

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 Ciclu 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	Proiectarea translaatoarelor				Codul disciplinei	48.2
2.2 Titularul de curs	Conf. dr. ing. Chifu Emil-Ștefan - emil.chifu@cs.utcluj.ro					
2.3 Titularul activităților de seminar / laborator / proiect	Conf. dr. ing. Chifu Emil-Ștefan - emil.chifu@cs.utcluj.ro					
2.4 Anul de studiu	IV	2.5 Semestrul	7	2.6 Tipul de evaluare	E	
2.7 Regimul disciplinei	Categorii formative: <i>DF – fundamentală, DS - de specialitate, DC - complementară</i>					DS
	Opționalitate: <i>DOB - obligatorie, DOP - opțională, DFA - facultativă</i>					DOP

3. Timpul total estimat

3.1 Număr de ore pe săptămână	5	din care:	Curs	2	Seminar	-	Laborator	2	Proiect	1
3.2 Număr de ore pe semestru	70	din care:	Curs	28	Seminar	-	Laborator	28	Proiect	14
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										25
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										15
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										26
(d) Tutoriat										10
(e) Examinări										4
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a))...3.3(f)))				80						
3.5 Total ore pe semestru (3.2+3.4)				150						
3.6 Numărul de credite				6						

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Limbaje formale și translaatoare, Programarea calculatoarelor, Structuri de date și algoritmi
4.2 de competențe	- Cunoștințe de bază de programare și structuri de date (de preferință în limbajele C și Java) - noțiuni de gramatici generative și limbaje formale - Cunoașterea principiilor de bază în proiectarea interpretoarelor și translaatoarelor pentru limbaje artificiale

5. Condiții (acolo unde este cazul)

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

6. Competențele specifice acumulate

6.1 Competențe profesionale	C4 - Îmbunătățirea performanțelor sistemelor hardware, software și de comunicații C4.1 - Identificarea și descrierea elementelor definitorii ale performanțelor sistemelor hardware, software și de comunicații C4.2 - Explicarea interacțiunii factorilor care determină performanțele sistemelor hardware, software și de comunicații C4.3 - Aplicarea metodelor și principiilor de bază pentru creșterea performanțelor sistemelor hardware, software și de comunicații C4.4 - Alegerea criteriilor și metodelor de evaluare a performanțelor sistemelor hardware, software și de comunicații C4.5 - Dezvoltarea de soluții profesionale pentru sisteme hardware, software și de comunicații bazate pe creșterea performanțelor C5 - Proiectarea, gestionarea ciclului de viață, integrarea și integritatea sistemelor hardware, software și de comunicații C5.1 - Precizarea criteriilor relevante privind ciclul de viață, calitatea, securitatea și interacțiunea sistemului de calcul cu mediul și cu operatorul uman C5.2 - Utilizarea unor cunoștințe interdisciplinare pentru adaptarea sistemului informatic în raport cu cerințele domeniului de aplicații C5.3 - Utilizarea unor principii și metode de bază pentru asigurarea securității, siguranței și usurinței în exploatare a sistemelor de calcul C5.4 - Utilizarea adecvată a standardelor de calitate, siguranță și securitate în prelucrarea informațiilor C5.5 - Realizarea unui proiect incluzând identificarea și analiza problemei, proiectarea, dezvoltarea și demonstrând o înțelegere a nevoii de calitate C6 - Proiectarea sistemelor inteligente C6.1 - Descrierea componentelor sistemelor inteligente C6.2 - Utilizarea de instrumente specifice domeniului pentru explicarea și înțelegerea funcționării sistemelor inteligente C6.3 - Aplicarea principiilor și metodelor de bază pentru specificarea de soluții la probleme tipice utilizând sisteme inteligente C6.4 - Alegerea criteriilor și metodelor de evaluare a calității, performanțelor și limitelor sistemelor inteligente C6.5 - Dezvoltarea și implementarea de proiecte profesionale pentru sisteme inteligente
6.2 Competențe transversale	N/A

