An Introduction to Functional Programming Through Lambda Calculus

Greg Michaelson

Errata

Preface to Dover Edition

piii Ruby ==> Ruby and Scala

piv New calculi building on l calculus ==> New calculi building on λ calculus

Original book

|  |  |  |  |
| --- | --- | --- | --- |
| page | line/ exercise | old | new |
| 10 | 25 | *consistent* | **consistent** |
| 10 | 26 | *complete* | **complete** |
| 10 | 28 | describe arithmetic | describe arithmetic consistently |
| 11 | 4 | Turing machines, recursive functions | Turing machines, and recursive functions |
| 37 | 8 | == | ==> |
| 37 | 16 | == | ==> |
| 47 | 2.1 (e) | (λp.(λq.p λx.(x p)) λi. λj.(j i)) | ((λp.(λq.p λx.(x p)) λi. λj.(j i)) λa. λb.(a (a b))) |
| 59 | 2 | == | <deleted> |
| 59 | -12 | def pred = λn.((iszero n) n (n select\_second)) | def pred = λn.((iszero n) zero (n select\_second)) |
| 71 | 10 | else f f x y | else f f (succ x) (pred y) |
| 94 | 11 | errortype | error\_type |
| 112 | 8 | Static and dynamic type checking | Static type checking |
| 142 | 15 | but must be made type specific | but INSERT must be made type specific |
| 144 | -15 | or IDELETE 0 (H::T) = H | or IDELETE 0 (H::T) = T |
| 151 | 2 | “<character> <characters>” == <character>::”<characters>” | “<character> <characters>” == ‘<character>’::”<characters>” |
| 166 | -15 | Initially, FLIST and LIST | Initially, FLIST and SLIST |
| 169 | -2 | TADD I (NODE NI L R) = NODE NI L (TADD I L) | TADD I (NODE NI L R) = NODE NI L (TADD I R) |
| 170 | 7 | ELSE NODE NITEM L (TADD I R) | ELSE NODE NI L (TADD I R) |
| 186 | -1 | EVAL [[3,”\*”,4],”-“,[5,”+”,6]] => ... => 11 | EVAL [[3,”\*”,4],”-“,[5,”+”,6]] => ... => 1 |
| 200 | -16 | THEN (SUCC 1) | THEN (SUCC 2) |
| 201 | 23 | 0 | 4 |
| 239 | 9.3 (e) | Use smerge from (iv) above | Use smerge from (d) above |
| 258 | 17 | (cond ((null tree) (node v nil nil)) | (cond ((null tree) (node i nil nil)) |
| 261 | 19 | approach to objects | approach to functions |
| 280 | 3.2 | def equiv = λx.λy.(x y (not y) | def equiv = λx.λy.(x y (not y)) |
| 292 | (g) | IF STARTS L2 L3 | IF STARTS L1 L3 |
| 292 | (g) | THEN APPEND L1 (REMOVE L2 L3) | THEN APPEND L2 (REMOVE L1 L3) |
| 305 | -1 | on and Hope | on Hope |

With thanks to everyone who helped identify these.