COS333 Practical 1

u21467456

7 August 2023

1 Research Component

1. A Turing machine is a hypothetical machine proposed by Alan Turing that, given enough time and memory, can solve any problem so long as that problem may be expressed in code instructions and has an answer that can be calculated [1]. Thus, a programming language is considered "Turing Complete" if it can imitate the Turing Machine, i.e., can solve any given problem that could be solved by a Turing Machine [1].

2. An esoteric programming language (Esolang) is a programming language designed to test the limits of programming in a way that is unconventional, interesting, and entertaining [4]. Esolangs are technically Turing Complete but are by no means meant to provide an efficient or effective solution to solving problems like traditional programming languages. Instead, Esolangs are designed to explore programming language design in a unique way and provide a proof of concept to some of the theory behind programming language design and programming in general [4].

3. It is true that esolangs are typically created as parodies to traditional programming languages and concepts, with a fair portion of these languages being created as jokes or as side projects and all of them in a sense "parodying" traditional programming language design principles by making use of unconventional syntax, data and instruction representations, as well as unusual or difficult interpretation / compilation of the language (for example, Befunge, which can be read in 4 directions) [4]. Esolangs are in name "esoteric" and aren't meant to be taken seriously or be used for typical projects / software (many of them have very limited practical application). Thus, it can be argued that these languages are simply meant to be amusing diversions.

   However, these languages are also important studies in the limits of programming and programming language design because they provide an unconventional and unique viewpoint for programmers on what is possible for computers to interpret, and highlights some of the important programming

language concepts such as readability and writability (that a lot of these languages aim to undermine), which, without these languages, can seem like abstract or unimportant concepts for someone learning programming. Esolangs challenge the traditional formula for constructing a language, which positively impacts the development of languages in general because for new and better languages to be developed and programming in general to progress, we need to know what is possible. Esolangs promote creative and unconventional thinking, and this line of thinking may lead to new developments in traditional languages.

4. The Chef programming language was developed in 2002 by David Morgan-Mar, who has designed various esoteric languages over the years [2].The language is stack-oriented, whereby variables can be placed onto a stack and operations can then be performed on the stack. The language is designed to resemble a recipe, and thus consists of ingredients, and a method [9].

   Ingredients are variables. They are referenced by a programmer-given name (e.g., eggs). Ingredients also have an integer value corresponding to its quantity (e.g., 110), and a state, either liquid, dry or unspecified, which determines how the value of the ingredient is printed (e.g., liquid = Unicode). The language is untyped and scalar, meaning the language does not support arrays / array processing [9].

   The Method consists of a sequence of written instructions each beginning with a reserved keyword in Chef that describes the operation that takes place, followed typically by the ingredient (variable) name and then either a mixing bowl, or baking dish [9]. Mixing bowls or baking dishes are two types of stacks in Chef. Baking dishes are used exclusively to print values, while other operations are performed on ingredients in mixing bowls [9]. There can be multiple of these dishes, in which case the dish is referenced by its ordinal index (e.g., the 2nd baking dish) [9].

   As mentioned, a program in Chef is structured as a recipe, and thus must consist of a recipe title (descriptive title of the program), ingredient list, method, and "serves" statement, which prints the contents of each of the baking dishes [9]. Optionally, one can also add a cooking time, oven temperature and comments (under the title) to make the program more descriptive [9]. Procedures are structured as auxiliary recipes, which operate on either their own variables and stack, or copies of the main recipe's. Once completed, the output of the auxiliary recipe is placed on top of the main recipe's stack [9].

   Some examples of instructions in Chef include

   ```
   Put [ingredient] into [index] mixing bowl
   ```

   which pushes the variable's value onto the stack, and

```
Fold [ingredient] into [index] mixing bowl
```

which pops the value from the stack and stores it in that variable. Loops are supported in Chef through the `[Any verb]` and `[Any verb] until [verb-ed]` commands [9]. Chef is Turing complete as it supports branching and other basic commands needed to simulate a Turing Machine, although working with floating point values is a challenge.

Printing "Hello, World!" in Chef:

```
Lobsters with Fruit and Nuts.

This recipe prints "Hello, World!" in a most delicious way.

Ingredients.
72 g hazelnuts
101 eggs
108 g lobsters
111 ml orange juice
44 g cashews
32 g sugar
87 ml water
114 g rice
100 g durian
33 passion fruit
10 ml lemon juice

Method.
Put lemon juice into the mixing bowl.
Put passion fruit into the mixing bowl.
Put durian into the mixing bowl.
Put lobsters into the mixing bowl.
Put rice into the mixing bowl.
Put orange juice  into the mixing bowl.
Put water into the mixing bowl.
Put sugar into the mixing bowl.
Put cashews into the mixing bowl.
Put orange juice into the mixing bowl.
Put lobsters into the mixing bowl.
Put lobsters into the mixing bowl.
Put eggs into the mixing bowl.
Put hazelnuts into the mixing bowl.
Liquify contents of the mixing bowl.
Pour contents of the mixing bowl into the baking dish.
```

```
Serves 1.
```

The JSFuck programming language originally began as a JavaScript encoding project (jjencode) by Yosuke Hasegawa in 2009 that made use of 18 symbols to obfuscate any JavaScript code. In 2010, this was narrowed down to 6 characters, `[]()!+`, and JSFuck was officially created by Hasegawa by the end of 2010 [5]. JSFuck is completely valid JavaScript code, and thus does not require its own interpreter or compiler and can run on any machine that can interpret JavaScript, thus it is sometimes used in XSS attacks due to its ability to bypass malicious code detection filters [5].

