

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Facultatea de Matematică și Informatică
1.3 Departamentul	Departamentul de Informatică
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Licență
1.6 Programul de studiu / Calificarea	Ingineria Informației (în limba engleză)

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)	Limbaje formale și automate (Formal Languages and Automata)						
2.2 Titularul activităților de curs	Prof.dr. Simona Motogna						
2.3 Titularul activităților de seminar	Prof.dr. Simona Motogna						
2.4 Anul de studiu	4	2.5 Semestrul	7	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie DD
2.8 Codul disciplinei	MLE5181						

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

3.1 Număr de ore pe săptămână	5	Din care: 3.2 curs	2	3.3 seminar/laborator	1 S 2 LP
3.4 Total ore din planul de învățământ	70	Din care: 3.5 curs	28	3.6 seminar/laborator	14+ 28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					25
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					25
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					20
Tutoriat					5
Examinări					5
Alte activități:					
3.7 Total ore studiu individual					80
3.8 Total ore pe semestru					150
3.9 Numărul de credite					6

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	•
4.2 de competențe	• Abilități medii de programare într-un limbaj de nivel înalt

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none">•
5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none">• Sală de laborator dotată cu calculatoare; medii de programare (.NET, Java, etc.)

6. Competențele specifice acumulate

Competențe profesionale	C1.2 Utilizarea de teorii și instrumente specifice (algoritmi, scheme, modele, protocoale etc.) pentru explicarea structurii și funcționării sistemelor hardware, software și de comunicații
	C1.3 Construirea unor modele pentru diferite componente ale sistemelor de calcul
	C1.5 Fundamentarea teoretică a caracteristicilor sistemelor proiectate
Competențe transversale	CT1 Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei
	CT3 Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none">• Abilitatea de a înțelege structura unui compilator și a conceptelor teoretice din teoria compilării• Îmbunătățirea abilităților de programare
7.2 Obiectivele specifice	<ul style="list-style-type: none">• Abilitatea de a înțelege și lucra cu conceptele din teoria limbajelor formale: ierarhie Chomsky, gramatici regulate, autoamte finite, echivalența dintre ele; gramatici independente de context, automate push down și echivalența dintre ele; mașini Turing• Abilitatea de a înțelege și lucra cu concepte din teoria compilării: analiza lexicală, analiza sintactică

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Introducere în noțiuni de limbaje formale: Gramatici și automate finite (AF)[1,2]	Expunere: descriere, explicație, exemplificare, dezbateri, dialog	
2. Minimizarea AF. Eliminarea ϵ -tranzității [1,2]	Expunere: descriere, explicație, exemplificare,	

	dezbateri, dialog	
3. Limbaje și expresii regulate, echivalența dintre AF, gramatici regulate și expresii regulate [1,2]	Expunere: descriere, explicație, exemplificare, dezbateri, dialog	
4. Echivalența dintre AF, gramatici regulate și expresii regulate (cont.). Lema de pompare [1,2]	Expunere: descriere, explicație, exemplificare, dezbateri, dialog	
5. Gramatici independente de context (GIC), arbore sintactic. Transgornări echivalente ale GIC [1,2]	Expunere: descriere, explicație, exemplificare, dezbateri, dialog	
6. Automate Push-down (APD). Echivalență GIC - APD [1,2]	Expunere: descriere, explicație, exemplificare, studii de caz	
7. Analiză lexicală [3,4,7,8]	Expunere: descriere, explicație, exemplificare, dezbateri, dialog	
8. Analiză sintactic: noțiuni generale, clasificare. Analiză sintactică descendentă cu reveniri [3,4,7,8]	Expunere: descriere, explicație, exemplificare, dezbateri, dialog	
9. Analiză sintactică LL(1) [3,4,7,8]	Expunere: descriere, explicație, exemplificare, dezbateri, dialog	
10. Analiză sintactică LR(k). Analiză sintactică LR(0) [3,4,7,8]	Expunere: descriere, explicație, exemplificare, dezbateri, dialog	
11. Analiză sintactică SLR, LR(1), LALR [3,4,7,8]	Expunere: descriere, explicație, exemplificare, dezbateri, dialog	
12. Generator de analiză lexicală (lex) și sintactică (yacc) [4]	Expunere: descriere, explicație, exemplificare, demo live	
13. Mașini Turing [1,2]	Expunere: descriere, explicație, exemplificare, studii de caz	
14. Structura generală a unui compilator. Fazele compilării [3,4,7,8]	Expunere: descriere, explicație, exemplificare, dezbateri, dialog	

Bibliografie

1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978.

