

Featherweight Java: Theory and Implementation

by Benjamin Azan
Supervised by Brigitte Pientka

Featherweight Java, as introduced by Igarashi, Pierce, and Wadler [1], provides an entirely formalized, striped down version of Java. This simple basis offers a neat workbench for reasoning about language extensions, allowing types preservation and progress proofs to be easily written out. We propose Featherweight Java as a high level spring board for testing new additions to the Java language, particularly those related to type systems. To facilitate this, we have implemented different flavors of Featherweight Java in SML\NJ. These include, Featherweight Java, Generic Featherweight Java, and what we call BJava, an intermediate to FJ and Java itself. As an example of language experimentation, we will see how our study brought us to consider, and later dismiss, the use of conjunctive types in Java.

References

- [1] A. Igarashi, B. Pierce, and P. Wadler. Featherweight Java: A minimal core calculus for Java and GJ *In ACM Symp. on Object-Oriented Programming: Systems, Languages and Applications 1999*, pages 132-146, November 1999.