

COS333 Practical 1

Dylan Kapnias
u18108467

August 15, 2022

1 Research Questions

1.1

An esoteric programming language is a programming language that has been designed not for practical use, but rather as a language meant for experimentation or as a fun joke project [2].

1.2

a.)

Advantage:-

Disadvantage:- One of the disadvantages with all of the Fungus family of languages (with the exception of Befunge-98), is the limit to the size of the playfield (the space in which code can be entered), thus making these languages not able to be Turing-complete.

b.)

Advantage:-

Disadvantage:-

1.3

brainfuck

This language was invented by Urban Müller in 1993, whilst attempting to make a language such that he could write the smallest possible compiler for the version 2.0 of Amiga OS [1]. There is not much to the language in terms of syntax as other than the characters `> < + - . []` everything else is considered a comment, and thus is ignored. Semantically the language is Turing-complete and operates by manipulating an array of memory cells. The following is a simple "Hello World!" in this language:-

```
+++++++ [>++++ [>+++>++++>++++>+<<<<-]>+>->+ [<] <-> .>---.+++++. .+++> .<- .< .+++ .---  
---.----- .>+ .>+ .
```

Piet

This language was invented by David Morgan-Mar, who name the language after a pioneer in the sphere of geometrical abstract art, Piet Mondrian [3]. The syntax of the language is quite interesting due to the fact that the way to input instructions is through the use of the 20 colours that the language supports. Semantically the language is Turing-complete and operates by controlling and

manipulating a stack of integer values. All the colours are cyclically linked and thus the instructions for the stack manipulation are determined through the transition of the colours as the interpreter flows through the program. The following figure is of a simple "Hello World!" program in this language:-



1.4

Design by contract is a way of software design that focuses on specifying contracts of interaction between the components of the system, assuming that the components interact with each other in a client-server model of architecture. Thus, the contract between the client and the server provides the client the assurance that the server will uphold it's promises as long as both the server and the client abide by the postconditions and the invariants set on the contract. Due to these contracts essentially just being assertions, the esolangs Daft and ComeFrom2 should also have native support design by contract.

References

- [1] esolangs.org. brainfuck, 2022.
- [2] esolangs.org. Esoteric programming language, 2022.
- [3] David Morgan-Mar. Piet, 2018.