Research Questions - Task 4

Scott Bebington - u21546216

09 March 2024

Question 1

The Turing completeness of TeX means that TeX documents can be programmed to run any algorithm that can be run by a Turing machine. Because TeX documents are Turing complete, it means that they can solve complex algorithms at very fast speeds. The downside to this is that it can be used to run malicious code and can be used to exploit systems.

Question 2

Esoteric programming languages are programming languages that are not designed to be used for everyday use, rather they are a way to test proof of concept ideas. They are often designed to be difficult to write in and read.

Question 3

For: Esoteric programming languages are intended to show a programmer's artistic side in creating an ”Alternative” programming language, which should not be used in day-to-day use due to its difficulty in reading and writing.

Against: Esoteric programming languages may start off as a creative outlet for programmers but in time could evolve into fully functional programming languages that could be used in day-to-day use, or to solve a specific problem that other programming languages cannot solve.

Question 4

Language 1: Whitespace [1]

- Designers: Edwin Brady and Chris Morris
- Year of initial design: 2003
- Syntactic and semantic characteristics: Whitespace is an esoteric programming language that uses only spaces, tabs, and new lines to write code, thus making it extremely difficult to read and debug. The commands are tabs (Negative / 1), spaces (Positive / 0), and new lines (Stack Manipulation). This is used to code in binary which is then converted to text to be read by the computer.
- Turing completeness: Yes, Whitespace is Turing complete.

Language 2: Brainfuck [3]

- Designer: Urban Müller
- Year of initial design: 1993
- Syntactic and semantic characteristics: Brainfuck is an esoteric programming language that uses only 8 characters that execute a command.
- Turing completeness: Yes, Brainfuck is Turing complete.

Question 5

For: [4] One reason why Bash can be considered a programming language is it can store variables, complete complex algorithms, it has its own syntax and can tell the computer to do certain tasks. It can also be compiled.

Against: [5] Due to the nature of Bash, the fact that it is a low-level programming language makes it very difficult to program large and complicated because of its lack of built-in data structures.

Question 6

ALF combines the following paradigms: [6] [7]

- Functional and logic programming techniques
- Solve literals and narrowing to evaluate functional expressions

Question 7

Syntax: [8]

- Start/End symbols: Represent the beginning and end of the program.
- Decision symbols: Represent conditional statements and branching in the program.
- Looping symbols: Represent iterative operations and loops in the program.
- Input/Output symbols: Represent input and output operations in the program.
- Process symbols: Represent actions and operations performed by the program.

Semantics: [9]

- The semantics of Visual Logic are based on the execution of the flowchart.
- Each symbol represents a specific action or decision, and the program executes sequentially following the flowchart's structure.

Advantage: It provides a basic visual aid for understanding how the process of the program works. This makes it a valuable tool for teaching programming concepts to beginners.

Disadvantage: Programs that become larger and more complicated make visualizing the flowchart difficult and can become confusing for beginner and advanced programmers.

Question 8

The Dr. Memory tool is a tool similar to Valgrind in that it checks for memory leaks and memory corruption and monitors memory usage and allocations [10]. The main difference is its ability to check for memory leaks and memory corruption in Windows. [2]

References

[1] Whitespace. https://esolangs.org/wiki/Whitespace. Accessed: 09 March 2024.

[2] Using dr. memory with chromium. https://www.chromium.org/developers/how-tos/using-drmemory. Accessed: 09 March 2024.

[3] Brainfuck. https://en.wikipedia.org/wiki/Brainfuck. Accessed: 09 March 2024.

[4] Bash. https://www.gnu.org/software/bash/manual/bash.html. Accessed: 09 March 2024.

[5] Bash (unix shell). https://en.wikipedia.org/wiki/Bash$_{(}Unix_{s}hell).Accessed:09March2024.$

[6] Programming languages: Types and features. https://www.chakray.com/programming-languages-types-and-features. Accessed: 09 March 2024.

[7] Alf. https://www.informatik.uni-kiel.de/ mh/systems/ALF.html. Accessed: 09 March 2024.

[8] Visual logic. https://www.visuallogic.org/. Accessed: 09 March 2024.

[9] Flowchart. https://en.wikipedia.org/wiki/Flowchart. Accessed: 09 March 2024.

[10] Dr. memory. https://en.wikipedia.org/wiki/Dr.$_{M}emory.Accessed : 09March$2024.