Abstract

WebAssembly (Wasm) is an emerging compilation target for programming languages, executed by web browsers and sandboxed environments like the Web Assembly System Interface (WASI)[1]. It offers portability, security and near-native execution speed, making it an attractive compilation target.

However, the Minimum Viable Product (MVP) release of Wasm lacks built-in support for many useful numeric representations such as arbitrary precision integers, exact rational numbers, and exact/inexact complex numbers. Source languages like CommonLisp, Haskell, Python and Racket require these representations at run-time to back their built-in numeric types and support their standard libraries.

This lack of representations, and operations that can be used on mixtures of these representations, makes compiling these source languages to Wasm modules unnecessarily difficult. To tackle this issue I have developed a library in C++ for representing these numeric abstractions and cross-compiled it to Wasm, the source code for which is available here:

https://github.com/ScottButler87/ExtendedNumerics

This thesis details background information about this issue and also serves as an exposition of the performance characteristics of the ExtendedNumerics library. The library's implementation is outlined and comparisons are drawn between its performance in native and Wasm compilations. Methods used for verifying correctness and benchmarking performance are also explained.

Despite Wasm's unusual choice of LEB128 as a core number encoding, the library performs well when cross-compiled for browsers/Nodejs. While there still exist other barriers to compiling Haskell, Racket, etc. to Wasm, this library serves as an effective bridge for crossing the existing numeric representation gap.