COMPUTER SCIENCE TRIPOS Part II – 2024 – Paper 9

9 Optimising Compilers (tmj32)

Compilers for functional languages sometimes perform strictness analysis to pass parameters by value rather than by name.

(*a*) Define the concept of *strictness* and how it differs from *neededness*. [3 marks]

(*b*) Write functions with two parameters for each of the following cases.

(*i*) A function that is strict in both parameters but only the first is needed. [2 marks]

(*ii*) A function that is strict in its first parameter and only the first is needed. [2 marks]

(*iii*) A function that is strict in neither parameter and neither is needed. [1 mark]

(*c*) Perform strictness analysis on the following program to obtain its strictness function. In which parameter(s) is `f` strict?

```
f(a, b, c)
  = if a<1 then b elif a<2 then c else f(a-c, b, c);
```

You may use the following built-in strictness functions.

$$\begin{aligned} 1^\sharp &= 1 \\ 2^\sharp &= 1 \\ lt^\sharp(x,y) &= x \wedge y \\ sub^\sharp(x,y) &= x \wedge y \\ cond^\sharp(p,x,y) &= p \wedge (x \vee y) \end{aligned}$$

[6 marks]

(*d*) After strictness optimisation, some parameters remain passed by name, yet you wish to evaluate these as early as possible within the function whilst maintaining strictness properties. Describe an analysis to do this, and explain the results on the program in part (*c*). [6 marks]