006, pp. 175–181.
- [4] Martin Plümicke. “Java Type System – Proposals for Java 10 or 11”. In: *Tagungsband des 18. Kolloquiums Programmiersprachen und Grundlagen der Programmierung (KPS’15)*. Ed. by Jens Knoop. Technische Berichte der TU Wien. Pörtschach, 2015.