School of Computing
CA326 Year 3 Project Proposal Form

SECTION A

Project Title: Fafel- A Functional Programming Language for EVM Compatible Blockchains
Student 1 Name : Senan Warnock   ID Number : 20752725
Student 2 Name : Zak Smith   ID Number : 20723579
Student 3 Name ______________________________   ID Number ___________
(A third team member is exceptional and requires detailed justification.)
Staff Member Consulted Geoffrey Hamilton

Project Description (1-2 pages):

Description:

“Finally, a Functional Ethereum Language” == Fafel.

The goal of this project is to create a rudimentary functional smart contract programming language and an associated compiler targeting the Ethereum Virtual Machine. The compiler translates source code of our language to EVM bytecode which the virtual machine executes. The EVM bytecode is a hexadecimal representation of the virtual machine’s opcodes. The architecture is similar to how Java is run with the Java Virtual Machine.

What is Ethereum and the EVM?

Ethereum is a blockchain with a computer embedded in it. It is the foundation for building apps and organizations in a decentralized, permissionless, censorship-resistant way.
In the Ethereum universe, there is a single, canonical computer (called the Ethereum Virtual Machine, or EVM) whose state everyone on the Ethereum network agrees on. Everyone who participates in the Ethereum network (every Ethereum node) keeps a copy of the state of this computer. Additionally, any participant can broadcast a request for this computer to perform arbitrary computation. Whenever such a request is broadcast, other participants on the network verify, validate, and carry out ("execute") the computation. This execution causes a state change in the EVM, which is committed and propagated throughout the entire network.

There are not that many high level smart contract programming languages on Ethereum, with Solidity alone accounting for well over 95% of all contracts. There is also Vyper, which is closer to a python implementation of a smart contract language.

Our language will be designed as a statically typed functional programming language, with the aim to allow for formal verification of smart contracts and allow for the language to be simpler and easier to understand.

To sum up, our project can be broken down into two major sections:

1. Design and implementation of Fafel, a smart contract programming language for the ethereum virtual machine.

2. Compiler construction for Fafel.

Where our compiler is broken up into the following pieces:

1. Lexer
2. Parser
3. Type Checker / Static Analysis
4. Optimisation
5. Code Generation

Division of Work

We will share equally the workload and ensure that each commit is reviewed by the other team member so that we both understand every part.

We will try to do live pair programming sessions where possible.

Programming Languages

Haskell

Programming Tools

Parsec, Happy (?)

Learning Challenges

For us this is an exciting, but substantial challenge. We are currently inexperienced with compiler construction, Haskell and language design. Thankfully a lot of the material covered this year covers language design and

Haskell. All elements of this project are expected to be challenging. Should we succeed, we will become much better programmers.

Hardware / Software Platform

Mac / Linux hardware, VS Code

Special Hardware / Software Specification

N/A