Real-world Challenges for JavaScript Analysis: Efficient String Domains and Beyond

Alexander Jordan[*] and Roberto Amadini[†]

[*]Oracle Labs, Brisbane
[†]Department of Computing and Information Systems, The University of Melbourne

As part of a collaboration under ARC linkage project LP140100437, researchers at Oracle Labs Australia and the University of Melbourne have been collaborating to address some of the hard challenges related to static program analysis for JavaScript.

While JavaScript is one of the most popular programming languages today, tools that can automatically alert developers to unwanted behavior or security vulnerabilities are either drastically limited (e.g., *linter* and *checker* tools) or fail to scale to real-world applications. This can in part be explained by the JavaScript language itself. JavaScript is dynamically typed, has higher-order functions and supports reflective (string-based) access to the properties of objects. In addition, the heavy use of third-party libraries, which may provide substantial core building blocks of an application, as well as meta-programming techniques (e.g., dynamic code generation) and event-driven applications, operating in a variety of execution environments further complicate analysis.

Based on our experience with industry-scale software products, we conclude that static analysis alone cannot succeed in analysing JavaScript applications. Rather, a pragmatic analysis tool is forced to abandon soundness in certain parts in favour of useful under-approximations. These can be either provided by a user or, more realistically and at scale, through supplementary (dynamic) analysis techniques.

Another core property that is a prerequisite for automated reasoning about practically any kind of runtime property in JavaScript, is precise string analysis. Although the literature presents a considerable number of abstract domains for capturing and representing specific aspects of strings, none of the existing JavaScript analysers was able to support the flexible combination of different string abstract domains. A closely related question that has not been address before, is how information can be efficiently shared amongst a collection of abstract domains. For this purpose, we propose an analysis framework for constructing, automating and refining products of string abstract domains using the class of regular languages as the reference domain.

We start this talk by giving a brief overview of the challenges we need to face when applying static analysis techniques to current systems built on JavaScript, especially modern web applications. We continue by reporting on our work on string analysis as part of abstract interpretation of JavaScript, which has recently lead us to augment a state-of-the-art JavaScript analysis tool with an enhanced framework for string domains.