3 Compiler Construction (TGG)

(a) Explain why some programming languages require automatic memory management (“garbage collection”) for program execution. [4 marks]

(b) At a given point in the execution of a program, what can be considered as garbage? How can garbage be located in memory? [4 marks]

(c) Suppose a programmer is implementing garbage collection using reference counting. Discuss whether or not they need to consider the possibility of a reference count overflowing when incremented. [4 marks]

(d) Suppose we are writing a compiler for an ML-like language. We want to employ the equation

\[(\text{map } f) \circ (\text{map } g) = \text{map } (f \circ g)\]

as a left-to-right rewrite rule for optimisation. The symbol \(\circ\) represents function composition — for any value \(v\) the expression \((f \circ g) \ v\) evaluates to the value of \(f(g \ v)\).

Discuss the merits of this idea. Is it always correct? [8 marks]