**SOLUTION NOTES**

## Optimising Compilers 2003 Paper 8 Question 7 (AM)

(a) bookwork

(b) For each function $f$, whenever its $i$th argument is determined to be strict by the analysis, i.e. $f^\sharp(1,\ldots,1,0,1,\ldots,1)$ then at run-time all calls $f(e_1,\ldots,e_k)$ which return will have evaluated $e_i$ before return.

(c) This can be done modelling functions-as-functions as for strictness so that $T$ represents "value may be list containing part of argument" $F$ represents "no part of argument may be returned". Thus an abstract *escape function* $f^\sharp$ will have that $f^\sharp(F,\ldots,F,T,F,\ldots,F)=T$ implies that the $i$th argument (may) escape at run-time, $F$ means definitely does not escape.

$$\begin{aligned}
\mathit{cond}^\sharp(x,y,z) &= y \vee z \\
+^\sharp(x,y) &= F \\
\mathit{hd}^\sharp(x) &= F \\
\mathit{tl}^\sharp(x) &= x \\
\mathit{cons}^\sharp(x,y) &= y
\end{aligned}$$

[Experts note: because there is no need for $\wedge$ as well as $\vee$ abstract representation of functions could be simplified to subsets of arguments (think of CNF), thus e.g. $\mathit{cond}^\sharp = \{2,3\}$, $+^\sharp = \{\}$, This is also an acceptable solution, but really should have a definition of what it means to compose functions (in order to get the abstract meaning of an expression representing the body of a function).]

(d)

$$\begin{aligned}
f^\sharp(x,y,z) &= y \vee z \\
g^\sharp(x,y) &= F \\
h^\sharp(x,y) &= x \\
k^\sharp(x,y) &= y
\end{aligned}$$