



UNIVERSITATEA „VASILE ALECSANDRI” din BACĂU
Facultatea de Inginerie
 Calea Mărășești, Nr. 157, Bacău, 600115, Tel./Fax +40 234
 580170
<http://inginerie.ub.ro>; e-mail: decaning@ub.ro



FIȘA DISCIPLINEI
(licență)

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea „Vasile Alecsandri” din Bacău
1.2. Facultatea	Facultatea de Inginerie
1.3. Departamentul	Energetică și Știința Calculatoarelor
1.4. Domeniul de studii	Calculatoare și Tehnologia Informației
1.5. Ciclul de studii	Licență
1.6. Programul de studii/calificarea	Tehnologia Informației
1.7. Forma de învățământ	Învățământ cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei	Limbaje formale și translațoare				
2.2. Titularul activităților de curs	Lect Dr Dan Popa				
2.3. Titularul activităților de seminar	Lect Dr Dan Popa				
2.4. Anul de studiu	III	2.5. Semestrul	I	2.6. Tipul de evaluare	examen
2.7. Regimul disciplinei	Categoría formativă a disciplinei DF - fundamentală, DD - în domeniu, DS - de specialitate, DC - complementară				
	Categoría de opționalitate a disciplinei: DI - obligatorie (impusă), DO - opțională (la alegere), DL - facultativă (liber aleasă)				

3. Timpul total estimat (ore alocate activităților didactice)

3.1. Număr de ore pe săptămână	3	3.2. Curs	2	3.3. Seminar/Laborator/Proiect	1
3.4. Totalul de ore pe semestru din planul de învățământ	56	3.5. Curs	28	3.6. Seminar/Laborator/Proiect	14

Distribuția fondului de timp pe semestru:	ore
Studiul după manual, suport de curs, bibliografie și notițe	10
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren	9
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	4
Tutoriat	5
Examinări	5
Alte activități (precizați):	

3.7. Total ore studiu individual	33
3.8. Total ore pe semestru	75
3.9. Numărul de credite	3

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none"> • Programarea calculatoarelor și limbaje de programare • Algebra și teoria mulțimilor
4.2. de competențe	<ul style="list-style-type: none"> •

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	•
5.2. de desfășurare a seminarului/laboratorului/proiectului	• Laborator cu computere cu S.O Linux și proiector video

6. Competențe specifice acumulate

6.1. Competențe profesionale	C5.1. Identificarea și descrierea instrumentelor de modelare, simulare a sistemelor hardware, software și de comunicații C5.2. Utilizarea unor cunoștințe interdisciplinare pentru domeniului de aplicații C5.3. Utilizarea unor principii și metode de bază pentru asigurarea securității, sistemelor hardware, software și de comunicații C5.5. Dezvoltarea de sisteme și aplicații
6.2. Competențe transversale	

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

7.1. Obiectivul general al disciplinei	C1.1. Recunoașterea și descrierea conceptelor proprii C1.2. Utilizarea de teorii și instrumente specifice pentru explicarea structurii și funcționării sistemelor software și de comunicații C1.3. Construirea unor modele pentru diferite componente ale sistemelor de calcul C1.4. Evaluarea formală a caracteristicilor funcționale și nefuncționale ale sistemelor de calcul C1.5. Fundamentarea teoretică a caracteristicilor sistemelor proiectate
7.2. Obiectivele specifice	<ul style="list-style-type: none"> • Însușirea tehnicilor de citire a datelor și de specificare a formatelor de date și a limbajelor. • Folosirea gramaticilor ca instrumente de specificare. • Folosirea și implementare automatelor • Algoritmi de analiza lexicala și sintactica • Construcția unui compilator.

