Formalization of the Java 5.0 Type System

– Abstract –

Martin Plümicke

As the base of a Java 5.0 type inference system we developed a formalization of the Java 5.0 type system.
The base of the type system is given in [TEPH05, GJSB05]. In Java 5.0 types are given as type terms, like

`Vector<Vector<Integer>>`, `Vector<? extends Object>`, or `? extends List<Object>`.

There, we have to differ between type terms, which can be used in Java 5.0 programs, explicitely, and type terms, which can only be inferred for expressions. The type terms

`Vector<Vector<Integer>>` and `Vector<? extends List<Object>>`

can be used, explicitely, whereas `? extends List<Object>` can only be inferred for an expression.
The type terms, which can be used, explicitely, are constructed over the class/interface names. The given class/interface names form a finite rank alphabet, where the number of parameters is mapped to the corresponding class/interface names. The parameters can be bounded by further type terms. This restrict the type term construction, such that as arguments only types are allowed, which fullfils the bounds. We describe the restriction of the rank alphabet as a type signature.
Then, we consider the inheritance hierarchy. From the *extends relation*, which is defined by the `extends` and the `implements` declarations, respectively, we derive the subtyping ordering.
Furthermore we extend the set of type terms by the types, which can only be inferred for expressions. Then, we continue the subtyping ordering on this extended set of type terms.
We will illustrate all of these formal definitions by examples.

References

[GJSB05] James Gosling, Bill Joy, Guy Steele, and Gilad Bracha. *The Java^{TM} Language Specification*. The Java series. Addison-Wesley, 3rd edition, 2005.

[TEPH05] Mads Torgersen, Erik Ernst, and Christian Plesner Hansen. Wild FJ. In Philip Wadler, editor, *Proceedings of FOOL 12*, Long Beach, California, USA, January 2005. ACM, School of Informatics, University of Edinburgh.