3C04 - Exercises 2

Sean Holden
Department of Computer Science
University College London

Exercises marked with a star are a little more difficult than others. Exercises marked with two stars are more challenging still.

1. An alternative definition of a recursive language is as follows: L is recursive if and only if there is a Turing machine M having only a single halting state such that if $x \in L$, M outputs a 1 for input x, and if $x \notin L$, M outputs a 0 for input x.

   Prove that this definition is equivalent to the definition given in the lectures.

2. Some definitions of a TM do not use the $\triangleright$ symbol to stop the head from moving off of the tape. In such definitions it is generally case that if the head moves off of the tape then computation stops and the output of the TM is undefined.

   Prove that this alternative definition of a TM is equivalent to the definition used in the lectures. In the sense that any computation performed by a Turing machine under one definition can also be performed by a Turing machine under the alternative definition. (This needs to be proved—an appeal to Church's thesis is insufficient!)

3. It was stated in the lectures that, when given any string of 0s and 1s, there is a well-defined procedure for deciding whether or not this string can be obtained by encoding a Turing machine. Church's thesis therefore tells us that it should be possible to design a Turing machine that takes as input any string of 0s and 1s, which always halts and succeeds, which produces an output of 1 if the input can be obtained by encoding some Turing machine, and which produces an output of 0 otherwise. Design such a Turing machine.

4. In the lectures, a Turing machine was presented for performing unary addition. Encode this Turing machine using the encoding scheme presented in the lectures.

5. Using the coding scheme presented in the lectures, find the TM that is encoded to give the following string:

$$\begin{array}{c}
101111101011100101111110110111111001011 \\
010111001101111101011110011011111101110 \\
111110011011011011100111011111011101111 \\
00111011111101110111001110110111011100
\end{array}$$

6. Prove that the set of all functions from $\mathbb{N}$ to $\mathbb{N}$ is uncountable. What does this allow you to conclude?

7. Prove that all non-recursive languages are infinite.