

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Calculatoare română / Inginer
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Limbaje formale și translaoare				Codul disciplinei	38
2.2 Titularul de curs	Conf. dr. ing. Mărginean Anca - Anca.Marginean@cs.utcluj.ro Șl. dr. ing. Feier Cristina - Cristina.Feier@cs.utcluj.ro					
2.3 Titularul activităților de seminar / laborator / proiect	Conf. dr. ing. Mărginean Anca - Anca.Marginean@cs.utcluj.ro Șl. dr. ing. Feier Cristina - Cristina.Feier@cs.utcluj.ro					
2.4 Anul de studiu	III	2.5 Semestrul	6	2.6 Tipul de evaluare	E	
2.7 Regimul disciplinei	Categorია formativă: <i>DF – fundamentală, DS - de specialitate, DC - complementară</i>					DF
	Opționalitate: <i>DOB - obligatorie, DOP - opțională, DFA - facultativă</i>					DOB

3. Timpul total estimate

3.1 Număr de ore pe săptămână	4	din care:	Curs	2	Seminar	-	Laborator	2	Proiect	-
3.2 Număr de ore pe semestru	56	din care:	Curs	28	Seminar	-	Laborator	28	Proiect	-
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										7
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										5
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										4
(d) Tutoriat										-
(e) Examinări										3
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a)...3.3(f)))					19					
3.5 Total ore pe semestru (3.2+3.4)					75					
3.6 Numărul de credite					3					

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Programarea calculatoarelor, Structuri de date și algoritmi
4.2 de competențe	Cunoștințe de bază de programare și structuri de date (de preferință în limbajul C)

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tablă, proiector, calculator
5.2. de desfășurare a laboratorului	Calculatoare, software specific / Cont Teams

6. Competențele specifice acumulate

6.1 Competențe profesionale	• 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. C3 - Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor • C3.1 - Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice • C3.2 - Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea rezultatelor lor • C3.3 - Aplicarea tiparelor de soluții cu ajutorul uneltelor și metodelor ingineresti • C3.4 - Evaluarea comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanțelor • C3.5 - Dezvoltarea și implementarea de soluții informatice pentru probleme concrete
6.2 Competențe transversale	N/A

7. Rezultatele așteptate ale învățării

Cunoștințe	Describe, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare, medii de programare, tehnici de programare, baze de date, inteligență artificială și inginerie software și modul lor de aplicare în probleme concrete.
Aptitudini	Alege și explică concepte proprii specifice proiectării algoritmilor, programării orientate pe obiecte, programării logice și funcționale. Specifică cerințe, analizează, elaborează, dezvoltă și testează programe în limbaje de programare de uz general (C, etc.) și /sau obiect-orientate (C++, Java, etc.), aplicând elementele specifice ingineriei software. Proiectează, implementează, dezvoltă și /sau gestionează aplicații ce includ diverse tipuri de baze de date.
Responsabilitate și autonomie:	Are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.

8. Obiectivele disciplinei

8.1 Obiectivul general al disciplinei	• Cunoașterea fazelor, componentelor și algoritmilor utilizați în translatoarele de limbaj tipice • Crearea unei baze formale pentru dezvoltarea conceptelor legate de procesoarele lexicale și sintactice din translatoare
---------------------------------------	--

8.2 Obiectivele specifice	• Cunoașterea modelelor formale de bază, adică automatele finite și automatele stivă, și înțelegerea relației în care se află ele cu definițiile de limbaj prin expresii regulate și gramatici • Înțelegerea relațiilor dintre descrierea formală a automatelor din teoria limbajelor formale și implementările practice ale acestora ca analizoare lexicale și sintactice în transatoare • Cunoașterea claselor de limbaje pentru care se poate implementa un analizor sintactic determinist • Descrierea sintaxei limbajelor de implementat folosind expresii regulate și gramatici • Proiectarea, dezvoltarea și testarea unui proiect software, utilizând unelte software specializate (generatoare de analizoare), pentru a ajunge la un translator pentru un limbaj artificial • Controlul fenomenelor de ambiguitate și nedeterminism (conflicte) care apar la generatoarele de analizoare lexicale și sintactice • Introducere în metode de procesare a limbajului natural
---------------------------	--

