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

Catalogo Corsi

Pagina compilata il 18/09/2019

Programma del Corso integrato "Model-Driven Engineering and Formal Methods"		
Il corso e' composto da 2 moduli: 1) Formal Methods, 2) Model Driven Engineering		
Programma del Modulo "Formal Methods"		
• Codice: DT0202 • Tipo di corso: • Livello del corso: • Semestre: 2		
Numero di crediti ECTS: (Laurea Magistrale in Informatica) 6 (carico 150 ore)		
Docenti: Monica Nesi (monica.nesi@univaq.it)		
1	**Obiettivi del corso**	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	**Contenuti del corso e risultati formativi (descrittori di Dublino)**	Gli argomenti trattati nel corso comprendono: • Sistemi di riduzione astratti, forma normale, convertibilità, grafi di riduzione. Proprietà di confluenza e Church-Rosser e loro equivalenza. Locale confluenza, terminazione, canonicità. Principio di induzione noetheriana, lemma di Newman e sua dimostrazione. • Termini del prim'ordine, sostituzioni, sostituzioni istanziatrici ed unificatrici, mgu. Algoritmo di unificazione sintattica. Sistemi di riscrittura di termini. Terminazione: ordinamenti di riduzione, di semplificazione e per cammino ricorsivo (rpo). • Confluenza: sovrapposizione di regole, coppie critiche, Lemma di Huet e sua dimostrazione. Problema della parola e sua decidibilità, teorema di Knuth-Bendix. Procedure di completamento tramite regole di inferenza, terminazione e divergenza del completamento (pattern di divergenza). E-unificazione di termini, relazione di narrowing, procedura di E-unificazione basata su narrowing, narrowing normale e basilare. • Formule booleane, soddisfacibilità, tautologia. Formule in forma CNF ed algoritmo di Davis- Putnam. Deduzione naturale. Logica predicativa: predicati, funzioni, variabili, quantificatori, regole di deduzione naturale. Forma prenex DNF. • Introduzione alle logiche di ordine superiore e al lambda-calcolo. Lambda-calcolo senza tipi, beta-riduzione, teoria dei tipi semplice, un calcolo per l'assegnamento di tipi, polimorfismo. Alla fine del corso, lo studente dovrebbe: • 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	**Prerequisiti**	Basic notions of mathematical logic and functional programming are helpful.

4	**Modalita' e lingua di insegnamento**	Lectures and exercises **Lingua**: Inglese **Testi/Bibliografia** • 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	**Metodi di accertamento**	La prova di esame consiste in una prova scritta più una prova orale.

Programma del Modulo "Model Driven Engineering"

- Codice: F0193
- Tipo di corso: Opzionale (Laurea Magistrale in Informatica percorso SDRC), Opzionale (Laurea Magistrale in Informatica percorso ASSC), Obbligatorio (Laurea Magistrale in Informatica percorso GSEEM), Opzionale (Laurea Magistrale in Informatica percorso Generale)
- Livello del corso: Lauree Magistrali
- Semestre: 2

Numero di crediti ECTS: (Laurea Magistrale in Informatica) 6 (carico 150 ore)

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

1	**Obiettivi del corso**	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	**Contenuti del corso e risultati formativi (descrittori di Dublino)**	Gli argomenti trattati nel corso comprendono: • 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 Alla fine del corso, lo studente dovrebbe: • 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

		APPLICATION: 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 EVALUATION: 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
3	**Prerequisiti**	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	**Modalita' e lingua di insegnamento**	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. **Lingua**: Inglese
5	**Metodi di accertamento**	Mandatory assignments: Taking the oral exam requires that the mandatory assignments have been approved and that the project report has been delivered together with the developed artefacts. Methods of assessment: 30-minute oral exam about the project work. Both the oral part and the project report part must result in a pass grade in order to pass the course. Moreover, an assessment of the individual assignments will be considered. Grades are awarded on a scale from 1/30 to 30/30, where 30/30 is the best grade and below 18/30 is a fail. In case, the grade awarded is 30/30 in exceptional cases a “Summa cum Laude” can be awarded as well.