PRIFYSGOL CYMRU; UNIVERSITY OF WALES

DEGREE EXAMINATIONS JANUARY 2003

SWANSEA

Computer Science

CS 221 Functional Programming I

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University

Students are NOT permitted to use calculators

CS_221
FUNCTIONAL PROGRAMMING 1

(Attempt 2 questions out of 3)

Question 1

a. State the α, β and η conversion rules for the λ-calculus. Show that the η rule can be demonstrated to be correct by applying the β rule.
For each of the following λ-expressions identify all β and η redexes indicating which is the leftmost redex.
Hence reduce the expressions to normal form.

i) $(\lambda x.((\lambda z.zx)(\lambda x.x)))y$
ii) $(\lambda x.((\lambda y.xy)z))(\lambda x.xy)$
iii) $\lambda p.(((\lambda q.qr)s)p)$

[17 marks]

b. Give the equivalent of the α, β and η conversion rules as they might apply in Gofer. Assuming that a Gofer function `adder` has been defined as below simplify the following Gofer fragments stating which rule you are using.

```
adder x y = x + y
```

i) `double = adder 7 7`
ii) `increment x = adder 1 x`

Derive the types of these three functions.

[8 marks]

Question 2

Consider the following definitions:-

```
reverse []          = []
reverse (x:xs)      = append (reverse xs) [x]

append [] ys        = ys
append (x:xs) ys    = x : (append xs ys)

shunt xs []         = xs
shunt ys (x:xs)     = shunt (x:ys) xs

rev xs              = shunt [] xs

foldr f u []        = u
foldr f u (x:xs)    = f x (foldr f u xs)

foldl f u []        = u
foldl f u (x:xs)    = foldl f (f u x) xs
```

a. Prove the following

i) `append xs (append ys zs) = append (append xs ys) zs`
ii) `reverse (append xs [x]) = x : (reverse xs)`
iii) `reverse (reverse xs) = xs`
iv) `shunt yx xs = append (reverse xs) ys`
v) `rev xs = reverse xs`

[15 marks]

b. The efficiency of a function can be expressed in terms of time efficiency (by counting the number of reduction/rewrite steps required), and space efficiency (by considering the length of the longest intermediate expression produced during reduction/rewriting).

Discuss the time and space efficiency of the following pairs of expressions

i) `rev [1,2,3]` and `reverse [1,2,3]`
ii) `foldr (*) 1 [1,2,3]` and `foldl (*) 1 [1,2,3]`

[10 marks]

Question 3

a. Explain the differences between the following definitions of a set datatype and discuss the advantages and disadvantages of each approach.

i) `type Set1 a = [a]`
ii) `data Set2 a = Mkset [a]`
iii) `data Set3 a = Emptyset | AddElt a (Set3 a)`
iv) `data Set4 a = Iset (a->Bool)`

[10 marks]

b. Define (if possible) versions of the following functions to work with each definition of a set.

i)	empty	- to create an empty set
ii)	member	- to test if a given element is a member of a given set
iii)	addelt	- to add an element to a set
iv)	union	- to define the union of two sets
v)	intersect	- to define the intersection of two sets
vi)	eqset	- to determine if two sets are equal
vii)	complement	- returns the set of all elements not in a set

[15 marks]