

## 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
1.8 Codul disciplinei	38.

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Limbe formale și translațoare</b>				
2.2 Titularii de curs	Conf. dr. ing. Anca Mărginean - <a href="mailto:Anca.Marginean@cs.utcluj.ro">Anca.Marginean@cs.utcluj.ro</a> Conf. dr. ing. Emil Ștefan Chifu - <a href="mailto:emil.chifu@cs.utcluj.ro">emil.chifu@cs.utcluj.ro</a>				
2.3 Titularul/Titularii activităților de seminar/laborator/proiect	Conf. dr. ing. Anca Mărginean - <a href="mailto:Anca.Marginean@cs.utcluj.ro">Anca.Marginean@cs.utcluj.ro</a> Conf. dr. ing. Emil Ștefan Chifu - <a href="mailto:emil.chifu@cs.utcluj.ro">emil.chifu@cs.utcluj.ro</a>				
2.4 Anul de studiu	III	2.5 Semestrul	6	2.6 Tipul de evaluare ( E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară DI – Impusă, DOP – opțională, DFac – facultativă				DD DI

### 3. Timpul total estimat

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:										0
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	Onsite: Tablă, proiector, calculator Online: Cont Teams, calculator
5.2. de desfășurare a laboratorului	Calculatoare, software specific / Cont Teams

### 6. Competențele specifice acumulate

6.1 Competențe profesionale	<p><b>C1</b> - Operarea cu fundamente matematice, ingineresti și ale informaticii</p> <p><b>C1.1</b> - Recunoașterea și descrierea conceptelor proprii calculabilității, complexității, paradigmelor de programare și modelării sistemelor de calcul și comunicații</p> <p><b>C1.2</b> - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, protocoale etc.) pentru explicarea structurii și funcționării sistemelor hardware, software și de comunicații</p>
-----------------------------	---

	<p><b>C1.3</b> - Construirea unor modele pentru diferite componente ale sistemelor de calcul</p> <p><b>C1.4</b> - Evaluarea formală a caracteristicilor funcționale și nefuncționale ale sistemelor de calcul</p> <p><b>C1.5</b> - Fundamentarea teoretică a caracteristicilor sistemelor proiectate</p> <p><b>C3</b> - Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor</p> <p><b>C3.1</b> - Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice</p> <p><b>C3.2</b> - Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea rezultatelor lor</p> <p><b>C3.3</b> - Aplicarea tiparelor de soluții cu ajutorul uneltelor și metodelor ingineresti</p> <p><b>C3.4</b> - Evaluarea comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanțelor</p> <p><b>C3.5</b> - Dezvoltarea și implementarea de soluții informatice pentru probleme concrete</p>
6.2 Competențe transversale	N/A

### 7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> <li>• Cunoașterea fazelor, componentelor și algoritmilor utilizați în translatoarele de limbaj tipice</li> <li>• Crearea unei baze formale pentru dezvoltarea conceptelor legate de procesoarele lexicale și sintactice din translatoare</li> </ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>• 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</li> <li>• Înțelegerea relațiilor dintre descrierea formală a automatelor din teoria limbajelor formale și implementările practice ale acestora ca analizoare lexicale și sintactice în translatoare</li> <li>• Cunoașterea claselor de limbaje pentru care se poate implementa un analizor sintactic determinist</li> <li>• Descrierea sintaxei limbajelor de implementat folosind expresii regulate și gramatici</li> <li>• Proiectarea, dezvoltarea și testarea unui proiect software, utilizând unelte software specializate (generatoare de analizoare), pentru a ajunge la un translator pentru un limbaj artificial</li> <li>• Controlul fenomenelor de ambiguitate și nedeterminism (conflicte) care apar la generatoarele de analizoare lexicale și sintactice</li> <li>• Introducere în metode de procesare a limbajului natural</li> </ul>

### 8. Conținuturi

8.1 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 grafică, 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 translațoare, 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.			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
Analizor lexical de C. Expresii regulate în python.	2	Explicații, implementare și testare exemple, exerciții.	
Generatorul de analizare lexicală 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 analizare 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		
Generatorul Yacc: suport pentru valori de tip arbitrar, exemple. Generatorul YACC: translațarea expresiilor aritmetice din infix în postfix	2		
Aplicații 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 descendentă deterministă a unui limbaj ce lucrează cu arbori binari.	2		
Exemple de utilizare a modelelor neuronale de reprezentare a limbajului/ Parsarea recursiv descendentă deterministă a unui limbaj ce lucrează cu liste și 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. <a href="https://www.cs.utexas.edu/users/novak/lexpaper.htm">https://www.cs.utexas.edu/users/novak/lexpaper.htm</a>			
2. <a href="https://www.cs.utexas.edu/users/novak/yaccpaper.htm">https://www.cs.utexas.edu/users/novak/yaccpaper.htm</a>			
3. Indrumator laborator online			
4. Hugging Face <a href="https://huggingface.co/">https://huggingface.co/</a>			

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

### 9. 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 instruește pe studenți cu principiile de bază în proiectarea interpretoparelor și translațoarelor 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).

## 10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	- Abilități de rezolvare a problemelor - Prezență, Activitate	- Test online: Moodle Quiz/ 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:	Titulari	Titlu Prenume NUME	Semnătura
12.06.2023	Curs	Conf. dr. ing. Anca Mărginean	
		Conf. dr. ing. Emil Ștefan Chifu	
	Aplicații	Conf. dr. ing. Anca Mărginean	
		Conf. dr. ing. Emil Ștefan Chifu	

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. Liviu Miclea