LEHRVERANSTALTUNGSBESCHREIBUNG

1. Angaben zum Programm

1.1 Hochschuleinrichtung	**Babes-Bolyai Universität, Cluj-Napoca**
1.2 Fakultät	Mathematik und Informatik
1.3 Department	Informatik
1.4 Fachgebiet	Informatik
1.5 Studienform	Bachelor
1.6 Studiengang / Qualifikation	Informatik

2. Angaben zum Studienfach

2.1 LV-Bezeichnung (de) (en) (ro)	Formale Sprachen und Compiler Formal languages and compilers Limbaje formale și tehnici de compilare						
2.2 Lehrverantwortlicher – Vorlesung		Dominik Knoll, PhD					
2.3 Lehrverantwortlicher – Seminar		Dominik Knoll, PhD					
2.4 Studienjahr	3	2.5 Semester	5	2.6. Prüfungsform	P	2.7 Art der LV	Pflichtfach

3. Geschätzter Workload in Stunden

3.1 SWS	6	von denen: 3.2 Vorlesung	2	3.3 Seminar/Labor	2+2
3.4 Gesamte Stundenanzahl im Lehrplan	84	von denen: 3.5 Vorlesung	28	3.6 Seminar/Übung	56

Verteilung der Studienzeit:	Std.
Studium nach Handbücher, Kursbuch, Bibliographie und Mitschriften	20
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung	10
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und Essays	20
Tutorien	6
Prüfungen	10
Andere Tätigkeiten:	-
3.7 Gesamtstundenanzahl Selbststudium	66
3.8 Gesamtstundenanzahl / Semester	150
3.9 Leistungspunkte	6

4. Voraussetzungen (falls zutreffend)

4.1 curricular	□ Datenstrukturen und Algorithmen
4.2 kompetenzbezogen	□ Programmierfähigkeiten

5. Bedingungen (falls zutreffend)

6. Spezifische erworbene Kompetenzen

5.1 zur Durchführung der Vorlesung	□ Vorlesungsraum, Beamer, Laptop
5.2 zur Durchführung des Seminars / der Übung	□ Computerraum

Berufliche Kompetenzen	K 4.1 Definieren der Grundkonzepte und Prinzipien der Informatik, sowie der mathematischen Theorien und Modelle K 4.2 Interpretation formaler Modelle der Mathematik und Informatik K 4.3 Identifizierung geeigneter Modelle und Methoden für die Lösung realer Probleme K 4.4 Anwendung der Simulation für die Untersuchung der Verhaltensweise der angewandten Modelle und Bewertung der Ergebnisse K4.5 Einbauen der formalen Modelle in geeignete Anwendungen für spezifische Aufgaben
Transversale Kompetenzen	TK1 Anwendung der Regeln für gut organisierte und effiziente Arbeit, für verantwortungsvolle Einstellung gegenüber der Lehre und der Wissenschaft, für kreative Förderung des eigenen Potentials, mit Rücksicht auf die Prinzipien und Normen der professionellen Ethik TK3 Anwendung von effizienten Methoden und Techniken für Lernen, Informieren und Recherchieren, für das Entwickeln der Fähigkeiten zur praktischen Umsetzung der Kenntnisse, der Anpassung an die Bedürfnisse einer dynamischen Gesellschaft, der Kommunikation in rumänischer Sprache und in einer internationalen Verkehrssprache

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	• Das Erlernen und Verstehen wie man Compiler aufbaut. • Verbesserung der Programmierfähigkeiten.
7.2 Spezifische Ziele der Lehrveranstaltung	• Kenntnisse über den Aufbau eines Compilers. • Aneignen der grundlegenden Begriffe der formalen Sprachen. • Aneignen der grundlegenden Begriffe über Compiler.

8. Inhalt

8.1 Vorlesung	Lehr- und Lernmethode	Anmerkungen
1. Überblick: • Aufbau eines Compilers • Extended Bakus-Naur Form (EBNF)	Vortrag, Erklärung, Debatte, praktische Beispiele	
2. Formale Sprachen: • Sprache, Grammatik, Chomsky-Hierarchie • Reguläre Grammatiken (RG) und Endliche Automaten (EA) • Thomson Konstruktion	Vortrag, Erklärung, Debatte, praktische Beispiele	

3. Reguläre Sprachen: • Reguläre Ausdrücke (RA) • Äquivalenz RA + EA	Vortrag, Erklärung, Debatte, praktische Beispiele	
4. Kontextfreie Sprachen (KFG) • Definition • Parsen und Syntaxbaum	Vortrag, Erklärung, Debatte, praktische Beispiele	
5. Kontextfreie Sprachen (KFG): • Umwandlungen	Vortrag, Erklärung, Debatte, praktische Beispiele	
6. Kontextfreie Sprachen (KFG): • Chomsky Normal Form	Vortrag, Erklärung, Debatte, praktische Beispiele	
7. Parsen: • Representation of the syntax tree • Classification of parsers • Top-Down Parser FIRST und FOLLOWS Mengen	Vortrag, Erklärung, Debatte, praktische Beispiele	
8. Top-Down Parser: • Rekursiver Abstieg • LL(1) Parse Algorithmus • LL(k) Grammatiken	Vortrag, Erklärung, Debatte, praktische Beispiele	
9. Bottom-Up-Parser: • LR Parsen, Shift-Reduce • LR(k) Algorithmus und Methoden	Vortrag, Erklärung, Debatte, praktische Beispiele	
10. Semantische Analyse: • Attributgrammatiken • Symboltabellen Verwaltung	Vortrag, Erklärung, Debatte, praktische Beispiele	
11. Scanner und Parser Generatoren: • Scanner Generator (lex, jflex) • Parser Generator (yacc, cup)	Vortrag, Erklärung, Debatte, praktische Beispiele	
12. Syntax Orientierte Übersetzung, Code Generierung, Optimierung, Register Allokation	Vortrag, Erklärung, Debatte, praktische Beispiele	
13. Turing Maschinen:	Vortrag, Erklärung, Debatte, praktische Beispiele	Je nach Bedarf, werden die Themen in