9. Conținuturi

9.1 Curs	Nr.ore	Metode de predare	Observații
Instrumente pentru reprezentare: șiruri și sisteme de rescriere, gramatici.	2	Ideile principale cu mijloace multimedia Materiale disponibile în platforma Moodle. Detaliile și exemplele la tablă tableta grafica, în interacțiune cu studenții. Teste Kahoot/teste Teams Forms	N/A
Instrumente pentru reprezentare: derivări și arbori de derivare, notația BNF extins.	2		
Gramatici regulate și automate finite: automate finite.	2		
Gramatici regulate și automate finite: diagrame de stare și expresii regulate.	2		
Gramatici independente de context și automate stivă.	2		
Analiza descendentă și gramatici LL(k): gramatici LL(k), algoritmul LL(k).	2		
Analiza descendentă și gramatici LL(k): eliminarea recursivității stânga, factorizarea stânga.	2		
Derivatoare LL: gramatici LL(k) tari, algoritmul derivator LL(1).	2		
Derivatoare LL: derivator LL(1) în variantă interpretată, calculul mulțimilor PRIM și URMA.	2		
Analiza ascendentă și gramatici LR(k): situații și închidere nonterminal, algoritmul LR(k).	2		
Analiza ascendentă și gramatici LR(k): algoritmul LR(k).	2		
Derivatoare LR: algoritmul derivator LR(0), stări LR(0).	2		
Procesarea limbajului natural: analiza sintactică, interpretare semantică, metode de reprezentare	2		
Procesare limbajului natural: modele neuronale de reprezentare a limbajului	2		
Bibliografie (<i>bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziția studenților într-un număr de exemplare corespunzător</i>) 1. I.A. Leția, E.Șt. Chifu, Limbaje formale și transatoare, Ed. Casa cărții de știință, 1998. 2. W.M. Waite, G. Goos, Compiler Construction, Springer-Verlag, 1984. 3. A.V. Aho, R. Sethi, and J.D. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley, 1986.			
9.2 Aplicații - seminar / laborator / proiect	Nr.ore	Metode de predare	Observații
Analizor lexical de C. Expresii regulate in python.	2		
Generatorul de analizoare lexicale Lex: sursa pentru Lex, expresii regulate Lex, acțiuni Lex, reguli ambigue, definiții sursă Lex.	2		
Generatorul Lex: dependență de context stânga, exemple, aplicații Lex.	2		
Generatorul de analizoare sintactice ascendente Yacc: specificații de bază, sintaxa Yacc, acțiuni, analiza lexicală, funcționare derivator.	2		
Colocviu de laborator: utilizare LEX	2		

Generatorul Yacc: ambiguitate și conflicte, precedență și asociativitate, tratarea erorilor	2	Explicatii, implementare si testare exemple, exercitii.	
Generatorul Yacc: suport pentru valori de tip arbitrar, exemple. Generatorul YACC: translatarea expresiilor aritmetice din infix in postfix	2		
Aplicatii Yacc/Lex: interpretor LISP/C.	2		
Colocviu de laborator (Utilizare Yacc și Lex).	2		
Definirea tematicii miniproiectului individual (Implementare translator folosind generatoarele Yacc și Lex/ Utilizare modele neuronale de reprezentare a limbajului)	2		
Exemple de utilizare a modelelor neuronale de reprezentare a limbajului/ Parsarea recursiv descendenta determinista a unui limbaj ce lucreaza cu arbori binari.	2		
Exemple de utilizare a modelelor neuronale de reprezentare a limbajului/ Parsarea recursiv descendenta determinista a unui limbaj ce lucreaza cu liste si polinoame.	2		
Implementarea proiectului.	2		
Evaluarea finală a proiectului individual.	2		
Bibliografie (<i>bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător</i>) 1. https://www.cs.utexas.edu/users/novak/lexpaper.htm 2. https://www.cs.utexas.edu/users/novak/yaccpaper.htm 3. Indrumator laborator online 4. Hugging Face https://huggingface.co/			

**Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.*

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

Este o disciplină de domeniu în Calculatoare și Tehnologia Informației, conținutul ei fiind și clasic și modern. Ea îi instruieste pe studenți cu principiile de bază în proiectarea interpretoparelor și translatoarelor pentru limbaje artificiale. Conținutul disciplinei a fost discutat cu alte universități și cu companii importante din România, Europa și SUA și a fost evaluat de agenții guvernamentale românești (CNEAA și ARACIS).

11. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	- Abilități de rezolvare a problemelor - Prezență, Activitate	Test scris	60%
Seminar	-	-	-
Laborator	- Abilități de rezolvare a problemelor - Prezență, Activitate	Colocviu de laborator 1 si 2 Verificare miniproiect individual	20% 20%
Proiect	-	-	-
Standard minim de performanță: Modelarea unei probleme tipice ingineresti folosind aparatul formal caracteristic domeniului Calcul nota disciplina: 40% laborator + 60% examen final Conditii de participare la examenul final: Laborator ≥ 5 Conditii de promovare: nota disciplina ≥ 5			

Data completării: 30.04.2026	Titulari	Titlu, prenume / NUME	Semnătura
	Curs	Conf.dr.ing. Anca MĂRGINEAN	
		Șl.dr.ing. Cristina FEIER	
	Aplicații	Conf.dr.ing. Anca MĂRGINEAN	
		Șl.dr.ing. Cristina FEIER	

Data avizării în Consiliul Departamentului Calculatoare	Director Departament, Prof.dr.ing. Rodica Potolea
Data aprobării în Consiliul Facultății de Automatică și Calculatoare	Decan, Prof.dr.ing. Vlad Mureșan