CO3007, CO4007, CO7007

Communication and Concurrency

Irek Ulidowski

E-mail: I.Ulidowski@le.ac.uk
(or simply iu3@le.ac.uk)

Office: G12, Computer Science Building

Research interests

1. Process Algebras, for example **CCS**, CSP, π-calculus or ambient calculus
2. **Structural Operational Semantics**, SOS for short
3. **Reversible Computation**: reversing process calculi for biochemistry, reversing multi-threaded programs for debugging

Co-organiser of CONCUR 2012 in Newcastle upon Tyne.

Established Reversible Computation Conference.

Chair of COST Action on Reversible Computation.

PhD positions and GTA positions for students interested in research are available.

Introduction

- Organisation
- Reading List
- Module Assessment
- Concurrent and Communicating Systems: definition, examples
- Aims and Objectives

Organisation

- Around 30 lectures:
  - Tuesday 14:00 - 16:00 BEN F75a
  - Friday 11:00 - 13:00 ATT LG03
- At most 10 surgeries/problem classes, Tuesday 11:00 in ATT 802

Note:

- Students should attend the lectures.
- Surgeries/problem classes are devoted to solving problems from the worksheets and the lecture notes, and to **feedback** on assessed work.
- Surgeries/problem classes are compulsory.

Reading List

- **Main Textbooks**:
  - Lecture Notes
  - Robin Milner, Communication and Concurrency, Prentice Hall
- Other Books:
  - C.A.R. Hoare, Communicating Sequential processes, Prentice Hall
  - C. Fencott, Formal Methods for Concurrency, Thomson Computer Press
  - A.W. Roscoe, The Theory and Practice of Concurrency, Prentice Hall

Module Assessment

Coursework: 40% of the overall module mark

- The first two CWs will be by small group projects.
- There will be one class test and one on-going individual assessed worksheet.
- Unexcused absence in a test results in a zero mark. Can use MCFs supported by evidence.
- There is a provision for re-siting coursework and exam.
- Avoid plagiarism.

Examination: 60% of the mark

- 3 hour examination in January. Four questions, full marks for four questions.

More details can be found in the Study Guide.

Many past exam paper questions are included in the lecture notes. Also available on the Web pages for CO3007.

Questions from past exam papers may also appear as coursework questions.

Overall: Intellectually stimulating, widens understanding of the challenges of concurrency and distributed systems, prepares for CO3090, useful for further study and PhD research.

Concurrent and Communicating Systems

A concurrent system is a dynamic system that is composed of several parts:

- each part acts concurrently with, and independently of, other parts—**concurrency**,
- parts interact (or communicate, react) with each other to synchronise their behaviour and exchange information—**communication**

Communication and concurrency are complementary notions, which

- are essential in understanding concurrent systems
- distinguish concurrent systems from sequential ones

Concurrent systems = communicating systems = concurrent communicating systems = reactive systems = interacting systems

Concurrent systems are complex in nature. Safety-critical systems, telecommunication systems and industrial plant systems are examples of concurrent systems. Thus their correctness, reliability and safety are crucial.

Examples

1. A university as a system of departments; and a department is as a system of people.

2. A chocolate vending machine system consisting of the machine and its customers.

3. A multi-threaded program with $n \geq 2$ threads.

4. Office computer network.

5. Mobile phone network.

And many others . . . .

Aims and Objectives

We will study a formalism called a Calculus of Communicating Systems, **CCS** for short, in order to

- understand better the essential aspects of concurrent and communicating systems,
- learn how to formally **specify** such systems,
- learn how to express how systems are constructed, or write down possible **designs** or **implementations** of systems,
- learn how to **verify** the correctness of a design (or an implementation) with respect to a specification,
- apply CCS to realistic (small) concurrent and communicating systems.

What is CCS?

- A process language for writing specifications and representing possible designs or implementations of concurrent systems at a relatively high level of abstraction.
- There are several theories that support CCS, and verification and testing within CCS
  - Structured Operational Semantics with the notions of bisimulations,
  - Axiomatic semantics with sets of laws (axioms) and rules of equational reasoning.
- There are logics in which we can express properties of concurrent systems (such as deadlock freedom). Also, there are denotational semantics. And there are testing techniques which could be used to check if properties are satisfied.