

## FIȘA DISCIPLINEI

### Metode formale de testare si verificare

Anul universitar 2017-2018

#### 1. Date despre program

1.1	Instituția de învățământ superior	Universitatea din Pitești
1.2	Facultatea	Electronică, Comunicații și Calculatoare
1.3	Departamentul	Electronică, Calculatoare și Inginerie Electrică
1.4	Domeniul de studii	Calculatoare și tehnologia informației
1.5	Ciclul de studii	Licență
1.6	Programul de studii / Calificarea	Calculatoare / Inginer

#### 2. Date despre disciplină

2. Date despre disciplina											
2.1	Denumirea disciplinei					Metode formale de testare si verificare					
2.2	Titularul activităților de curs					Prof. univ. dr. Tudor BĂLĂNESCU					
2.3	Titularul activităților de laborator					Lect. univ. Cristina TUDOSE					
2.4	Anul de studii	IV	2.5	Semestrul	I	2.6	Tipul de evaluare	C	2.7	Regimul disciplinei	D/O

#### 3. Timpul total estimat

3.1	Număr de ore pe săptămână	3	3.2	din care curs	2	3.3	laborator	1
3.4	Total ore din planul de învățământ	42	3.5	din care curs	28	3.6	laborator	14
Distribuția fondului de timp								ore
Studiul după manual, suport de curs, bibliografie și notițe								24
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren								12
Pregătire seminarii/laboratoare, teme, referate, portofolii, eseuri								12
Tutorat								2
Examinări								4
Alte activități .....								-
3.7	Total ore studiu individual	54						
3.8	Total ore pe semestru	96						
3.9	Număr de credite	4						

#### 4. Precondiții (acolo unde este cazul)

4.1	De curriculum	Programarea orientată pe obiecte, Ingineria programelor, Proiectarea algoritmilor
4.2	De competențe	C5 Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software

#### 5. Condiții (acolo unde este cazul)

5.1	De desfășurare a cursului	Sală dotată cu videoproector și ecran de proiecție, 1 tablă.
5.2	De desfășurare a laboratorului/proiectului	Sala de laborator dotată cu 18 calculatoare, capacitate maximă 18 studenți/ laborator. - Sala

#### 6. Competențe specifice acumulate

Competențe profesionale	<b>C5</b> Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software (4 p.c.) <b>C5.1</b> Precizarea criteriilor relevante privind ciclul de viață, calitatea, securitatea și interacțiunea sistemului de calcul cu mediul și cu operatorul uman (1 p.c.) <b>C5.2</b> Utilizarea unor cunoștințe interdisciplinare pentru adaptarea sistemului informatic în raport cu cerințele domeniului de aplicații (1 p.c.) <b>C5.3</b> Utilizarea unor principii și metode de bază pentru asigurarea securității, siguranței și ușurinței în exploatarea a sistemelor de calcul (1 p.c.) <b>C5.4</b> Utilizarea adecvată a standardelor de calitate, siguranța și securitate în prelucrarea informațiilor (1 p.c.)
-------------------------	---

#### 7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	- Cunoașterea conceptelor legate de modelarea sistemelor de programe, verificarea proprietăților modelelor și a protocoalelor de comunicare. Însușirea abilităților de elaborare a modelelor, de clasificare a proprietăților acestora și de exprimare formalizată a proprietăților. Familiarizarea cu tehnicile de verificare și testare automatizată.
7.2 Obiectivele specifice	<b>Cognitive:</b> Cunoașterea metodologiilor de demonstrare formală a proprietăților programelor cu instrucțiuni iterative. - Însușirea tehnicilor de specificare a sistemelor de programe utilizând sisteme de tranziție. - Cunoașterea criteriilor de clasificare a proprietăților sistemelor. - Familiarizarea cu tehnicile de verificare a modelelor (model checking)

	<p><i>Procedurale:</i></p> <ul style="list-style-type: none"> <li>- folosirea corectă a termenilor și conceptelor</li> <li>- utilizarea metodologiilor formalizate de elaborare a testelor și de testare automatizată.</li> <li>- utilizarea sistemelor specializate în verificarea și testarea automată.</li> </ul> <p><i>De atitudine:</i></p> <ul style="list-style-type: none"> <li>- deprindere de a verifica și testa sistemele software</li> </ul>
--	---

