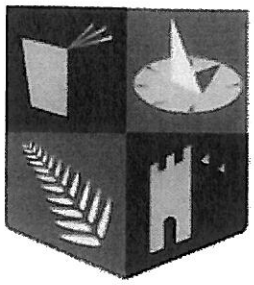

Maynooth University
National University of Ireland Maynooth

OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

JANUARY 2015 EXAMINATION

CS424

Programming Language Design & Language Semantics

Dr. D. Charles, Dr. A. Winstanley, Prof. B. Pearlmutter

Time allowed: 2 hours

Answer ***four*** questions

All questions carry equal marks

[25 marks]

1 Define a Scheme function after-filter which takes a predicate p and a list xs and returns a list of those elements of xs which immediately follow an element which passes the predicate p.

Examples:

```
(after-filter number? '(a b 2 3 c 4 d))
=> (3 c d)

(after-filter symbol? '(a b 2 3 c 4 d))
=> (b 2 4)

(after-filter symbol? '())
=> ()
```

[25 marks]

2 Define afterFilter in Haskell, with the same convention as the above Scheme function. Be sure to give it an appropriate type signature.

Examples:

```
> afterFilter (<0) [-4,7,-4,-8,3,-3,-6,0,-9,-1]
[7,-8,3 ,-6,0 ,-1]

> afterFilter (=='f') "fifferfefferfather"
"ifeefea"

> afterFilter (>0) []
[]
```

[25 marks]

3 In the simply typed lambda calculus, with basis elements

```
0:R
1:R
plus:R->R->R
```

is the expression

```
(Î» f:T . Î» g:U . (f 1) (g 0)) (plus 1) (plus (plus 1 1))
```

well typed, given appropriate types filled in for T and U? Explain why or why not.

4 In Prolog, define a predicate doublemember/2 which takes a potential element and a list and is true when the element appears (at least) twice in the list. **[25 marks]**

Examples:

```
?- doublemember(a,[the,quick,brown,fox]).
no

?- doublemember(a,[])
no

?- doublemember(a,[a])
no

?- doublemember(a,[a])
yes

?- doublemember(a,[the,quick,a,brown,X,fox]).
X=a
```

5 In the untyped lambda calculus, give a Church encoding for triples, by defining the following: **[25 marks]**

```
triple = Î» x . Î» y . Î» z . ???
fst = Î» t . ???
snd = Î» t . ???
thd = Î» t . ???
```

(i.e., fill in the missing ???s) which obey the algebraic properties

```
fst (triple A B C) = A
snd (triple A B C) = B
thd (triple A B C) = C
```

If these are regarded as being System F expressions (aka,the polymorphic typed lambda calculus) give the types of triple, fst, snd, and thd.