14. Automatentheorie	Vortrag, Erklärung, Debatte, praktische Beispiele	den letzten zwei Vorlesungen erweitert oder ausgelassen
Literatur [1] K.D. COOPER, L. TORCZON - Engineering a Compiler, Elsevier Science & Technology, 2011. [2] A.V. AHO, D.J. ULLMAN - Principles of compiler design, Addison-Wesley, 1978. [3] WAGENKNECHT, HIELSCHER M., Formale Sprachen, abstrakte Automaten und Compiler, Vieweg Teubner, 2009.		

8.2 **Seminar**	Lehr- und Lernmethode	Anmerkungen
1. RA in der Praxis	Beispiele, Diskussionen, Teamarbeit	
2-5. Übungen zu: • NEA und DEA • reguläre Ausdrücke • reguläre Sprachen • entsprechende Algorithmen	Beispiele, Diskussionen	
6-12. Übungen zu: • kontextfreie Grammatiken • reguläre Grammatiken • LL(k)-Grammatiken • LR(k)-Grammatiken	Beispiele, Diskussionen	
13-14. Turing Maschinen und/oder Automatentheorie	Beispiele, Diskussionen	Je nach Bedarf, werden diese Themen erweitert oder ausgelassen
Literatur [1] K.D. COOPER, L. TORCZON - Engineering a Compiler, Elsevier Science & Technology, 2011. [2] A.V. AHO, D.J. ULLMAN - Principles of compiler design, Addison-Wesley, 1978. [3] C. WAGENKNECHT, HIELSCHER M., Formale Sprachen, abstrakte Automaten und Compiler, Vieweg Teubner, 2009.		
8.3 **Labor**		
1. Einfaches Mini-Programm	Beispiele, Diskussionen, Teamarbeit	
2. Grammatik einer Mini-Sprache	Beispiele, Diskussionen, Teamarbeit	
3. Symboltabelle	Beispiele, Diskussionen, Teamarbeit	
4. Endliche Automaten	Beispiele, Diskussionen, Teamarbeit	

5. Scanner	Beispiele, Diskussionen, Teamarbeit	
6. Erweiterter Scanner	Beispiele, Diskussionen, Teamarbeit	
7. KFG der Mini-Sprache	Beispiele, Diskussionen, Teamarbeit	
8. Backtracking-freie Grammatik	Beispiele, Diskussionen, Teamarbeit	
9-10. Parser für Ausdrücke	Beispiele, Diskussionen, Teamarbeit	
11. Anwendung von lex: Generierter Scanner für Mini-Sprache	Beispiele, Diskussionen, Teamarbeit	
12. Anwendung von yacc: Generierter Parser für Mini-Sprache	Beispiele, Diskussionen, Teamarbeit	

Literatur:

[1] K.D. COOPER, L. TORCZON - Engineering a Compiler, Elsevier Science & Technology, 2011.
[2] C. WAGENKNECHT, HIELSCHER M., Formale Sprachen, abstrakte Automaten und Compiler, Vieweg Teubner, 2009.

9. Verbindung der Inhalte mit den Erwartungen der Wissensgemeinschaft, der Berufsverbände und der für den Fachbereich repräsentativen Arbeitgeber

Die besprochene Theorie wird aus der Perspektive des Aufbaus eines Compilers besprochen. Der Inhalt dient also als theoretische Grundlage für die Kenntnisse der Informatiker in Software-Unternehmen.

Formale Sprachen und Automaten dienen als Basis für Berechenbarkeitstheorie und Komplexitätstheorie.

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	Grundkenntnisse	Aktive Mitarbeit, Schriftliche Prüfung	5% + 30%
10.5 Seminar	Algorithmen-Anwendung	Quiz	15%
10.6 Labor	Praktische Umsetzung	Labor Arbeiten	50%

10.7 Minimale Leistungsstandards

- Fähigkeit aus regulären Ausdrücken einen minimalen deterministischen Automaten zu erhalten.
- Fähigkeit eine LL(1)-Grammatik zu erkennen und backtrackingfreies Parsen durchzuführen.
- Fähigkeit eine LR(1)-Tabellen zu erzeugen und backtrackingfreies Parsen durchzuführen.
- Fähigkeit Grammatiken und Automaten anhand der Chomsky-Hierarchie zu erklären.
- Für die Laboraufgaben ist eine Mindestbewertung von 5 zu erreichen.
- Für die schriftliche Prüfung ist eine Mindestbewertung von 5 zu erreichen.

Ausgefüllt am:	Vorlesungsverantwortlicher	Seminarverantwortlicher
1. Februar 2021	Dominik Knoll, PhD	Dominik Knoll, PhD

Genehmigt im Department am: Departmentdirektor

.....