

УДК 004.912

ИССЛЕДОВАНИЕ РЕАЛИЗАЦИИ СТАТИЧЕСКОГО АНАЛИЗА ДЛЯ ОЦЕНКИ ИНТЕРОПЕРАбельНОСТИ ECMAScript-КОДА

Мальцев А.С. (ИТМО)

Научный руководитель – кандидат педагогических наук, доцент Государев И.Б.
(ИТМО)

Введение. JavaScript — язык программирования, соответствующий спецификации ECMAScript и являющийся главным языком в мире веб-технологий, имеет несколько сред исполнения: различные браузеры, такие как Google Chrome, Firefox, Safari и другие, а также серверные среды Node.js, Deno и Bun. Существование большого количества платформ для запуска JavaScript создает проблему разного уровня соответствия этих платформ спецификации языка. Поэтому разработчикам требуется следить за поддержкой тех функций языка, которые используют, на целевой платформе, под которую разрабатывается приложение. Ситуация усугубляется, если целевыми являются несколько платформ. Существуют базы данных совместимости, но проверять в них информацию приходится вручную, т.к. на данный момент не существует инструмента для автоматического анализа.

Основная часть. Решить проблему оценки интероперабельности ECMAScript-кода с целевыми средами его исполнения можно с помощью инструмента статического анализа, который будет выполнять автоматическую проверку исходного кода программы на соответствие целевым платформам на основании информации из баз данных совместимости. Анализатор состоит из 4-х основных компонентов:

1. Парсер для преобразования исходного кода в абстрактное синтаксическое дерево (АСД);
2. Анализатор АСД, который извлекает использованные в коде возможности языка;
3. Система вычисления типов констант и переменных;
4. База данных совместимости.

Исходный код преобразуется в АСД, анализатор при помощи системы вычисления типов извлекает информацию о том, какие возможности языка были использованы, затем формируется отчет в каких версиях каких платформ код будет работать, а в каких нет.

Выводы. Статический анализатор интероперабельности ECMAScript-кода способен решить проблему автоматической проверки совместимости приложения с различными платформами его исполнения.

Список использованных источников:

1. Park J., An S., Ryu S. Automatically deriving JavaScript static analyzers from specifications using Meta-level static analysis //Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. – 2022. – С. 1022-1034.
2. Brito T. et al. Study of JavaScript Static Analysis Tools for Vulnerability Detection in Node.js Packages //arXiv preprint arXiv:2301.05097. – 2023.
3. Камкин А. С. Введение в формальные методы верификации программ //Москва. – 2018.

Автор _____ Мальцев А.С.

Научный руководитель _____ Государев И.Б.