Algebraic and Coalgebraic Methods in Software Development

MAP-i — 2017-18

Summary

This document describes a proposal for a course on Fundations of Computing to be offered in the 2017-18 MAP-i edition. The proposal is supported by a team from Minho University (Dep. of Informatics) and Aveiro University (Dep. of Mathematics).

1 Context and Objectives

An increasing number of computer based systems rely on the cooperation of distributed, heterogeneous components or services organised into open software architectures that, moreover, can survive in loosely-coupled environments and be easily adapted to changing application requirements. Such is the case, for example, of applications designed to take advantage of the increased computational power provided by massively parallel systems or of the whole business of Internet-based software development. In order to develop such systems in a systematic way, the focus of development methods has switched, along the last decade, from functional to architectural issues: both data and processes are encapsulated into software units which are connected into large systems resorting to a number of techniques intended to support reusability and modifiability. This encapsulation principle is essential to both the object-oriented and the more recent component-based software engineering paradigms.

This entails the need for semantic techniques able to cope either with date structuring and prescription of functionality, as well as with specification and analysis of (externally observable) behaviour. Both qualitative and quantitative, i.e. QoS-related, aspects of software have to be taken into consideration.

If on data-intensive applications the main element to tackle is the structure of information and its transformations, in dynamic, reactive computing the focus is placed on system’s behaviours and their interactions. Quoting Robin Milner, in his Turing Award Lecture, computing science has become a structural theory of interaction: Thus software, from being a prescription for how to do something — in Turing’s terms a ”list of instructions” — becomes much more akin to a description of behaviour, not only programmed on a computer, but occurring by hap or design inside or outside it.

Both initial algebras and final coalgebras provide abstract descriptions of a variety of phenomena in programming, in particular of data and behavioural structures, respectively. As universal properties, they both entail definitional and proof principles, i.e., a basis for the development of program calculi directly based on (actually driven by) type specifications. Moreover, such properties can be turned into programming combinators and used, not only to calculate programs, but also to program with. In functional programming the role of such universals has been fundamental to a whole discipline of algorithm derivation and transformation. On the other hand, coalgebraic modelling of dynamical systems and reasoning by coinduction has recently emerged as active area of research.

This course explores the role of such algebraic and coalgebraic structures, and corresponding logics, in program development. As expected, initial algebras turn out to be inductive data types, i.e., abstract descriptions of data structures. Dually, final coalgebras entail a notion of coinductive,
behaviour types, representing the dynamics of systems. Therefore, the course will cover the core ideas, techniques and results in

- Algebraic specification, induction and equational logic
- Coalgebraic specification, coinduction and modal logic for coalgebras

The complexity of real systems entails, sometimes, the combination of both aspects in the same formalism, for instance, using structured enriched modal logics. The modulo

- Modal logics for specification – a generic, building oriented, approach

does the generation of specific modal logics through a systematic process, so that their models and proof theory (and, consequently, the corresponding tool support) can be characterised in a generic, parametric way. Each application domain and/or computational paradigm at hand induces a parameter to a base logic able to talk about its main features and express meaningful properties (e.g., if the system reacts deterministically or not; if it interacts synchronously or asynchronously with physical devices, if it exhibits probabilistic behaviours; has weighted computations; concurrency, etc.). Build and use Modal logics “on-demand”, i.e., parametric on the specific constraints of each application domain, opposes the more common attitude favouring the definition of yet another logic in a more or less ad hoc way. This approach is being worked by the team along the last years and is on the basis of an ongoing FCT research project.

In software development, relations nicely capture nondeterminism and vagueness of requirements. Probabilistic functions go a step further by quantifying the likelihood of each possible, expected or faulty, behaviour. Such a shift from qualitative to quantitative reasoning, which is becoming pervasive in Computer Science, calls for a language able to accommodate both these aspects, while preserving the polymorphic and calculational style of functional and relational program derivation. Recently, typed linear algebra has been suggested as a suitable candidate for such a unifying role, when restricting to discrete probability spaces. A new module in the course, first offered in the 2013-14 edition, will explore this perspective and its applications, namely to the study of fault-propagation in software systems. It builds on very recent and exciting results on

- Linear algebra of programming and applications

To provide a common background to formulate and discuss the topics above, the course will also include a brief

- Introduction to category theory

The course will build a roadmap to the broad area of algebraic and coalgebraic methods in software development, not only by providing an introductory survey, but also by exposing students to cutting-edge research topics and open problems, object of four research projects recently and currently coordinated by the proponent team:

- Mondrian (PTDC/EIA-CCO/108302/2008) on Foundations for architectural design: Service certification, dynamic reconfiguration and self-adaptability,
- Qais (PTDC/EIA-CCO/122240/2010) on Quantitative analysis of interacting systems: foundations and algorithms,
- DAlí (POCI-01-0145-FEDER-016692) on Dynamic logics for cyberphysical systems: towards contract based design

Motivated students will have the opportunity to discuss possible PhD topics in Foundations of Computing within the themes of this course. Funding opportunities may be available.
2 Learning outcomes

- Familiarity with the main topics, research questions and scientific challenges in the covered area (algebraic and coalgebraic methods);
- Ability to apply them to building and reasoning about, abstract models for software, its functionality, behaviour and composition.
- Ability to extract information from scientific papers in the area.
- Enhanced technical writing and presentation skills.