7. Rezultatele așteptate ale învățării

Cunoștințe	Describe, identifică și sumarizează concepte, modele și metode avansate privind sisteme și infrastructuri informatice complexe, precum și principiile managementului și metodologiei proiectelor informatice, și anume: - Proiectarea în detaliu a unui generator de parser-e sintactice descendente. - Analiza sintactică pentru construirea arborilor sintactici abstracți (AST). - Parcurgerea arborilor AST pentru generarea codului. - Generarea codului într-un compilator: generarea codului din expresii aritmetice, din instrucțiuni condiționale și din bucle. - Automate de analiză sintactică ascendentă SLR(1) și LALR(1).
Aptitudini	Aplică și proiectează soluții informatice integrate ca sisteme software, și anume interpretoare și compilatoare, utilizând metode de analiză, proiectare, implementare și documentare. Mai concret: - Implementează parser-e descendent recursive pentru construirea arborilor sintactici abstracți (AST). - Utilizând infrastructura pentru compilatoare LLVM, generează cod din expresiile aritmetice, din instrucțiuni condiționale, din bucle și din utilizarea variabilelor. - Utilizând toolkit-ul NLTK, implementează analizoare semantice de limbaj natural.

Responsabilitate și autonomie:	Își asumă responsabilitatea pentru calitatea, securitatea și corectitudinea soluțiilor dezvoltate și manifestă autonomie în organizarea activităților de proiectare și învățare, colaborând eficient în echipe și adaptându-se la cerințe și tehnologii noi.
--------------------------------	--

8. Obiectivele disciplinei

8.1 Obiectivul general al disciplinei	- Cunoașterea fazelor translaatoarelor pentru limbajele de programare: analiza lexicală, analiza sintactică și generarea codului. - Cunoașterea fazelor Prelucrării limbajului natural și a modelelor de limbă BERT.
8.2 Obiectivele specifice	- Să cunoască clasele de limbaje pentru care se pot implementa translaatoare și interpretoare eficiente. - Să cunoască regulile de procesare a instrucțiunilor tipice pentru interpretoare. - Utilizând limbajul Prolog, să construiască parser-e DCG pentru limbaj natural. - Să implementeze în platforma NLTK diferitele faze ale prelucrării limbajului natural. - Să definească, să antreneze și să testeze clasificatoare de texte în limbaj natural, utilizând modele de limbă preantrenate BERT.

9. Conținuturi

9.1 Curs	Nr.ore	Metode de predare	Observații
Derivatoare LL: algoritmul derivator LL(1) pentru gramatici BNF extins.	2		
Derivatoare LL: interpretoare și compilatoare descendente recursiv.	2		
Derivatoare LL: derivarea descendentă recursiv în varianta interpretată.	2		
Proiectarea unui generator de parser-e sintactice descendente.	2		
Analiza sintactică pentru construirea arborilor sintactici abstracti (AST).	2		
Parcursirea arborilor AST pentru generarea codului.	2		
Generarea codului într-un compilator: generarea codului din expresii aritmetice, din instrucțiuni condiționale și din bucle.	2		
Rezultate teoretice privind gramaticile LL(k) și LR(k).	2		
Derivatoare LR: stări LR(0), gramatici SLR(1).	2		
Derivatoare LR: gramatici LALR(1).	2		
Derivatoare LR: algoritmul LALR(1).	2		
Derivatoare LR: tranziții deplasare-reducere, eliminarea producțiilor lanț.	2		
Derivatoare LR: compactare tabel LR.	2		
Noțiuni fundamentale de gramatici atributate.	2		
Bibliografie (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)			
1. I.A. Leția, E.Șt. Chifu, Limbaje formale și translaatoare, Ed. Casa cărții de știință, 1998.			
2. W.M. Waite and G. Goos, Compiler Construction, Springer-Verlag, 1994.			
3. A.V. Aho, M.S. Lam, R. Sethi, and J.D. Ullman, Compilers: Principles, Techniques and Tools, Second Edition, Addison-Wesley, 2007.			
9.2.1 Aplicații - laborator	Nr.ore	Metode de predare	Observații
Construirea analizoarelor sintactice descendente recursiv din gramatici BNF extins.	2		N/A
Aplicații descendent recursive (DR): construirea arborilor sintactici abstracti (AST) pentru expresii regulate.	2		

