

FI A DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea de Vest din Timișoara
1.2. Facultatea	Matematică și Informatică
1.3. Departamentul	Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	licență
1.6. Programul de studii / calificarea*	Informatică / <i>Administrator baze de date - 252101; Administrator de rețea de calculatoare - 252301; Analist - 251201; Asistent de cercetare în informatica - 214918; Asistent de cercetare în matematica-informatica - 212024; Profesor în învățământul gimnazial - 233002; Programator - 251202; Proiectant sisteme informatice - 251101</i>

2. Date despre disciplină

2.1. Denumirea disciplinei	Metode formale in dezvoltarea aplicatiilor informatice						
2.2. Titularul activităților de curs	Madalina Erascu						
2.3. Titularul activităților de seminar	Madalina erascu						
2.4. Anul de studii	3	2.5. Semestrul	2	2.6. Tipul de evaluare	C	2.7. Regimul disciplinei	D O

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	3	din care: 3.2 curs	2	3.3. seminar/laborator	1
3.4. Total ore din planul de învățământ	42	din care: 3.5 curs	28	3.6. seminar/laborator	14
Distribuția fondului de timp*					ore
Studiu după manual, suport de curs, bibliografie și notițe					28
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate					31
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					21
Examinări					6
Tutorat					2
3.7. Total ore studiu individual	130				
3.8. Total ore pe semestru	88				
3.9. Număr de credite	5				

4. Precondiții (acolo unde e cazul)

4.1. de curriculum	Logica computationala, algoritmica
4.2. de competențe	Cunostiinte matematice si de rezolvare a problemelor

5. Condiții (acolo unde e cazul)

5.1. de desfășurare a cursului	Sala de curs cu tabla si proiector
5.2. de desfășurare a seminarului/laboratorului	Sala de laborator cu tabla si computere. Se va instala

	o mașină virtuală cu software-ul utilizat.
--	--

6. Competențe specifice acumulate

Competențe profesionale	Prezentarea și înțelegerea (1) utilității teoriilor logice în verificarea programelor, (2) analizei statice a programelor, invariante, termeni de terminare, (3) particularităților verificării programelor Java. Prezentarea tematicii alese implică competențe de înțelegere a conceptelor noi, sintetizarea și prezentarea lor (prin raport final și prezentare).
Competențe transversale	Abilitatea de a comunica cunoștințe despre diferite noțiuni de logică computațională și verificarea statică a programelor.

7. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

7.1. Obiectivul general al disciplinei	Înțelegerea diferitor termeni ce sunt implicați în verificarea statică a programelor.
7.2. Obiectivele specifice	<p><i>Objective de cunoaștere:</i> (1) cunoașterea a diferitor concepte ce intervin în verificarea formală (2) sinteza de invariante și termeni de terminare (3) specificarea și verificarea programelor Java și particularități.</p> <p><i>Objective de abilitare:</i> (1) Prove or disprove theorems in automata theory using its properties (2) Determine the decidability and intractability of computational problems</p> <p><i>Objective atitudinală:</i> explicarea importanței verificării programelor și a logicii în această activitate.</p>

8. Conținuturi*

8.1. Curs	Metode de predare	Observații
C1 - C3 (6h). Introducere. Prezentarea importanței verificării programelor. Noțiuni de logică propozițională și predicatelor (incluzând metode de demonstrație).	Prelegere, conversație, exemplificare	Referințe: 1. Madalina Erascu - notite de curs 2. A. Bradley, Z. Manna. The Calculus of Computation. Decision procedures with Applications to Verification. Springer 2007
C4 - C6 (6h) Specificarea și verificarea programelor secvențiale. Metoda lui Hoare. Generarea de condiții de verificare și demonstrarea lor. Terminare.	Prelegere, conversație, exemplificare	Aceleași ca mai sus

C7 - C8 (4h) Specificarea și verificarea programelor Java.	Prelegere, conversație, exemplificare	Aceleași ca mai sus
C9 - C13 (10h) Prezentare diferită tematică de actualitate în cercetare în metode formale de către cadrul didactic și/sau de către studenți. Acestea includ: model checking, satisfiability (SAT) checking, satisfiability modulo theories (SMT), metode pentru eliminarea cuantificatorilor în diferite teorii (quantifier elimination)	Prelegere, conversație, exemplificare	Studentii sunt îndemnați să caute materiale adiționale (incluzând bibliografia recomandată)