3 Pre-requisites

The course is almost self-contained, assuming only familiarity with elementary discrete mathematics at undergraduate level. Some previous experience on semantics of programming languages will help.

4 Format

Tutorial module, supported with demos and experimental lab work.

5 Grading

Assessment on base of an individual report on a research topic related with the course.

6 Course Contents

Plan

1. Introduction to category theory for computer science
   (a) Universal properties; categories; isomorphism; monomorphisms and epimorphisms.
   (b) Constructions in categories: duality, products, sums, limits and colimits.
   (c) Functors and natural transformations
   (d) The Yoneda lemma
   (e) Adjoint functors and adjoint functor theorems
   (f) Cartesian closed categories and λ-calculus
   (g) Monoidal categories

2. Algebras and algebraic specification
   (a) Signatures, models
   (b) Equational logic
   (c) Signature morphisms
   (d) Refinements
   (e) Introduction to the theory of institutions
   (f) Behavioural specifications

3. Coalgebras and coalgebraic specification
4. Modal logics for specification – a generic, building oriented, approach
   (a) Introduction to (classic) modal logics
   (b) Hybrid and dynamic variants
   (c) Parametric generation of (new) variants of Modal-Logics
   (d) Adding fuzziness to the picture
   (e) Illustrations and applications

5. Linear algebra of programming
   (a) Categories of matrices.
   (b) Probabilities and probabilistic functions.
   (c) Adjunction: probabilistic functions represented by column stochastic matrices. The ‘matrix-transform’.
   (d) Elementary probability theory encoded in linear algebra. Examples.
   (e) The distribution monad.
   (f) Applied linear algebra of programming: calculating fault propagation in mutually recursive functions.
   (g) Weighted automata and weighted coalgebraic systems.
   (h) Weighted bisimulation. Probabilistic behaviour.
   (i) LAoP for QoS calculation.

Textbooks and Reading Material

On category theory : [3, 43, 19, 2]
On algebraic specification : [13, 8, 42]
On coalgebraic modelling and coinduction : [41, 17, 18, 1]
On modal logics for specification – a generic, building oriented, approach : [9, 14, 24, 25]
On linear algebra of programming : [37, 20, 39]

7 Team

Alexandre Madeira (Coordinator) is a post-doc researcher at HASLab INESC TEC, currently co-
ordinating a FCT-funded research project on dynamic logic and contract-based programming. He
was a former MAP-i doctoral student. His PhD thesis [21] on hybrid logic and software reconfig-
uration was later awarded the IBM Scientific Prize for 2013. He has published more than twenty
papers in several journals and conferences over the past 5 years. Selected relevant publications
[33, 32, 22, 23, 12, 25, 24].

Manuel António Martins is Assistant Professor at the Department of Mathematics of Aveiro
University, and a researcher at the Center for Research and Development in Mathematics and
Applications. His research interests are related to Abstract Algebraic Logic (AAL) and Algebraic
Specification of abstract data types; namely on the application of tools and results of AAL to
the specification and verification of software systems. On this topic he has published 6 papers in international journals. He has supervised 4 MSc thesis (2 in the area of the current proposal) and is currently supervising 2 PhD projects (1 in the area of the current proposal). Selected relevant publications on specification and verification of software systems: [26, 31, 27, 28, 30, 29].

**Dirk Hofmann** is Assistant Professor with habilitation at Department of Mathematics at the University of Aveiro, and researcher at the Center for Research and Development in Mathematics and Applications. His main interests of research focus on the development and application of categorical methods in Mathematics, more specifically in algebra, topology and domain theory. On this topic he has published more than ten papers in several journals over the past 4 years. He has supervised 4 MSc thesis and is currently supervising 1 PhD project. Selected relevant publications on category theory: [16, 10, 11, 15].

**Luis Soares Barbosa** is Associate Professor, with tenure, at the Department of Informatics of Minho University, Adjunct Associate professor of the United Nations University and a researcher at CTC (area of Theory and Formal Methods). His research interests are related to program semantics and calculi applied to systems understanding and rigorous software construction. A particular application area concerns the development of formal models and calculi for software components, services and architectures. On this topic he has published over the past 4 years more than 15 papers in several journals and conferences. He has supervised 2 PhD thesis (1 in the area of the current proposal) and is currently supervising 5 PhD projects (2 in the area of the current proposal). e is also a member of IFIP WG 1.3 (Foundations of System Specification) and the Portuguese representative at IFIP TC1. Selected relevant publications on coalgebraic modelling and coinductive reasoning: [4, 5, 34, 40, 6, 35, 7, 30, 29].

**José N. Oliveira** is a full professor at the Computer Science Department of the University of Minho, Portugal, and a member of the High Assurance Software Laboratory (HASLab) of INESC TEC. He graduated in electrical engineering in 1978 from the University of Porto in Portugal and received the MSc and PhD degrees in computer science in 1980 and 1984, respectively, from the University of Manchester, United Kingdom. He is also a member of IFIP WG2.1 (Algorithmic Languages and Calculi) and of the Formal Methods Europe (FME) association. He has been working on formal methods since his PhD, with a recent interest in quantitative formal techniques relying on linear algebra and category theory. In the last 3 years (2011-2013) he published 6 journal papers and supervised 4 PhD theses on these topics. Selected relevant publications: [38, 37, 36, 20, 6, 39]

**References**