Infrastructura pentru compilatoare LLVM: generator de cod pentru un limbaj imperativ, utilizând AST drept cod intermediar. Generarea de cod din expresii aritmetice.	2	Scurtă prezentare la tablă (cardul didactic), implementare și testare exemple și exerciții pe calculator (studentii).	
Infrastructura pentru compilatoare LLVM: generator de cod pentru un limbaj imperativ, utilizând AST drept cod intermediar. Generarea de cod din instrucțiuni condiționale.	2		
LLVM: generarea de cod din bucle.	2		
LLVM: generarea de cod din utilizarea variabilelor; tabele de simboluri.	2		
LLVM: Variabile și instrucțiuni imbricate. Analiza codului multi-linie.	2		
Gramatici cu clauze definite (DCG) pentru analiza limbajului natural: tratarea ambiguității limbajului natural. Verificarea acordului în limba română, traducerea automată.	2		
Framework-ul NLTK: analiza semantică a limbajului natural cu ajutorul calculului Lambda.	2		
NLTK: cadre (frames) de subcategorizare.	2		
NLTK: utilizarea resursei lexicale FrameNet, etichetarea cu roluri semantice (SRL).	2		
NLTK: structuri de reprezentare a discursului (DRS), rezolvarea anaferei.	2		
NLTK: Gramatici de dependență și parsing de dependență.	2		
NLTK: fazele unui pipeline de prelucrare a limbajului natural: lematizarea, etichetarea părților de vorbire, recunoașterea entităților denumite, utilizarea tezaurului lexical WordNet.	2		
8.2.2 Aplicații - proiect			
Codificarea numerică a textelor în limbaj natural: saci de cuvinte (BoW), TF-IDF și bag of n-grams.	2		
Clasificator de tip analizor de sentimente folosind Logistic Regression (LR).	2		
Categorizarea documentelor utilizând Logistic Regression: antrenare și testare.	2		
Codificarea textelor folosind Word to Vec (word2vec): categorizarea documentelor.	2		
Utilizarea modelelor de limbă preantrenate BERT: analizor de sentimente utilizând Logistic Regression.	2		
Utilizarea BERT: reglajul fin (fine-tuning) al vectorilor preantrenați BERT.	2		
Utilizarea BERT: transfer learning pentru diferite downstream tasks: sumarizare, traducere automată, etichetarea cu roluri semantice (SRL).	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. Online lab manual			
3. Hugging Face https://huggingface.co/			

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 instruește pe studenți cu principiile proiectării și implementării eficiente a interpretoparelor și translaatoarelor 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	- Evaluare pe parcurs în timpul cursurilor, prin dialog cu studenții și ieșiri la tablă în timpul cursurilor - Se organizează consultații înainte de examen, la care se acordă bonus pentru examenul final - Examenul final este examen oral	44%
Seminar	-	-	-
Laborator	- Abilități de rezolvare a problemelor - Prezență, Activitate	Lucrări de laborator: - Evaluarea pe parcurs a activității studenților, la fiecare oră de laborator - Se acordă bonus pentru examenul final	35%
Proiect		Sesiuni de laborator de proiect: - Evaluarea pe parcurs a activității studenților, la fiecare oră de laborator de proiect	21%
Standard minim de performanță: Modelarea unei probleme tipice ingineresti folosind aparatul formal caracteristic domeniului Calcul nota disciplina: 35% laborator + 21% proiect + 44% examen final Conditii de participare la examenul final: Laborator ≥ 5 Conditii de promovare: Nota ≥ 5			

Data completării: 29.04.2026	Titulari	Titlu, prenume / NUME	Semnătura
	Curs	Conf.dr.ing. Emil-Ștefan CHIFU	
	Aplicații	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. Vlad Mureșan