

# Foundations of Computer Science – Problem Sheet 4

Heidi Howard

November 2, 2014

All of these questions are real past exam questions, you can find the questions at <http://www.cl.cam.ac.uk/teaching/exams/pastpapers>. This supervision will cover the material from lectures 10 to 12 and review the materials from the rest of the course. Limit yourselves to around 30 mins per question to practice your timings for exams.

## Exercise 1:

2002, Paper 1 Question 5 - for those not familiar with foldr, is is defined and used as follows:

```
(* given a function f, a list and accumulator, foldr recursively
   applies the head of the list and accumulator to f *)
fun foldr f ([], e) = e
  | foldr f (x::xs, e) = f(x, foldr f (xs,e))

fun sum xs = foldr (fn (x,acc) => x+acc) (xs,0)

(* an example call trace for sum
   assume f means (fn (x,acc) => x+acc)
   sum [3,5] =>
   foldr (f ([3,5],0)) =>
   f (3, foldr f ([5],0))) =>
   f (3, (5, foldr f ([],0))) =>
   f (3, (5, 0)) =>
   f (3,5) =>
   8
   *)
```

## Exercise 2:

2005, Paper 1 Question 1 - covering representing polynomials

## Exercise 3:

2006, Paper 1 Question 5 - covering queues

## Exercise 4:

2008, Paper 1 Question 6 - covering searching

## Exercise 5:

2006, Paper 1 Question 6 - covering mutability and reviewing previous materials