Static Resource Analysis of Smart Contracts – Milestone 1

Nick Roberts
Carnegie Mellon University
nroberts@andrew.cmu.edu

December 18, 2017

1 Summary of updates

The goal of the project remains to be to write a compiler for a subset of OCaml, targeting the Ethereum Virtual Machine (EVM). Based on discussions with my advisor, Professor Jan Hoffman, I am promoting a part of my stated 100% goal to my 125% goal: any static resource analysis is now a "reach" goal.

2 Accomplishments

I have read the foundational literature for both the OCaml compiler and the EVM bytecode primitives. To put this in practice, I have written a basic compiler for a straight-line code segment of OCaml, including only `let`-bindings and binary operator expressions. Finally, based on this experience, I have a draft of an OCaml module whose procedures grant access to EVM primitives inaccessible from a high-level language.

2.1 Meeting the milestone

What I've accomplished corresponds to what I intended to accomplish by this milestone.

2.2 Surprises

Solidity, an existing language targeting the EVM, has its quirks. Even simple programs compile to a complex sequence of bytecode instructions. It's crucial for me to understand another language's approach to this task, so I intend to spend more time disentangling Solidity before next semester starts.

3 Looking ahead

3.1 Revisions to future milestones

As discussed above, I am relocating static resource analysis to my 125% goal instead of my 100% goal. This affects my later milestones. For my April 4th milestone, instead of a gas cost inference mechanism, I now want to run a smart contract on the Ethereum blockchain. My April 18th milestone remains, tentatively, to support some sort of gas cost inference, but for this even a minimal technique will suffice. I retain May 2nd as an evaluational milestone.

3.2 Resources needed

Seeing that I have already installed the necessary software, I don't anticipate needing further resources. So far, I have installed and configured the OCaml compiler's source code, the Solidity compiler, and a Java implementation of the EVM.