COMPUTER SCIENCE TRIPOS Part II – 2016 – Paper 9

9 Optimising Compilers (AM)

(a) Explain the scenario in which a strictness analyser is used to optimise a program. Your answer should consider the following: for what languages strictness optimisation is useful, where it is beneficial to be able to place strict or non-strict annotations on a program (seeing the strictness analyser as a black-box oracle), and how such annotations can safely allow an optimiser to represent strict or non-strict values differently at run time. Give an example program which has different run-time space complexity before and after strictness optimisation. [5 marks]

(b) One implementation for a strictness analyser determines *strictness functions* associated with each user-defined or built-in function. Given a user-defined function taking n integer arguments to an integer result, state the domain and range of its associated strictness function. How can such a strictness function be used to produce the strict or non-strict annotations in Part (a)? [3 marks]

(c) Give a data structure suitable for representing strictness functions within a strictness analyser. Can ordinary functions be used? Would your data structure represent strictness functions $\lambda(x, y, z).\ x \wedge (y \vee z)$ and $\lambda(x, y, z).\ (x \wedge y) \vee (x \wedge z)$ differently? Would these two strictness functions enable different strictness optimisations in Part (a)? [4 marks]

(d) Give the strictness functions for the following source-language functions.

(i) The built-in addition and 3-argument conditional functions. [2 marks]

(ii) A built-in *parallel-if* function, which evaluates all its three arguments in parallel, and returns a result as soon as enough of its arguments terminate. This includes returning value v when the second and third arguments evaluate to v even if the first argument is still computing. [2 marks]

(iii) The user-defined function `f` defined by

```
f(x,y,z,t,u) =
  if x=0 then y else f(x-1, t+2, u+3, y*4, z*5);
```

[4 marks]