NUI MAYNOOTH

Ollscoil na hÉireann Má Nuad

OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

AUTUMN 2014 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

1 (Scheme) [25 marks]

Define a function add-numbers which takes an s-expression and returns the sum of all the numbers contained therein. E.g.,

```
(add-numbers 17)              => 17
(add-numbers '(a (1 (2) 3) 4))  => 10
(add-numbers '(the quick fox))  => 0
```

Note: the Scheme predicate "number?" can be used to test if a given value is a number.

2 (Haskell) [25 marks]

Define a function

```
sort :: (a -> a -> Bool) -> [a] -> [a]
```

which sorts a list according to the given predicate.

3 (Prolog) [25 marks]

Define a predicate sibling/3 which takes a list of parent-child pairs and two names and is true if the two are siblings. E.g.,

```
sibling([[fred,sam],[fred,sue]], sam, sue)
```

would be true, while

```
sibling([[fred,sam],[fred,sue]], sam, fred)
```

would be false.

4 (Lambda Calculus) [25 marks]

Give a valid untyped lambda calculus expression which cannot be given a type in the simply typed lambda calculus, and explain why this is the case.

5 (Scheme/Haskell) [25 marks]

Consider the following Scheme function definition

```
(define foo
  (lambda (f xs ys)
    (if (null? xs)
        xs
      (cons (f (car xs))
            (foo f ys (cdr xs))))))
```

(a) Describe what foo does, and give some examples.

(b) Translate foo into Haskell, and give a type declaration for foo.