Maynooth University
National University of Ireland Maynooth

OLLSCOIL NA hÉIREANN MÁ NUAD

THE NATIONAL UNIVERSITY OF IRELAND MAYNOOTH

AUTUMN 2017 EXAMINATION

CS424

Programming Language Design & Language Semantics

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

Time allowed: 2 hours

Answer at least four questions
Your mark will be based on your best ***four*** answers

All questions carry equal marks

Q1: Scheme **[25 marks]**

Define a Scheme function `foo` that takes two lists and yields a list combining all the elements in the two input lists, taking 1 from the first list, 2 from the second list, 3 from the first list, 4 from the second list, etc, until both are exhausted.

Examples:

```
(foo '(a b c d e f g) '(aa bb cc dd ee ff gg))
  => (a aa bb b c d cc dd ee ff e f g gg)

(foo '(a b c d e f g) '())
  => (a b c d e f g)

(foo '() '(aa bb cc dd ee ff gg))
  => (aa bb cc dd ee ff gg)
```

Q2: Haskell **[25 marks]**

Define a Haskell function `foo`, including a type signature, that takes two lists and yields a list combining all the elements in the two input lists, taking 1 from the first list, 2 from the second list, 3 from the first list, 4 from the second list, etc,until both are exhausted.

Examples:

```
foo [1,2,3,4,5,6,7,8] [11,12,13,14,15,16,17,18]
  => [1,11,12,2,3,4,13,14,15,16,5,6,7,8,17,18]
```

Q3: Prolog [25 marks]

Define a Prolog predicate `thrice/2` which is true when its first argument appears three times in its second argument, which must be a list.

Examples:

```
?- thrice(e,[t,h,e,b,e,a,t,b,e]).
yes

?- thrice(e,[t,h,e,b,e,a,t,b]).
no
```

Q4: Lambda Calculus [25 marks]

In the simply typed lambda calculus, why does the formal parameter of a lambda expression require a type (in the formal syntax) but the body does not?

Q5: [25 marks]

Give an example of a legal reasonable Scheme program which cannot be easily translated into Haskell because the types won't work.