DI Manuel Rigger, M.Phil.
Institute for System Software

T +43 732 2468 4356
F +43 732 2468 4345
manuel.rigger@jku.at

Secretary:
Birgit Kranzl
Ext 4341
birgit.kranzl@jku.at

Master's Thesis / Project in Software Engineering

An ECMAScript 2015-Compliant Automata-based Regular Expression Engine for Graal.js

Student:	Josef Haider
SKZ/Matr.Nr.:	921 / 1055777
Email:	josef.haider@khg.jku.at
Advisors:	DI Manuel Rigger, M.Phil. DI Dr. Christian Wirth
Start date:	01.10.2017

Graal.js is a JavaScript (ECMAScript 2017) engine, implemented in Java as an Abstract Syntax Tree (AST) interpreter, based on the Truffle framework [1], in which each operation is represented as a node. It compiles frequently executed functions to machine code using a partial evaluation mechanism. This partial evaluation mechanisms inlines each node of a function, which then forms a compilation unit that is optimized by Graal.js's compiler.

Currently, Graal.js uses a Java version of the JONI library [2] for Regular Expression (RegExp) parsing and matching. While JONI is largely ECMAScript 5.1-compliant it is slow when compiled by Graal.js. First, it cannot benefit from the Truffle language implementation framework which would allow a RegExp engine to specialize on the program state. Second, JONI is not well-suited for compilation using partial evaluation because it uses many methods of the JDK, which are inlined during partial evaluation and result in too large compilation units.

For Graal.js, a regular expression engine based on deterministic automata (similar to [3]) and Truffle, that uses simple operations on characters, would be better suited for partial evaluation. The goal of this thesis is to implement and evaluate the performance of such a regular expression engine.

The goals of this thesis are:

- An implementation of a regular expression engine based on deterministic automata.
- The exploration of possible performance benefits.
- A demonstration of correctness of the implementation by passing relevant tests from Test262 or TestV8, and Node.js modules.

Explicit non-goals are:

- Providing complete support for all possible regular expressions. Expressions that cannot be handled by the engine should be delegated to a different engine.

The work's progress should be discussed with the supervisor at least every 2 weeks. Please note the guidelines of the Institute for System Software when preparing the written thesis.

[1] Thomas Würthinger et al.. 2013. One VM to rule them all. In Proceedings of Onward! 2013. ACM, New York, NY, USA, 187-204.

[2] https://github.com/jruby/joni

[3] https://github.com/google/re2