2001

p89 7
AM

Optimising Compilation `optcomp8.tex`

Explain the ideas of strictness analysis, including over what languages the ideas are applicable and what transformations are enabled by it. Describe how strictness functions for (a) built-in and (b) user-defined functions are defined, clarifying the similarities and differences. [10 marks]

A language has a user-defined function f which is defined in terms of built-in functions $a_1, \cdots, a_t$ and possibly recursion. Later, to aid efficiency, and additional function a_{t+1} is added to the set of system functions, but its effect (semantics) is the same as that of f. By considering examples similar to those used to show analyses are safe but imprecise, or otherwise, determine a relationship between the strictness functions $f^\sharp$ and $a^\sharp_{t+1}$. [5 marks]

It is noted that strictness functions, e.g.

$$cond^\sharp(x, y, z) = x \wedge (y \vee z)$$

do not generally use negation in their defining boolean expressions. Show that all strictness functions can be written without negation or find a counter-example. Hint: no computable function f can have semantics such that there are x and y which satisfy

$$f(x, y) = \perp \text{ and } f(x, \perp) \neq \perp.$$

[5 marks]

Solution Notes

Part 1: bookwork. Letting $2 = \{0, 1\}$ then the strictness space for a k-adic function is $2^k \to 2$.

$$plus^\#(x, y) = x \wedge y$$

$$cond^\#(p, x, y) = p \wedge (x \vee y)$$

f is strict in ith arg if $f^\#(1, ...1, 0, 1, ..., 1) = 0$. If f is strict in ith argument it can be implemented using CBV.

Part 2:

$$f^\sharp(x_1, \ldots, x_k) \leq a^\sharp_{t+1}(x_1, \ldots, x_k)$$

Part 3: True since f is monotic w.r.t $\perp \leq x$ then $f^\sharp$ is monotonic w.r.t $0 \leq 1$.