

要 旨

ANTLR を用いた信頼性の高い JavaScript 構文解析器の開発

中尾 成一

JavaScript のプログラムを実行するためのソフトウェアが JavaScript 処理系である。JavaScript 処理系の開発では JavaScript の構文解析器を作る必要がある。JavaScript の構文解析器は、JavaScript のプログラムを入力として受け取り、それが言語の文法に従ったものであるかどうかを調べて木構造の構文木を生成する。JavaScript のプログラムを構文解析する既存の構文解析ライブラリがあれば、それを利用することで開発コストを軽減することができる。Java 言語を使って JavaScript 処理系を開発する場合は、Java 言語で記述された JavaScript の構文解析ライブラリが必要になる。しかし、Java 言語で記述された JavaScript の構文解析ライブラリで信頼性が高いものは存在しない。

本研究では、Java 言語で記述された信頼性の高い JavaScript 構文解析ライブラリを開発した。本研究では信頼性を、様々な JavaScript のプログラムを入力として、仕様通りに構文木を構築できる割合として定義する。対象とする JavaScript のバージョンは ECMAScript 5th Edition (以下, ES5) である。本研究で開発した構文解析器は、ESTree というスキーマに従った構文木を生成する。ESTree は既存の JavaScript の構文解析ライブラリである Esprima が生成する構文木のスキーマである。本研究の構文解析器の一部は、構文解析器生成系の ANTLR を用いて、オープンソースで公開されている JavaScript の文法記述から自動生成した。それに加えて、自動生成された構文解析器が出力する解析木を ESTree に変換するプログラムを開発した。さらに、Esprima のテストケースを使って開発した構文解析器の信頼性を検査した。Esprima は、JavaScript で書かれた信頼性の高い JavaScript の構文解析器である。Esprima は、1300 個ほどの JavaScript プログラムからなる網羅率の高いテストケースを持っている。このテストケースは本研究で開発した構文解析器の信頼性を確か

めることに適している．構文解析器の信頼性検査では，各テストケースを構文解析し，構文木が正解と一致するかどうかを調べた．なお，Esprima は ES5 と上位互換性のある上位のバージョンに対応しているため，ES5 の範囲のテストケースのみを用いた．検査の結果，文法記述にいくつかの誤りが見つかった．これらの誤りを修正して，ES5 の主要な構文のテストケースをすべて正しく構文解析する構文解析器を開発した．

キーワード 構文解析器，JavaScript，プログラムテスト

Abstract

Development of a Highly Reliable JavaScript Parser by Using ANTLR

Seichi NAKAO

A JavaScript processing system is software that executes JavaScript programs. It is necessary to develop a JavaScript parser in development of JavaScript processing systems. JavaScript parsers receive JavaScript programs as their input, examine whether the programs connect with respect to the JavaScript grammar, and generate abstract syntax trees. If we have a parser library that parses JavaScript programs, we can reduce the development cost of JavaScript processing systems by using the library. To develop a JavaScript processing system written in Java, the parser should also be written in Java. Unfortunately, we could not find any reliable JavaScript parser library written in Java.

In this study, we developed a highly reliable JavaScript parser library written in Java. We define the reliability as the ratio that JavaScript programs are parsed into correct abstract syntax trees. The parser generates abstract syntax trees ESTree. ESTree is the schema of abstract syntax trees generated by Esprima that is an existing JavaScript parser library. We generated a part of the parser from an open source JavaScript grammar definition. In addition, we developed a program to convert parse trees into ESTree. Furthermore, in this study, we tested the reliability of the parser by using test cases of Esprima. Esprima is a highly reliable JavaScript parser written in JavaScript. Esprima has test cases with high code coverage consisting of about 1300 JavaScript programs.

These test cases have been suitable for testing the reliability of the parser developed in this study. The reliability of the parser was tested by comparing the abstract syntax trees outputted by the parser with the correct abstract syntax trees. Esprima is a higher version that is compatible with ES5, so only we used the test cases that were programs of the range of ES5. As a result of the test, we found some grammatical errors in the JavaScript grammar definition. By correcting these grammatical errors, we developed the parser that correctly parses all the test cases with main grammatical constructions of ES5.

key words parser , JavaScript , program test