COMPSCI/SFWRENG 2FA3

Discrete Mathematics with Applications II

Winter 2021

Extra Credit Assignment 5

Dr. William M. Farmer

McMaster University

Revised: March 31, 2021

Extra Credit Assignment 5 consists of one problem. You must write your solution to the problem using LaTeX.

Please submit Extra Credit Assignment 5 as two files, `EC_Assignment_5_`*`YourMacID`*`.tex` and `EC_Assignment_5_`*`YourMacID`*`.pdf`, to the Extra Credit Assignment 5 folder on Avenue under Assessments/Assignments. *`YourMacID`* must be your personal MacID (written without capitalization). The `EC_Assignment_5_`*`YourMacID`*`.tex` file is a copy of the LaTeX source file for this assignment (`EC_Assignment_5.tex` found on Avenue under Contents/Assignments) with your solution entered after the problem. The `EC_Assignment_5_`*`YourMacID`*`.pdf` is the PDF output produced by executing

```
pdflatex EC_Assignment_5_YourMacID
```

This assignment is due **Sunday, April 18, 2021 before midnight.** You are allow to submit the assignment multiple times, but only the last submission will be marked. **Late submissions and files that are not named exactly as specified above will not be accepted!** It is suggested that you submit your preliminary `EC_Assignment_5_`*`YourMacID`*`.tex` and `EC_Assignment_5_`*`YourMacID`*`.pdf` files well before the deadline so that your mark is not zero if, e.g., your computer fails at 11:50 PM on April 18.

Although you are allowed to receive help from the instructional staff and other students, your submission must be your own work. Copying will be treated as academic dishonesty! If any of the ideas used in your submission were obtained from other students or sources outside of the lectures and tutorials, you must acknowledge where or from whom these ideas were obtained.

Extra Credit Problem [2 bonus points]

Given an encoding scheme over $\{0, 1\}$ for Turing machines, let A the set of codes for total Turing machines. Prove that A is not r.e. What does this result say about programming languages?

Mohammad Omar Zahir, zahirm1, 21 April, 2021

To prove that A is not recursively enumerable, we will first assume that A is r.e. and attempt to derive a contradiction. We know that for a set to be r.e., there must be a turing machine M that can accept the set. This turing machine will behave very similar to a universal turing machine, UTM, which takes in the encoding of a turing machine and determines whether it will halt or not. This is similar to the turing machine in the halting problem, except that this set consists of the encoding of all total turing machines, meaning that these turing machines will always halt. However, we know that this universal turing machine will not be able to accept the encoding over the set A if the set is uncountable. We can show this to be the case through the diagonalization argument. If we represent the bijective mapping of all the possible total turing machines and their enocding in the form of a diagonalization table, we can formulate a new encoding in the table by flipping the codes along the diagonal, we can see that we will get a unique encoding that does not exist anywhere else in the mapping. If the set A was infact r.e., then we would not have been able to produce such an encoding in this manner.

From the conclusion that we have made above, the correlation that we can make with programming languages is that they represent a set that is very similar, if not the same as A. Thinking abstractly, our use of programming languages is very similar to an encoding scheme, where we use these encodings to be processed by a computer, which is a turing machine. As such, from the result above, we can claim that programming languages and the set of programs that can be proved as a result of them, which we can consider the 'encodings' can also be confirmed to be r.e.. This conclusion makes sense logically as well as it is a well-known fact that there are an uncountable number of programs that can be constructed through the use of programming languages.