Informatics 1 - Computation and Logic (INFR08012)

http://www.drps.ed.ac.uk/16-17/dpt/cxinfr08012.htm

Current learning outcomes:

On completion of this course, the student will be able to:

1. Design a small finite-state system to describe, control or realise some behaviour, and evaluate the quality of such designs using standard engineering approaches
2. Apply the algebra of finite automata to design systems and to solve simple problems on creating acceptors for particular languages
3. Used propositional logic to describe simple problems, and truth table methods to determine truth values for given propositions
4. Apply a system of proof rules to prove simple propositional theorems
5. Describe the range of systems to which finite-state systems and propositional logic are applicable and be able to use the meta theory to demonstrate the limitations of these approaches in concrete situations

Proposed learning outcomes:

On completion of this course, the student will be able to:

1. Use sets, functions, and relations, to create a simple mathematical model of a real-world situation.
2. Use the syntax and semantics of propositional logic to express and solve simple sets of constraints.
3. Formalise simple propositional reasoning using various methods, including truth tables, Boolean algebra, and deduction.
4. Use regular expressions to search for simple patterns.
5. Understand how regular expressions link the structure and behaviour of finite automata, and design finite state acceptors for particular languages.
6. Use basic meta theory to discuss the applicability and limitations of these approaches.

I also propose the following minor amendments to Summary and Course Description

Summary

> The goal of this strand is to introduce the notions of computation and specification using finite-state systems and propositional logic. Finite state machines provide a simple model of computation that is widely used, has an interesting meta-theory and has immediate application in a range of situations. They are used as basic computational models across the whole of Informatics and at the same time are used successfully in many widely used applications and components. Propositional

logic, similarly is the first step in understanding logic which is an essential element of the specification of Informatics systems and their properties.

The goal of this course is to introduce the notions of computation and specification using finite-state systems and propositional logic. These provide examples of the logical ideas of syntax and semantics and the computational ideas of structure and behaviour. Finite state machines provide a simple model of computation that is widely used, has an interesting meta-theory and has immediate application in a range of situations. They are used as basic computational models across the whole of Informatics and at the same time are used successfully in many widely used applications and components. Propositional logic, similarly is the first step in understanding logic which is an essential element of the specification of Informatics systems and their properties.

> Finite-state systems as a basic model of computation: deterministic and non-deterministic automata; transducers; acceptors; structured design of finite state machines. Propositional logic: truth tables; natural deduction; resolution; elementary temporal logic. Introduction to software IP, and notions of verification, correctness, best practice and liability.

> Relevant QAA Computing Curriculum Sections: computer based systems, theoretical computing

The use of sets, functions, and relations to describe models of logic and computation. Finite-state systems as a basic model of computation: deterministic and non-deterministic automata; regular expressions; acceptors; structured design of finite state machines. Propositional logic: truth tables; natural deduction; resolution; satisfiability. Introduction to software IP, and notions of verification, correctness, best practice and liability.

Relevant ACM Computing Curriculum Knowledge Areas: Computational Science ● Discrete Structures; Systems design methodologies; Systems analysis & design.

Michael Fourman FRSE UKCRC FBCS

Professor of Computer Systems

School of Informatics

Informatics Forum

10 Crichton Street

Edinburgh EH8 9AB

michael.fourman@ed.ac.uk