Übersetzerbau

Compiler Construction

Modulnummer:

Bachelor

- Pflicht ☐
- Winf-Schwerpunkt-Pflicht ☐
- Winf-Schwerpunkt-Wahlpflicht ☐
- Winf-Wahl ☐

Schwerpunkt

- Computational Finance ☐
- E-Business ☐
- IT-Management ☐
- Logistik ☐

Anzahl der SWS

V	UE	K	S	Prak.	Proj.	Σ
3	1	0	0	0	0	4

Kreditpunkte: 6

Turnus

i. d. R. angeboten alle 2 Semester

Formale Voraussetzungen: -

Inhaltliche Voraussetzungen: Theoretische Informatik 1, Theoretische Informatik 2

Vorgesehenes Semester: ab 5. Semester

Sprache: Deutsch

Ziele:

- Prinzipien der Strukturierung von Übersetzern und Interpretern verstehen und anwenden können
- Konzepte und Methoden der lexikalischen, syntaktischen und kontextuellen (statisch semantischen) Analyse verstehen, anwenden, auf die Implementierung konkreter Sprachen übertragen, beurteilen und bewerten können
- Prinzipien der Übersetzung von imperativen und objektorientierten Programmiersprachen in Maschinencode verstehen, auf die Implementierung konkreter Konzepte übertragen und die Qualität des Codes beurteilen können.
- Prinzipien der Codeerzeugung (Registerzuteilung, Instruktionsauswahl, globale und lokale Optimierung) verstehen können
- selbstständig und in kleinen Teams Wissen und Verständnis erwerben und darstellen können.

Inhalte:

- Implementierung von Programmiersprachen mit Interpretern, und Übersetzern.
- Strukturierung von Übersetzern: Plattform(un)abhängigkeit, Bootstrap, Phasen.
- Lexikalische Analyse: reguläre Definitionen, endliche Automaten, Symboltabellen, Benutzung von flex.
- Syntaxanalyse: kontextfreie Grammatiken, ab- und aufsteigendes Parsieren, Baumaufbau, Fehlerbehandlung, Benutzung von bison.
- Kontext-Analyse: Attributgrammatiken, Auswerter, Vereinbarungstabellen.
- Transformation von imperativen und objektorientierten Programmen in abstrakten Maschinencode.
- Grundzüge der Codeerzeugung für konkrete Maschinen: globale Optimierung, Registerzuteilung, Instruktionsauswahl, lokale Optimierung.

In der Übung Anwendung der in der Vorlesung erworbenen Kenntnisse und Fähigkeiten auf spezifische Konstrukte von Programmiersprachen.

Insbesondere werden folgende theoretisch/methodische Grundlagen behandelt:

- Theorie der regulären und kontextfreien Sprachen
- Algorithmen zur Konstruktion von deterministischen endlichen Automaten für reguläre Definitionen
- Theorie des LL(k) und LR(k)-Parsierens, mit automatischer Fehlerbehandlung
- Methoden der Grammatikdefinition, -transformation und -disambiguierung.
- Theorie der Zweistufengrammatiken und Attributgrammatiken
- Algorithmen zum Erzeugens von Auswertern für Attributgrammatiken
- Methoden der Spezifikation von abstrakten Datentypen, für Bezeichnertabellen und Vereinbarungstabellen
- Methodik der rekursiven Syntax-orientierten Definition für die Transformation von Syntaxbäumen in abstrakten Maschinencode

Unterlagen (Skripte, Literatur, Programme usw.):

- A.V. Aho, M. S. Lam, R. Sethi, J.D. Ullman. Compilers - Prinzipien, Techniken und Werkzeuge, zweite Auflage, Bonn: Pearson Education Deutschland (2008).
- R. Wilhelm, D. Maurer. Übersetzerbau: Theorie - Konstruktion - Generierung. Berlin: Springer, 2. Auflage (1997).

Weiteres Lehrmaterial ist auf der Webseite des Veranstaltung zu finden:

- Folienkopien
- Übungsaufgaben.

Übersetzer-Werkzeuge lex/flex, yacc/bison stehen im Rechnernetz des Studiengangs zur Verfügung.

Form der Prüfung:
i.d.R. mündliche Prüfung

Arbeitsaufwand		
	Präsenz	56 h
	Übungsbetrieb/Prüfungsvorbereitung	124 h
	Summe	180 h

Lehrende:	Verantwortlich:
Dr. B. Hoffmann	Dr. B. Hoffmann