8. Conținuturi

Curs	Nr. ore	Metode de predare	Observații
<ul style="list-style-type: none"> • Introducere în gramatici și automate. Notății algebrice folosite: mulțimi, relații, relații binare, tipuri de relații, închiderea unei relații, inducție, scheme de inducție, inducție cu mai multe reguli, recursie, recursivitatea ca instrument de realizare a interpretoarelor și compilatoarelor; 	4	Demonstrații live pe videoproiector și prezentarea elementelor de proiectat în fiecare etapă	
<ul style="list-style-type: none"> • Noțiuni de teoria grafurilor, Limbaje formale, văzute ca mulțimi de cuvinte, Reuniuni și operații cu limbaje, Gramatici ale limbajelor formale, Derivare, Exemplu: Gramatica unui limbaj de programare. Clasificarea gramaticilor, Ierarhia lui Chomsky, Echivalența gramaticilor, Limbaje decidabile. 	4		

<ul style="list-style-type: none"> Automate finit(e) deterministe: Complexitatea analizei sintactice, de ce limbajele de tip 3 sunt interesante. O gramatică de tip 3. Reprezentarea grafică. Introducere informală în automate. Tranziții deterministe și tranziții nedeterministe. Automate finite deterministe și automate finite nedeterministe, Configurații și relații de tranziție, Configurație inițială, configurație finală cu stare finală, Limbaje acceptate de automate finite. Automate, reprezentări de automate, reprezentare tabelară, reprezentare graf, automate echivalente, funcționarea automatelor, mișcare, blocare, oprire și staționare. Stări accesibile, stări inaccesibile. Stări accesibile, stări inaccesibile., Algoritmul de determinare a stărilor accesibile și respectiv productive, Echivalența AFD – AFN. Transformarea AFN în AFD. Exemplu. Definiția AFD redus. Automate cu epsilon-mișcări. Transformarea unui AF pentru eliminarea lor. 	4	
<ul style="list-style-type: none"> Expresii regulate, Mulțimi regulate, Expresii regulate asociate mulțimilor, Operațiile cu automate asociate. Automate cu epsilon-mișcări care corespund expresiilor regulate. Exemple de limbaje. 	4	
<ul style="list-style-type: none"> Proprietăți algebrice ale expresiilor regulate. Construcția expresiei regulate corespunzătoare limbajului recunoscut de un automat. Metoda I: construcția inductivă a unui set de mulțimi, Metoda a II-a, Construcția unui sistem de ecuații liniare și rezolvarea acestuia. 	2	
<ul style="list-style-type: none"> Echivalența dintre limbaje regulate specificate prin gramatici și limbaje regulate recunoscute de automate. Gramatici și limbaje independente de context. Proprietăți de închidere pentru limbajele de tip 2. 	2	
<ul style="list-style-type: none"> Arbori de derivare, Analiza sintactică, Frontiera unui arbore de analiză sintactică, Gramatici ambigue și neambigue, Simplificarea gramaticilor dependente de context. Simboluri inaccesibile – eliminarea lor, simboluri neproductive – eliminarea lor, simboluri neutilizabile – eliminarea lor, epsilon - producții și eliminarea lor, redenumiri și eliminarea lor. 	2	
<ul style="list-style-type: none"> Recursivitate și eliminarea recursivității la stânga. Leme de pompare pentru limbaje independente de context. 	2	
<ul style="list-style-type: none"> Implementari: Implementarea unui compilator și a unei mașini virtuale cu flex și bison. 	2	
<ul style="list-style-type: none"> Implementari: Transformarea regulilor gramaticale în funcții mutual recursive. Limitările metodei. 		
Bibliografie		
<p>[1] Grigor Moldovan, <i>Limbaje Formale și Teoria automatelor</i>, Edusoft, Bacău, 2005</p> <p>[2] Grune, Dick; Jacobs, J.H. Ceriél, <i>Parsing Techniques, A Practical Guide</i>, 2008 Springer Verlag</p> <p>[3] Linz Peter, <i>An Introduction to formal languages and automata</i>, Londra , Jones & Bartlett Learning International, 2011 https://fall14cs.files.wordpress.com/2017/04/an-introduction-to-formal-languages-and-automata-5th-edition-2011.pdf</p> <p>[4] *** Automata Theory, Tutorials Point, 2016 https://www.tutorialspoint.com/automata_theory/automata_theory_tutorial.pdf</p> <p>[5] John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. <i>Introduction to automata theory, languages, and computation</i>, Pearson Education, 2004-2006 http://ce.sharif.edu/courses/94-95/1/ce414-2/resources/root/Text%20Books/Automata/John%20E.%20Hopcroft,%20Rajeev%20Motwani,%20Jeffrey%20D.%20U</p>		

[lman-Introduction%20to%20Automata%20Theory,%20Languages,%20and%20Computations-Prentice%20Hall%20\(2006\).pdf](#)

