TreatJS: Higher-Order Contracts for JavaScript (Artifact)

Matthias Keil and Peter Thiemann

Institute for Computer Science
University of Freiburg
Freiburg, Germany
{keilr,thiemann}@informatik.uni-freiburg.de

Abstract

TreatJS is a language embedded, higher-order contract system for JavaScript which enforces contracts by run-time monitoring. Beyond providing the standard abstractions for building higher-order contracts (base, function, and object contracts), *TreatJS*'s novel contributions are its guarantee of non-interfering contract execution, its systematic approach to blame assignment, its support for contracts in the style of union and intersection types, and its notion of a parameterized contract scope, which is the building block for composable run-time generated contracts that generalize dependent function contracts.

TreatJS is implemented as a library so that all aspects of a contract can be specified using the full JavaScript language. The library relies on JavaScript proxies to guarantee full interposition for contracts. It further exploits JavaScript's reflective features to run contracts in a sandbox environment, which guarantees that the execution of contract code does not modify the application state. No source code transformation or change in the JavaScript run-time system is required.

1998 ACM Subject Classification D.2.4 Software/Program Verification

Keywords and phrases Higher-Order Contracts, JavaScript, Proxies

Digital Object Identifier 10.4230/DARTS.1.1.1

Related Article Matthias Keil and Peter Thiemann, "*TreatJS*: Higher-Order Contracts for JavaScript", in Proceedings of the 29th European Conference on Object-Oriented Programming (ECOOP 2015), LIPIcs, Vol. 37, pp. 28–51, 2015.
http://dx.doi.org/10.4230/LIPIcs.ECOOP.2015.28

Related Conference 29th European Conference on Object-Oriented Programming (ECOOP 2015), July 5–10, 2015, Prague, Czech Republic

1 Scope

The artifact is designed to support repeatability of all the experiments of the companion paper, allowing users to test the contract system on a variety of benchmarks. In particular, it allows to include *TreatJS* in existing JavaScript code, to specify contracts by plain JavaScript functions, to construct contracts by an unrestricted combination of other contracts, and to enforce contracts in all contexts of use.

2 Content

The artifact package includes:

- the main source of *TreatJS*;
- a set of test cases to examine the feature of the contract system;
- modified version of the Google Octaine 2.0 benchmark suite;
- detailed instructions for using the artifact, provided as an `index.html` file.

Dagstuhl Artifacts Series, Vol. 1, Issue 1, Artifact No. 1, pp. 01:1–01:2
Dagstuhl Artifacts Series
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). The latest version of our code is available on GitHub: https://github.com/keil/TreatJS and at http://proglang.informatik.uni-freiburg.de/treatjs/.

4 Tested platforms

The implementation is based on the JavaScript Proxy API, a proposed addition scheduled for the upcoming ECMAScript 6 standard. *TreatJS* requires a SpiderMonkey JavaScript shell supporting the ECMAScript 6 standard.

The SpiderMonkey shell is not part of the artifacts package, but it can be downloaded from the Mozilla's repository. The SpiderMonkey build instructions can be found at the SpiderMonkey Build Documentation MDN page.

5 License

The MIT License (MIT) (`http://opensource.org/licenses/MIT`)

6 MD5 sum of the artifact

1bb0d8dd6efe40fbde8b1c182b50673e

7 Size of the artifact

5.1 MB