

Paper 6 Question 10

ACN – Foundations of Func. Programming

9 Foundations of Functional Programming 2

Show how to use pure lambda terms to represent

(a) truth values and the ability to make tests on them;

(b) tuples; *cons car cdr clarity*(c) something like the effect of the ML datatype `blist = empty | node of bool*blist`;(d) a fixed-point operator; *def, lam, justify* *nil cons isnil car/cdr*

(e) code to reverse a list of boolean values.

code

[4 each]

9.1 Marking notes

Syllabus Part A section 3.

I will write these in ML/ISWIM style notation but can easily expand to lambda terms as needed!

1. `fun true a b = a; fun false a b = b; fun if b p q = b p q;`
2. `fun pair a b f = f a b; fun first p = p true; fun second p = p false;`
3. `empty = pair false (anything); fun node a b = pair true (pair a b);`
The testing for empty and taking head and tail are all easy!
4. `Y f = let g x = f (x x) in g g` as per notes/lectures.
5. `rev a = rev1 a empty; rev1 a b = if empty(a) then b else rev1 (tail a) (node (head a) b);` now we have all the lambda terms needed to do this easily!

END