Bibliografie

- [1] C.-L. Chang, R. C. T. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Computer Science Classics
- [2] C.A.R. Hoare. *An axiomatic Basis for Computer Programming*.
- [3] M. Huth, M. Ryan. *Logic in Computer Science. Modelling and Reasoning about Systems*.
- [4] L. de Moura, N. Björner. *Satisfiability Modulo Theories: Introduction and Applications*.
- [5] J. Woodcock et al. *Formal Methods: Practice and Experience*
- [6] Formal Verification of Object-Oriented Software: <http://www.cost-ic0701.org/>
- [7] A. Biere, M. Heule, H. Van Maaren, T. Walsh. *Handbook of Satisfiability*. IOS Press 2009
- [8] A. Bradley, Z. Manna. *The Calculus of Computation. Decision procedures with Applications to Verification*. Springer 2007
- [9] D. Kroening, O. Strichman. *Decision Procedures An Algorithmic Point of View*. Springer 2008

8.2. Seminar/laborator	Metode de predare/ invățare	Observații
1. Exerciții teoretice de specificare și verificare a programelor, generare de invariante și termeni de terminare folosind 2. Exerciții practice ca cele de la (1) folosind tool-uri ca RISC Program Explorer and RISC Program Navigator	Problematizare, dialog, învățare prin dialogare	Based on the notions presented in the lecture, the students will be able to access the homework from the course website. They have to prepare it. It will then be discussed in the class.

Bibliografie

- [1] C.-L. Chang, R. C. T. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Computer Science Classics
- [2] C.A.R. Hoare. *An axiomatic Basis for Computer Programming*.
- [3] M. Huth, M. Ryan. *Logic in Computer Science. Modelling and Reasoning about Systems*.
- [4] L. de Moura, N. Björner. *Satisfiability Modulo Theories: Introduction and Applications*.
- [5] J. Woodcock et al. *Formal Methods: Practice and Experience*
- [6] Formal Verification of Object-Oriented Software: <http://www.cost-ic0701.org/>

- [7] A. Biere, M. Heule, H. Van Maaren, T. Walsh. *Handbook of Satisfiability*. IOS Press 2009
 [8] A. Bradley, Z. Manna. *The Calculus of Computation. Decision procedures with Applications to Verification*. Springer 2007
 [9] D. Kroening, O. Strichman. *Decision Procedures An Algorithmic Point of View*. Springer 2008

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului

Continutul cursului este similar cu altele, pe aceeași tematică, de la alte universități. Acoperă aspectele fundamentale necesare pentru familiarizarea cu problemele rezolvate de verificarea statică a programelor. La momentul actual, cursul nu pare foarte folosit pentru industria IT din România, dar metodele formale sunt din ce în ce mai mult întâlnite în industriile care dezvoltă software pentru sisteme critice vieti (industria de avioane, mașini, medicală).

10. Evaluare*

Tip de activitate	10.1. Criterii de evaluare**	10.2. Metode de evaluare***	10.3. Pondere din nota finală
10.4. Curs	Evaluarea cunoștințelor din cursurile C1-C7 precum și partea practică L1-L7 mai puțin părțile care țin de sintaxa toolurilor RISC Proof Navigator and RISC Program Explorer.	Examen scris în ultima întâlnire de la curs sau prezentare (oral și raport)	50%
10.5. Seminar/laborator	Rezolvarea de teme bazate pe noțiunile prezentate la curs (teoretic și practic).	Teme	40%

10.6. Standard minim de performanță

Condițiile minimale (nota 5) de promovare sunt:

- dobândirea unei înțelegeri fundamentale a cunoștințelor de bază din logica computațională și verificarea statică a programelor
 abilitatea de a specifica programe simple în pseudocod dar și în Java folosind formalisme adecvate
 sinteza de invariante și termeni de terminare

Nota finală se calculează ca medie ponderată a notelor de la 10.4-10.5. Examenul se poate promova chiar dacă una dintre notele de la examen sau laborator nu este 5. În toate sesiunile, incluzând restante și mariri, nota se calculează la fel. Se poate reexamina doar partea scrisă (examen).

Remarca: Studenții au posibilitatea de a participa la ore de consultanță (2 ore / săptămână conform orarului stabilit la începutul semestrului) unde profesorul de la curs/laborator va răspunde neclarităților acestora legate de materialul de la curs, seminar/laborator.

Studenții care optează pentru proiect trebuie să se întâlnească cu cadrul didactic de la curs de 2 ori pentru stabilirea conținutului prelegerii și a raportului final.

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

07.10.2016

Madalina Erascu

Madalina Erascu

Semnătura directorului de departament
Conf.dr. Victoria Iordan