

Literary Review Rough Draft

1 Problem Context

The problem of this project is that there are issues with many commonly used programming languages. I would like to highlight some of these issues by taking bad design to an extreme with a joke programming language. The joke programming language made in this project is not meant to be used, but highlights specific problems that plague many languages such as Java, Javascript, and python. The main issue this project will deal with is ambiguous syntax in regards to specifying types. For example Javascript lets you do bizarre things like add an int to a sting. In python, you can have arrays with different data types like [1, 'hello', True]. It is concerning that programming languages used for tasks that may be sensitive like banking or voting can be coding in a language that can do operations that may allow for vulnerabilities from a hack or accidentally hurt someone.

2 Technical Background

While this programming language is a joke, the process for writing a joke programming language is the same as a normal one, just that the semantics of this programming language are going to cause problems for the coder. The basic components of a programming language are the parser and the interpreter. The parser takes the raw text from the file and converts it into an abstract syntax tree (AST). This AST is then put through the interpreter to process the instructions of the program which then runs that program. There are many different ways to write a parser and an interpreter so I specify the approach that will be taken by this project. The parser for this project will make use of lexing, which is the process of taking the text from the programming and converting the keywords, symbols and names into tokens that then are converted into the AST. This project will be written in Ocaml, a functional language. For those not familiar with the paradigm of functional programming, the idea is that code is based on pure functions for everything. The type system of Ocaml will be very useful for the interpreter and AST for this project.

3 Prior Work

The joke programming language being created for this project has not been done before and nothing found on the internet works like it. However, the idea of exaggerating bad features of normal programming languages or just taking a simple concept to the extreme has been done many times. The most famous esoteric programming language is brainfuck, a language meant to use as few symbols as possible to perform basic operations on the stack. Morr (2014) [1]. Brainfuck is a very different kind of language being a low level language dealing with pointers and memory access whereas my project is a high level language so although the concept of a language that is not meant to be used in similar, the way the parser and interpreter is written for brainfuck is much different. An example of a joke language that is more similar to mine in design is J.A.V.A., which mocks objects by making the coder jumps through several hoops to get the ability to actually start coding.

References

- [1] Morr, Sebastian. "Esoteric Programming Languages: An introduction to Brainfuck, INTERCAL, Befunge, Malbolge, and Shakespeare". In: (2014).