JSFuck takes advantage of JavaScript being a weakly typed language, thus allowing an expression of any type to be evaluated, including empty arrays and Booleans, which JSFuck makes use of to produce numbers and character strings. Since an empty array prepended by a plus sign is converted into numeric 0 (`+[]`), the value true can then be created by negating this (`!+[]`) to force JavaScript to convert the numeric 0 to Boolean false (additionally, `![]` will also produce false, and true can thus be created by `!![]`) [5]. This can then in turn be converted to a numeric 1 by prepending a plus sign again (`+!+[]`, since JavaScript will convert the value to an integer). Other digits are then made by chaining multiple expressions for 1 together, and so on [5].

Letters can be accessed through indexing string representations of numeric or Boolean values (which can themselves be obtained by summing the JSFuck representation of that value with an empty array, i.e., `\true"+[]` or `!+[]+[]`) [5].

Triggering JavaScript functions can be done by calling the function constructor (an empty array) followed by the function name as a string (e.g., `[][\alert()"]`) [5].

As is apparent from the examples here, JSFuck is extremely verbose (the JSFuck equivalent for the JavaScript `alert(\Hello, World!")` is 8148 characters long) and very difficult to read, making it perfect for code obfuscation [6][5]. JSFuck is also fully Turing Complete, as it evaluates to valid JavaScript code, and can thus simulate any instruction.

A Portion of `alert("Hello, World!")` *in JSFuck:*

```
[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]
+(!![]+[])[+[]]][([][(![]+[])[+[]]+(![]+[])[!+[]+!+[
]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!+[]+!+[]+!+[
]]+(!![]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[
])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+[]]]+([][[]]+[])[
```

```
+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]
+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+(![]+
[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]+[])[!
+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]
+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]]
)[+!+[]+[+[]]]+(!![]+[])[+!+[]]]((!![]+[])[+!+[]]+(!
![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+([][[]]+[])[+
[]]+(!![]+[])[+!+[]]+([][[]]+[])[+!+[]]+(+[![]]+[][(
![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!!
[]+[])[+[]]])[+!+[]+[+!+[]]]+(!![]+[])[!+[]+!+[]+!+[
]]+(+(!+[]+!+[]+!+[]+[+!+[]]))[(!![]+[])[+[]]+(!![]+
[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]
+(!![]+[])[+[]]])[+!+[]+[+[]]]+([]+[])[([][(![]+[])[
+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+
[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+(![]
+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+
[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]
]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+(
[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]
...
```

5. Emacs utilizes C and a dialect of Lisp as its back-end. Lisp is used to provide users with various commands and macros users can execute to carry out various actions (available as Lisp functions) in the Emacs editor efficiently through different keybindings or combinations of keybindings [7]. For example, `Ctrl+f` is used to open a file specified by the user, `Ctrl+w` is used to copy text and `Ctrl+k` is used to close a file [7]. This is an example of the imperative programming paradigm.

6. Scala is a programming language based off the Java language. Scala extends on Java's object-oriented approach by implementing a pure object-oriented and functional language as every value is treated as an object described by a class, and every function can be represented and passed around as a value [8].

7. Alice is an educational programming language and software that was designed to teach individuals about object-oriented programming concepts in a way that is accessible and easy to understand [3]. It was first developed in 1994 at the University of Virginia and continued to be developed from 1997 at Carnegie Mellon University [3].

   All coding in Alice is done through its IDE software. Alice focuses on object-oriented concepts through manipulation of interactive 3D environments via the Alice IDE [3].

   Objects are placed into the scene, and code snippets (represented as tiles)

can then be dragged into the program (animation) to manipulate those objects and construct a story. Objects in the scene can be accessed through the scene hierarchy (including the camera and lights), where after several intuitively named properties, methods, and functions specific to that object can be accessed as tiles and dragged into the program [3]. Objects also support inheritance by having children objects visible in the hierarchy that can be accessed [3]. Many of the methods use parameters and variables which can be clicked on and given values once in the program [3].

In addition, there are some control structure tiles that allow for program looping and branching as well as parallel processing, for example: `Do in order`, `While`, `For all together` and `Wait`. Methods can be placed in and around these control structures to direct program flow [3].

Every method tile, control structure, and variable are descriptive, which increases the readability of the programs created. There are also a variety of customizable functions and combinations, making programs very expressive and thus writable.

8. The Valgrind tool is used for detecting memory leaks and debugging memory issues in programs.

References

[1] Binance Academy. Turing complete, 2023.

[2] Esolang. Chef, 2021.

[3] Wikimedia Foundation. Alice (software), Aug 2023.

[4] Wikimedia Foundation. Esoteric programming language, Aug 2023.

[5] Wikimedia Foundation. Jsfuck, Feb 2023.

[6] Yosuke Hasegawa, 2012.

[7] Klaatu. What is emacs?, 2023.

[8] Martin Odersky, Philippe Altherr, Vincent Cremet, Gilles Dubochet, Burak Emir, Philipp Haller, Stéphane Micheloud, Nikolay Mihaylov, Adriaan Moors, Lukas Rytz, and et al. Table of contents, 2023.

[9] Progopedia. Chef programming language, Jun 2011.