

Advances in Programming Languages

2016–2017 Semester 1

Selected Past Exam Questions

From 2014/15 Question 1

1. This question is about concurrent programming with threads.

The following Java is from a class written to represent a person's name. Some of the code is to be used in concurrent programming.

// Class to represent a person's name, made up of their first name and last name.

```
public class FullName {  
  
    private String first = "";  
    private String last = "";  
  
    // Operation to copy the contents of one fullname into another  
    public static void copy(FullName p, FullName q) {  
        q.first = p.first;  
        q.last = p.last;  
    }  
  
    // Wrapper around the copy operation  
    public static void safe_copy(FullName p, FullName q) {  
        synchronized(p){ // Claim first fullname p  
            synchronized(q){ // Claim second fullname q  
                copy(p,q); // Copy across contents  
            }  
        }  
    }  
  
    // Remainder of class omitted  
}
```

- (a) Describe what it means for methods in Java to be *thread safe*. [1 mark]
- (b) The `copy` method is not thread safe. Explain why, showing fragments of code and their execution to demonstrate how this can be a problem in practice. [6 marks]
- (c) The method `safe_copy` is a wrapper around `copy` that is intended to be thread safe. However, it is still problematic for use in concurrent code, as it may cause *deadlock*.
- (i) Describe what it means for threaded concurrent code to *deadlock*. [2 marks]
- (ii) Explain why `safe_copy` may cause deadlock, and give code fragments demonstrating how this can be a problem in practice. [6 marks]

2. This question is about types and type systems.

(a) The following are three variations on the idea of *polymorphism* in programming languages.

- (i) Subtype polymorphism.
- (ii) Parametric polymorphism.
- (iii) Ad-hoc polymorphism.

For each of these give a brief explanation of what it is, and give an example. [6 marks]

Note: Each example can be in any programming language — Haskell, Java, Scala, or whatever you think appropriate — but you must say which language it is. You can use different languages for each example if you think that will help your explanations.

(b) Suppose we have a dependently-typed lambda calculus which includes types *Int* of integers, *Num* of non-negative integers, and *Matrix* $n\ m$ of integer matrices with n rows and m columns, for $n, m : \text{Num}$. One possible operation in the language is to generate an identity matrix:

$$\textit{identity} : \forall n : \text{Num}. \textit{Matrix}\ n\ n .$$

- (i) Give a suitable dependent type for the operation of matrix addition *add*.
- (ii) Give a suitable dependent type for matrix multiplication *mult*.
- (iii) Use some or all of *identity*, *add*, and *mult* to write out a term that computes the 5×5 matrix that has leading diagonal elements all 2 and zero elsewhere (i.e. double the identity matrix). [5 marks]