Università degli Studi di L'Aquila - Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Course catalogue

Page compiled on 20/09/2019

Programme of Integrated course "Model-Driven Engineering and Formal Methods"		
This course is composed of 2 Modules: 1) Formal Methods, 2) Model Driven Engineering		
Programme of Module "Formal Methods"		
• Code: DT0202 • Type of course unit: • Level of course unit: • Semester: 2		
Number of ects credits: (Master Degree in Computer Science) 6 (workload 150 hours)		
Teachers: Monica Nesi (monica.nesi@univaq.it)		
1	**Course objectives**	The goal of this course is to introduce symbolic techniques for the specification and verification of systems properties based on equational reasoning and theorem proving. On successful completion of this course, the student should understand the basic notions of first-order rewriting and logic, and be able to reason on properties of terms by means of symbolic manipulation modulo an equational theory or the deduction rules of a given logic.
2	**Course content and learning outcomes (dublin descriptors)**	Topics of the module include: • Abstract reduction systems, normal form, convertibility. Confluence, Church-Rosser property and their equivalence. Local confluence, termination, canonicity. Principle of Noetherian Induction and Newman's lemma (proof included). • First-order terms, substitutions, match and mgu. Algorithm of syntactic unification. Equational theories and equational deduction. Term rewriting systems. Termination: reduction ordering, simplification ordering, recursive path ordering. • Confluence: overlapping of rewrite rules, critical pairs, Huet's lemma (proof included). The word problem, completion procedures, divergence of completion. E-unification of terms, narrowing relation, E-unification procedure based on normalized and basic narrowing. • Boolean formulae, satisfiability, tautology. Formulae in CNF and Davis-Putnam's algorithm. Natural deduction. Predicate logic: predicates, functions, variables, quantifiers,rules of natural deduction. Prenex DNF. • Introduction to higher-order logics and lambda-calculus. Untyped lambda-calculus, beta-reduction, simple type theory, type assignment calculus, polymorphism. On successful completion of this module, the student should : • have profound knowledge of the basic concepts of first-order rewriting and first-order logic, relate the termination and confluence properties, have knowledge and understanding of pattern matching, syntactic and semantic unification, have knowledge and understanding of the natural deduction rules for propositional logic and predicate calculus, understand lambda-calculus as the base of the syntax for higher-order logic and functional programming; • understand and apply definitions, inference rules and theorems; • analyse and discuss different variants of a concept, discuss different proof techniques for deriving properties of terms and formulae; • explain and illustrate the fundamental notions of unification of terms, reduction ordering and critical pairs, explain the word problem and the completion of equational theories; • demonstrate skill in equational reasoning, formal derivation and symbolic manipulation, demonstrate ability to derive types for higher-order terms and properties of terms and formulae, demonstrate capacity for building proofs.
3	**Course prerequisites**	Basic notions of mathematical logic and functional programming are helpful.
4	**Teaching methodsand**	Lectures and exercises **Language**: English

	language	**Reference textbooks** • L. Thery, ***Lectures notes***. *http://www-sop.inria.fr/marelle/Laurent.Thery/formal/* • J.-G. Smaus, ***Pearls of Computer-Supported Modeling and Reasoning - Lecture in l'Aquila***. *http://www.informatik.uni-freiburg.de/~ki/teaching/ws0910/csmr/aquila.html* • M. Nesi e M. Venturini Zilli, ***Sistemi di riduzione astratti. Research Report SI-98/06***. Facoltà di Scienze MM.FF.NN., Università degli Studi di Roma La Sapienza. 1998. *http://www.di.univaq.it/monica/MFI/NoteARS.pdf* • P. Inverardi, M. Nesi e M. Venturini Zilli, ***Sistemi di Riscrittura per Termini del Prim'Ordine***. Dipartimento di Matematica Pura e Applicata, Università degli Studi di L'Aquila. 1999. *http://www.di.univaq.it/monica/MFI/NoteSRT.pdf*
5	**Assessment methods**	Written and oral examination. The written exam can be split into a midterm exam + a final exam at the end of the course.

Programme of Module "Model Driven Engineering"

- Code: F0193
- Type of course unit: Elective (Master Degree in Computer Science curriculum SDRC), Elective (Master Degree in Computer Science curriculum ASSC), Compulsory (Master Degree in Computer Science curriculum GSEEM), Elective (Master Degree in Computer Science curriculum General)
- Level of course unit: Postgraduate Degrees
- Semester: 2

Number of ects credits: (Master Degree in Computer Science) 6 (workload 150 hours)

Teachers: Alfonso Pierantonio (Alfonso.Pierantonio@univaq.it)

1	**Course objectives**	LEARNING OUTCOME On successful completion of this module, students should be able to: * Knowledge Explain the principles and concepts underlying model-driven engineering Describe concept and approaches for defining the syntax and semantics of domain-specific modelling languages Define and explain the concepts, syntax and semantics of model transformation languages and mode-to-text tools Explain the basic concepts and techniques underlying the automated generation of (diagrammatic and textual) modelling editors and environments * Skills Use abstraction in the construction of software models and in the definition of domain-specific modelling languages Apply the EMF frameworks for model-driven engineering, including the definition of meta-models for domain-specific modelling languages Apply tools for model transformation and model-to-text generation Apply tools for model construction, model differencing and comparison, model management * Competence Assess the applicability and limitations of model-driven engineering and tools for development of software Judge the practical application of modelling and model management in realistic scenarios Discuss and document the construction and validation of models and extensions of supporting software tools
2	**Course content and learning outcomes (dublin descriptors)**	Topics of the module include: • Introduction, Metamodeling, General-purpose vs domain-specific modeling, Modeling languages (concrete vs abstract syntax), the metamodeling architecture, the Meta-Object Facility. • Eclipse EMF • Model Transformations: MOF Query-View-Transformation, ATL, JTL • Model management: Model weaving, Model differencing • Concrete Syntax: EMFText, GMF • Coupled Evolution: Metamodel/Model co-evolution, Metamodel/Transformation co-evolution, EMF Migrate
3	**Course prerequisites**	General admission requirements for the study programme. Background knowledge on the Unified Modelling Language (UML) is an advantage as well as a solid knowledge of the object-oriented paradigm.
4	**Teaching methodsand language**	The course consists of 4 hours of combined lectures and hands-on exercises per week. In addition, there are smaller mandatory assignments and a larger project. The project work will be concerned with the study and/or practical application of recent techniques for model-driven development. Regular assessments of the project progresses are recommendable must not mandatory, the project outcome must be documented in a 10 page written report. Assignments are individual, whereas the project can be conducted in

		groups of 2-4 participants. **Language**: English
5	**Assessment methods**	