**UNIVERSITY OF EDINBURGH**

**FACULTY OF SCIENCE AND ENGINEERING**

**MASTER OF SCIENCE IN COMPUTER SCIENCE**

# PROGRAMMING PARALLEL COMPUTERS

**Tuesday 25 April 2000**

**9:30 to 11:00**

**Examiners:**

**I. Watson (External)**

**R.N. Ibbett (Chairman)**

**INSTRUCTIONS TO CANDIDATES**

**Answer any TWO questions.**

**All questions carry equal weight.**

1. (a) Explain why message-passing is necessary for languages supporting private address spaces.

*(2 marks)*

(b) What are the restrictions on message type in message passing languages? Illustrate your answer with examples and describe to what extent type checking is supported in each case.

*(6 marks)*

(c) Describe the relative benefits of synchronous and asynchronous communication.

*(3 marks)*

(d) Describe, with illustrative examples, how asynchronous message-passing can be implemented in (i) MPI (ii) Occam (iii) Poly-ML.

*(7 marks)*

(e) Message-passing provides a natural means of supporting pipeline parallelism. Describe to what extent pipeline parallelism can be supported in the following non-message-passing languages:

(i) CC++ (ii) Linda (iii) C*

*(7 marks)*

2. (a) Describe and contrast the thread creation primitives in CC++, Sequent C and Posix threads.

   *(6 marks)*

   (b) Why do certain languages with a shared address space require synchronisation primitives?

   *(3 marks)*

   (c) Describe and contrast the mechanisms in CC++ and Sequent C to

      i. prevent concurrent execution of critical sections of code.
      ii. guarantee a variable is written to before it is read

   *(6 marks)*

   (d) How is correct synchronisation in data parallel languages such as C* and HPF managed?

   *(5 marks)*

   (e) How are concurrent threads or processes synchronised in Linda and OpenMP?

   *(5 marks)*

3. (a) Critically compare and contrast the parallelisation approaches supported by OpenMP and HPF. To what extent is work and data layout explicit in each case?

   *(7 marks)*

   (b) Compare OpenMP to MPI in terms of

      i. program development time
      ii. attainable execution time.

   *(3 marks)*

   (c) How do work and data layout affect (i) parallelism (ii) load balance (iii) synchronisation (iv) communication?

   *(5 marks)*

   (d) How can program transformations be used to discover parallelism? Illustrate your answer with several examples and describe whether or not the resulting parallelism can be efficiently expressed in either OpenMP or HPF.

   *(10 marks)*