## 8. Conținuturi

8.1. Curs		Metode de predare	Observații Resurse folosite
1	<b>1. Introducere</b> 1.1. Formal și informal în specificare, verificare și testare 1.2. Verificare vs. Validare 1.3. Verificare statică vs. Verificare dinamică 1.4. Sisteme deductive pentru demonstrarea proprietăților programelor -Timp alocat 4 ore	Prelegere Dezbateri	Calculator, Videoproiector, Suport documentar
2	<b>2. Sisteme deductive pentru instrucțiuni iterative</b> 2.1. Corectitudine parțială și corectitudine totală 2.2. Proprietăți invariante și funcții de terminare 2.3. Modelarea calculului paralel cu instrucțiuni nedeterminate 2.4. Sisteme deductive pentru verificarea statică a programelor secvențiale și a celor paralele -Timp alocat 4 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Suport documentar
3	<b>3. Specificarea formalizată prin sisteme de tranziție</b> 3.1. Sisteme de tranziție cu număr infinit de stări 3.2. Modelarea procesării paralele prin întreprindere 3.3. Specificarea protocoalelor de comunicare prin sisteme de grafuri program 3.4. Calcul asincron și calcul sincron. Sincronizare prin handshaking. 3.5. Comunicare prin canale -Timp alocat 4 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Suport documentar
4	<b>4. Verificarea formalizată a modelelor sistemelor de programe</b> 3.1. Clasificarea proprietăților sistemelor. Proprietăți safety și proprietăți liveness 3.2. Criterii de executare imparțială a proceselor. Fairness. Imparțialitate slabă și imparțialitate tare. 3.3. Logică temporală. Specificarea proprietăților utilizând formule LTL sau formule CTL -Timp alocat 4 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Suport documentar
5	<b>5. Sisteme specializate de verificare</b> 1.1. SPIN 2.1. UPPAAL 3.1. NuSMV 4.1. Limbajul Promela -Timp alocat 4 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Suport documentar
6	<b>6. Specificare prin mașini cu număr finit de stări</b> 6.1. Testare vs. demonstrare formală 6.2. Specificare prin mașini cu număr finit de stări 6.3. Relația de similaritate și disimilaritate între stări. Calculul algoritmic al relațiilor 6.4. Mașini cu număr minim de stări -Timp alocat 4 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Suport documentar
7	<b>7. Testare automată</b> 7.1. Metoda W de elaborare a testelor 7.2. Metoda Wp de elaborare automată a testelor. 7.3. Implementarea sistemelor de testare automată -Timp alocat 4 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Suport documentar
<b>Bibliografie:</b> 1. T. Bălănescu: Metodologii formale de verificare și testare, Note de curs, 2017-2018 2. Aditya P. Mathur: Foundation of Software Testing, Pearson Education, 2008 3. Christel Baier, Joost-Pieter Katoen: Principles of Model Checking , The MIT Press, 2008 4. T. Bălănescu: Corectitudinea algoritmilor, Editura Tehnică, 1995 5. E.M. Clarke Jr, O. Grumberg, D. A. Peled: Model Checking, The MIT Press, 2000 6. M. Holcombe, F. Ipate: Correct Systems: building business process solutions, Springer Verlag, 1998.			
8.2. Aplicații – Laborator		Metode de predare	Observații Resurse folosite
1	<b>1. Introducere</b>	Prelegere	Calculator, Videoproiector,

