

Type system for JavaScript

Phillip Heidegger, Peter Thiemann

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

Many scripting languages are object-based and have weak, dynamic type systems to simplify the development of short programs. As larger programs are built with scripting languages, static type systems and analyses become important tools for detecting programming errors.

We define such a type system based on an imperative object-based calculus that models essential features of the JavaScript language, as an example of a typical and widely used scripting language. The calculus supports objects as property tables, type change during object initialization, and precise support for JavaScript's prototype mechanism.

Our work transfers the idea of recency abstraction from abstract interpretation to a type system. The corresponding feature in our system, the precise object pointer, enables strong, type changing updates on object types during a generalized initialization phase. The same precise object pointers enable an accurate treatment of the prototype mechanism. Unlike linear types, precise object pointers can be nested and mixed arbitrarily with imprecise object pointers in the description of a data structure. We have proved type soundness of the system and we have defined and implemented a typing algorithm.