[6] Xavier,S.P.Eugene, Theory of Automata, Formal Languages and Computation, New Delhi, New Age International, 2005

https://cse4projects.files.wordpress.com/2013/09/theory_of_computation_reference_e-book.pdf

Bibliografie minimală

O pereche formata dintr-un curs de L.F. și T.A .cum este [1] si un manual de construcția compilatoarelor cu Flex și Bison cum este 1) Pentru exercitii teoretice: [6]

Aplicații (Seminar / laborator / proiect)	Nr	Metode de predare	Observații
1. Implementarea limbajul Simple:	1	Demonstrații live pe videoproiector și prezentarea elementelor de proiectat în fiecare etapă	
2. Implementarea analizorului sintactic	1		
3. (parserul)			
4. Implementarea scannerului (analizorul lexical)	1		
5. Implementarea contextului	1		
6. Optimizarea (arborelui)	1		
7. Implementarea masinii virtuale pe care va rula codul	1		
8. Implementarea generatorului de cod,			
9. Optimizarea codului	1		

Bibliografie

0) Alfred V. Aho , Monica S. Lam , Ravi Sethi ,Compilers: Principles, Techniques, and Tools (Editia a II-a) Hardcover – Sept. 10, 2006

1) Aaby, Anthony. A.; Popa, D; *Construcția compilatoarelor folosind Flex și Bison*, MatrixRom, Bucuresti, 2009 (editia a II-a)

2) Șerbănași, L. D., *Limbaje de programare și compilatoare*, București, Editura Academiei, 1987

3) Ionescu, C., Zsako, I., „Structuri arborescente în procesele de de compilare”, în *Structuri arborescente cu aplicațiile lor* (Partea a II-a), sub controlul științific al prof. dr. L. Livovschi, București, Editura Tehnică, 1990

4) Terry, P. D., Compilers and Compiler Generators. An Introduction with C++, Rhodes University, 1996.

5) Géza Horváth, Benedek Nagy, *Formal Languages and Automata Theory, 2014 Géza Horváth, Benedek Nagy, University of Debrecen*

6) John C. Martin, *Introduction to Languages and The Theory of Computation* Fourth Edition North Dakota State University, 2011 by The McGraw-Hill Companies,

7)Modern Compiler Implementation in C, ANDREW W. APPEL, CAMBRIDGE UNIVERSITY PRESS, 1998

8) Grigor Moldovan, *Limbaje formale si teoria automatelor*, Edusoft 2005

Bibliografie minimală

1) Aaby, Anthony. A.; Popa, D; *Construcția compilatoarelor folosind Flex și Bison*, MatrixRom, Bucuresti, 2009 (editia a II-a)

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

Este coroborat fiind curs de nivel european de specialitate cu acces la o nișă de pe piața muncii:

Construcția modulelor translaatoare care sunt folosite în aplicațiile informatice la completarea meniul File cu opțiunile Open, Load, Save și Save As.

10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Evaluarea culturii generale (în domeniu) corespunde setului de intrebari nr. 1. Evaluarea cunostintelor teoretice (în domeniu) corespunde setului de intrebari nr. 2.	Examenare prin test cu intrebari întrebări care verifică existența cunstintelor fundamentale.	60%
10.5. Seminar/laborator/pro	Evaluarea abilitatilor de programator și rezolvitor de exercitii (în domeniu)	Examinare prin întrebări care verifică existența	40%

iect	corespunde setului de intrebari nr. 3. Evaluarea cunostintelor suplimentare nivelului 3, individual acumulate corespunde setului de intrebari nr. 4.	abilităților practice și experienței ce decurge din ele și (la nivelul 4) integrarea teoriei cu practica.	
10.6. Standard minim de performanță:			
Nivelul 1 + Nivelul al 2-lea – a se vedea 10.4			

Data completării	Semnătura titularului de curs	Semnătura titularului de seminar
20.9.2021	Lector dr. Dan Popa	Lector dr. Dan Popa

Data avizării în departament	Semnătura directorului de departament
23.09.2021	Prof. univ. dr. ing. George CULEA

Data aprobării în Consiliul Facultății	Semnătura decanului
27.09.2021	Conf. univ dr. ing. Mirela PANAINTE-LEHĂDUȘ