University of Amsterdam

Individual Written Assessment

Designing `legalease`, a Smart Contracts programming language

February 21, 2021

Lecturer:
dr. Ana Oprescu

Student:
René Kok
13671146

Course:
Programmeertalen

1 Introduction

Smart Contracts [1] are key to ...

1.1 Relevant characteristics of application domain

1.2 Use case description

2 Analysis

2.1 The type system

2.1.1 Type checking

Because smart contracts are designed for transactions, you want a smart contract to be accurate and secure before executing the smart contract. That is why I chose to have the language static typed. Static type checking has the advantages that the smart contract is guaranteed to meet a number of type safety features for all possible inputs. In addition, a static typing language is better optimized as opposed to dynamically typed language, because the compiler knows if a program is correctly typed. This results in a smaller and faster binary because no dynamic safety checks need to be performed.

2.2 State management

2.3 Compilation/interpretation strategy

2.4 Evaluation strategy (lazy/eager)

2.5 Parameter evaluation strategy (call by value/reference)

2.6 Communication semantics (synchronous/asynchronous)

2.7 Higher-order functions

2.8 Anonymous functions

3 Discussion

4 Conclusions

References

[1] “Relevant Research ...” In: ().