COMPUTER SCIENCE TRIPOS Part IA – 2018 – Paper 1

1 Foundations of Computer Science (AM)

(*a*) Sets containing integers can be represented as `int list` values. Consider two such representations called *unordered* and *ordered*. In the former elements can appear in any order; in the latter elements are required to be in ascending order. In both representations elements must not be repeated.

(*i*) Using the *unordered* representation give ML functions corresponding to set intersection and set union. [3 marks]

(*ii*) For your answers to Part (*a*)(*i*) give the associated time complexities, assuming both input sets have at most n elements. [2 marks]

(*iii*) Now, using the *ordered* representation, give ML functions corresponding to set intersection and set union along with their time complexities, noting reasons for any differences in complexity compared to those for the *unordered* representations. [4 marks]

(*iv*) Without giving any ML code, suggest a technique whereby set intersection for *unordered* can be implemented in $O(n \log n)$ time. [1 mark]

(*b*) One often hears "in ML, all functions take exactly one argument". Explain *two* techniques which enable us to circumvent this rule, illustrating your answer by giving ML definitions for the standard `map` and a variant which "takes the same arguments in the same order".
Call the variant `map'`. [3 marks]

(*c*) For each of the five following ML expressions give definitions of `f`, `g`, `h`, `xs` `ys` and `zs` (as appropriate) which cause the expression to evaluate to `true`, or explain, giving reasons, why this is impossible. (The functions `map` and `map'` are as discussed in Part (*b*).)

(*i*) `map f [1,2] = [[1,2],[3,4]]`

(*ii*) `map g [1,2,3,4] = [[1,2],[3,4]]`

(*iii*) `map (map h) xs = [[1,2],[3,4]]`

(*iv*) `map map' ys = [[1,2],[3,4]]`

(*v*) `map map zs = [[1,2],[3,4]]`

[7 marks]