	1.1. Verificarea statică a algoritmilor iterativi 1.2. Proprietăți invariante și funcții de terminare -Timp alocat 2 ore	Dezbateri	Support documentar
2	<b>2. Sisteme deductive pentru instrucțiuni iterative</b> 1.3. Verificarea/testarea unor algoritmi de sortare 1.4. Verificarea/testarea unor algoritmi de căutare -Timp alocat 2 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Support documentar
3	<b>3. Specificarea formalizată prin sisteme de tranziție</b> 3.1. Specificarea unui sistem automat de trecere la nivel de cale ferată 3.2. Specificarea protocoalelor de sincronizare cu semafoare 3.3. Protocolul lui Pettersson -Timp alocat 2 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Support documentar
4	<b>4. Verificarea formalizată a modelelor sistemelor de programe</b> 4.1. Proprietăți safety și proprietăți liveness ale sistemelor utilizând formule LTL sau formule CTL 4.2. Programare în Promela 4.3. Verificarea modelelor cu sistemul SPIN -Timp alocat 2 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Support documentar
5	<b>5. Sisteme specificate prin automate cu cronometru</b> 5.1. Variantă cu cronometru a sistemului de trecere la calea ferată 5.2. Verificarea acestui model utilizând utilitarul UPPAAL -Timp alocat 2 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Support documentar
6	<b>6. Specificare prin mașini cu număr finit de stări</b> 6.1. Specificare prin mașini cu număr finit de stări a unor protocoale de comunicare <b>6.2. Implementarea și testarea acestor sisteme</b> -Timp alocat 2 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Support documentar
7	<b>7. Testare automată</b> 7.1. Metoda W de elaborare a testelor 7.2. Metoda Wp de elaborare automată a testelor. 7.3. Implementarea sistemelor de testare automată -Timp alocat 2 ore	Prelegere Dezbateri Studiu de caz	Calculator, Videoproiector, Support documentar
<b>Bibliografie:</b> <ol style="list-style-type: none"> <li>1. V.S. Alagar, K. Periyasamy: Specification of Software Systems, Springer, 1998</li> <li>2. C. Tudose: Îndrumar Laborator, Metode Formale de Verificare și Testare, 2017-2018</li> <li>3. T. Bălănescu: Corectitudinea algoritmilor, Editura Tehnică, 1995</li> <li>4. E.M. Clarke Jr, O. Grumberg, D. A. Peled: Model Checking, The MIT Press, 2000</li> <li>5. M. Holcombe, F. Ipate: Correct Systems: building business process solutions, Springer Verlag, 1998</li> <li>6. Aditya P. Mathur: Foundation of Software Testing, Pearson Education, 2008</li> <li>7. G. Mühl, L. Fiege, P. Pietzuch: Distributed Event- Based Systems, Springer, 2006</li> </ol>			

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

Competențele dobândite la disciplină permit absolvenților să coordoneze echipele de specificare, implementare și validare a sistemelor software complexe de natură paralelă, concurență sau distribuită. Utilizarea cunoștințelor dobândite contribuie la diminuarea costurilor de realizare a unor astfel de sisteme software prin: specificarea formalizată; elaborarea sistematică a datelor de test; utilizarea unor sisteme software de testare automată; complementarea activității de testare prin includerea acțiunilor de verificare a proprietăților modelelor.

**10. Evaluare**

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	a) Interes pentru obiectivele disciplinei b) Examen	a) Evaluarea periodică a implicării în însușirea tehnicilor de verificare integrată a sistemelor b) Scris - verificare cunoștințe teoretice	10% 30%
10.5 Laborator	Verificarea și testarea sistemelor utilizând produse software specializate	Evaluare periodică	60%
10.6 Standard minim de performanță	<p>* Prezență totală și notă minimă 5 la activitățile de laborator și notă minimă 5 la fiecare din subiectele de la examenul final.</p> <p>* Set de cunoștințe minimale pentru promovarea examenului final:</p> <ul style="list-style-type: none"> <li>- Utilizarea proprietăților invariante pentru demonstrarea corectitudinii parțiale</li> <li>- Proprietăți safety vs. proprietăți liveness</li> <li>- Elemente esențiale de logică temporală. Operatorii modali next, eventually, always, until.</li> </ul>		

	- Metoda W -Utilitarul UPAAL
--	---------------------------------

Data completării  
22.09.2017

Titular de curs și proiect  
Prof. univ. dr. Tudor BĂLĂNESCU

Titular de laborator  
Lect. univ. Cristina TUDOSE

Data avizării în departament  
25.09.2017

Director de departament  
Prof. univ. dr. ing. Gheorghe SERBAN