2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973.
3. D. GRIES - Compiler construction for digital computers,, John Wiley, New York, 1971.
4. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006
5. SIPSER, M., Introduction to the theory of computation, PWS Pub. Co., 1997
6. CSÖRNYEI ZOLTÁN, Bevezetés a fordítóprogramok elméletébe, I, II., ELTE, Budapest, 1996
7. L.D. SERBANATI - Limbaje de programare si compilatoare, Ed. Academiei RSR, 1987.
8. CSÖRNYEI ZOLTÁN, Fordítási algoritmusok, Erdélyi Tankönyvtanács, Kolozsvár, 2000.
10. GRUNE, DICK - BAL, H. - JACOBS, C. - LANGENDOEN, K.: Modern Compiler Design, John Wiley, 2000

8.2 Seminar / laborator	Metode de predare	Observații
1. Specificarea unui limbaj de programare; notația BNF	Explicație, dialog, studii de caz	
2. Gramatici; limbaj generat de gramatică; gramatică corespunzătoare unui limbaj	Explicație, dialog, studii de caz, demonstrație	
3. Automat finit: limbaj acceptat de AF; AF corespunzător unui limbaj	Explicație, dialog, studii de caz, demonstrație	
4. Transformări: AF – gramatici regulate – expresii regulate	Explicație, dialog, studii de caz, exemple	
5. Gramatici independente de context; analiză sintactică LL(1)	Explicație, dialog, studii de caz, exemple	
6. Analiză sintactică LR(0)	Explicație, dialog, studii de caz, exemple	
7. Automate push-down	Explicație, dialog, studii de caz, exemple	

Bibliografie

1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978.
2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973.
3. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006
4. G. MOLDOVAN, V. CIOBAN, M. LUPEA - Limbaje formale si automate. Culegere de probleme, Univ. Babes-Bolyai, Cluj-Napoca, 1996

8.3 Laborator	Metode de predare	Observații
1. Task 1: Specificarea unui mini-limbaj și implementare analiză lexicală folosind lex 1.1: Specificare mini limbaj (notație BNF)	Explicație, dialog, studii de caz	
2. Task 1: Specificarea unui mini-limbaj și implementare analiză lexicală folosind lex 1.2: Implementare folosind lex	Explicație, dialog, studii de caz	
3. Task 2: Gramatici regulate + AF + transformări 2.1: Definiere structuri de date pentru GR și AF; implementare transformări	Explicație, dialog, studii de caz	
4. Task 2: Gramatici regulate + AF + transformări 2.2: Program principal, testare, predare	Discuție date de testare, evaluare	

5. Task 3: GIC + transformări echivalente 3.1: Extindere task 2 pentru GIC; implementare transformări	Explicație, dialog, studii de caz	
6. Task 3: GIC + transformări echivalente 3.2: Program principal, testare, predare	Discuție date de testare, evaluare	
7. Task 4: Implementare analiză sintactică 4.1: Definiere structuri de date și arhitectura	Explicație, dialog, studii de caz	Unul dintre: descendent cu reveniri, LL(1), LR(0), SLR
8. Task 4: Implementare analiză sintactică 4.2: implementare funcții analiză sintactică	Explicație, dialog, studii de caz	Task 4 în echipă de 2 studenți
9. Task 4: Implementare analiză sintactică 4.3: Program principal și integrare module	Explicație, dialog, studii de caz	
10. Task 4: Implementare analiză sintactică 4.4: testare pe gramatică	Discuție date de testare, evaluare	
11. Task 4: Implementare analiză sintactică 4.5: testare pe mini limbaj și secvență; predare	Discuție date de testare, evaluare	
12. Task 5: Instrumente lex și yacc: 5.1: implementare și predare	Explicație, dialog, studii de caz	
13. Task 5: Instrumente lex și yacc: 5.2: integrare și predare	Discuție date de testare, evaluare	
14. Prezentare finală activitate laborator	Explicație, dialog, studii de caz	
Bibliography 1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978. 2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973. 3. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006 4. G. MOLDOVAN, V. CIOBAN, M. LUPEA - Limbaje formale si automate. Culegere de probleme, Univ. Babes-Bolyai, Cluj-Napoca, 1996.		

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

<ul style="list-style-type: none"> • Cursul respectă recomandările de curriculă ACM și IEEE pentru studii de Ingineria Informației; • Cursul există la majoritatea programelor de studii similare la universități de prestigiu din România și străinătate; • Conținutul cursului este considerat important de către companii pentru acumularea de cunoștințe de programare

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	- cunoașterea principiilor de bază din domeniu - Aplicarea conceptelor de la curs în rezolvarea de probleme	Examen scris	60%
10.5 Seminar/laborator	- abilitatea de a aplica algoritmi și a înțelege exemple – rezolvare de probleme	Probleme rezolvate; teme; observare continuă pe parcursul semestrului	10%
	-Abilitatea de implementa	Examinare practică continuă	30%

	conceptele și algoritmi cursului - aplicarea tehnicilor pentru diferite clase de limbaje de programare	pe parcursul semestrului – documentație, portofoliu Github	
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • Participarea la minim 75% din activitatea de seminar și minim 90% din activitatea de laborator pe timpul semestrului • Minim nota 5 (pe o scară de la 1 la 10) la examenul scris și activitatea de laborator • Înțelegerea conceptelor de bază din limbaje formale: gramatică, automat finit, automat push down, expresii regulate; înțelegerea principiilor compilării, analiză lexicală și sintactică 			

Data completării

12.05.2022

Semnătura titularului de curs



Semnătura titularului de seminar



Data avizării în departament

24.05.2022

Semnătura directorului de departament

Prof. dr. Laura Dioșan

