

Inhalt

1.	Einleitung	1
1.1	Höhere-Programmiersprachen	1
1.2	Implementierung von Programmiersprachen	2
1.2.1	Interpreter	2
1.2.2	Übersetzer	3
1.2.3	Reale und abstrakte Maschinen	4
2	Übersetzung imperativer Programmiersprachen	7
2.1	Sprachkonzepte und ihre Übersetzung	7
2.2	Die Architektur der P-Maschine	8
2.3	Wertzuweisungen und Ausdrücke	9
2.4	Bedingte und iterative Anweisungen, Anweisungsfolgen	13
2.5	Speicherbelegung für Variablen einfachen Typs	17
2.6	Speicherbelegung für Felder	18
2.6.1	Statische Felder	18
2.6.2	Dynamische Felder	23
2.7	Speicherbelegung für Verbunde	25
2.8	Zeiger und dynamische Speicherbelegung	27
2.9	Prozeduren	31
2.9.1	Speicherorganisation für Prozeduren	36
2.9.2	Adressierung von Variablen	39
2.9.3	Berechnung der Adreßumgebungen	42
2.9.4	Prozedureintritt und Prozedurverlassen	45
2.9.5	Parameterübergabe	48
2.9.6	Zugriff auf Variablen und formale Parameter	52
2.9.7	Formale Prozeduren als Parameter	52
2.10	Hauptprogramm	55
2.11	Übungen	57
2.12	Literaturhinweise	61
3	Übersetzung funktionaler Programmiersprachen	63
3.1	Sprachtyp und einleitendes Beispiel	63
3.2	LaMa, eine einfache funktionale Programmiersprache	72
3.3	Einführung in die Übersetzung von LaMa	78
3.3.1	Die Übersetzungsfunktionen	81
3.4	Umgebungen und Bindungen	82

3.5	Die Architektur der MaMa	84
3.5.1	Der Keller der MaMa	85
3.5.2	Die Halde der MaMa	86
3.6	Kellerverwaltung und Adressierung	88
3.6.1	Adressierung von Namen in der MaMa	89
3.6.2	Aufbau von Bindungen	91
3.7	Befehlsvorrat und Übersetzung	92
3.7.1	Programmausdrücke	92
3.7.2	Einfache Ausdrücke	92
3.7.3	Angewandte Vorkommen von Variablen	94
3.7.4	Funktionsdefinitionen	95
3.7.5	Funktionsanwendungen	97
3.7.6	Aufbau und Auswertung von Abschlüssen	100
3.7.7	Letrec-Ausdrücke und lokale Variablen	103
3.8	Implementierung von Listen	105
3.9	Übungen	110
3.10	Literaturhinweise	112
4	Übersetzung logischer Programmiersprachen	115
4.1	Logische Programmiersprachen	115
4.2	Prädikatenlogische Grundlagen	119
4.3	Unifikation	121
4.4	Ausführung von logischen Programmen	125
4.5	Prolog	134
4.5.1	Beweisbäume	136
4.5.2	Umgebungen	138
4.6	Prolog: Abstrakte Maschine und Übersetzung	140
4.6.1	Entwicklung der Prolog Implementierung	140
4.6.2	Die Architektur der WiM	141
4.7	Übersetzung von Prolog	143
4.7.1	Ziele	143
4.7.2	Kopfterme	146
4.7.3	Übersetzung von Klauseln	151
4.7.4	Zurücksetzen (Backtracking)	153
4.7.5	Prozeduren, Programme und Anfragen	156
4.7.6	Ein Beispiel	157
4.8	Effizienzverbesserungen	162
4.8.1	Argumentregister	162
4.8.2	Rücksetzrahmen	163
4.8.3	Verkleinerung der lokalen Umgebung	165
4.8.4	Letztes Ziel und Endrekursion	166
4.8.5	Indizieren von Klauseln	167
4.9	Übungen	168
4.10	Literaturhinweise	171

5	Struktur von Übersetzern	173
5.1	Übersetzerteilaufgaben	174
5.2	Die lexikalische Analyse	175
5.3	Der Sieber	175
5.4	Die syntaktische Analyse	176
5.5	Die semantische Analyse	176
5.6	Die maschinenunabhängige Optimierung	178
5.7	Die Adreßzuordnung	180
5.8	Die Erzeugung des Zielprogramms	180
5.9	Die maschinenabhängige Codeverbesserung	182
5.10	Reale Übersetzerstrukturen	182
5.11	Formale Spezifikation und Generierung von Übersetzermoduln	184
5.12	Literaturhinweise	185
6	Lexikalische Analyse	187
6.1	Die Aufgabe der lexikalischen Analyse	187
6.2	Theoretische Grundlagen	188
6.3	Sprache zur Spezifikation der lexikalischen Analyse	199
6.3.1	Zeichenklassen	200
6.3.2	Folgen von regulären Definitionen	200
6.3.3	Nichtrekursive Klammerung	201
6.4	Die Generierung eines Scanners	202
6.4.1	Zeichenklassen	202
6.4.2	Folgen von regulären Definitionen	203
6.4.3	Implementierung des <i>allbut</i> -Konstrukts	205
6.4.4	Die Darstellung eines Scanners	209
6.5	Der Sieber	210
6.5.1	Die Erkennung von Schlüsselwörtern	211
6.5.2	Scanner mit Aufrufchnittstelle zum Sieber	212
6.5.3	Symbolklassen	212
6.6	Übungen	214
6.7	Literaturhinweise	215
7	Syntaktische Analyse	217
7.1	Die Aufgabe der syntaktischen Analyse	217
7.2	Theoretische Grundlagen	221
7.2.1	Kontextfreie Grammatiken	221
7.2.2	Kellerautomaten	229
7.2.3	Der Item-Kellerautomat einer kontextfreien Grammatik	232
7.2.4	Grammatikflußanalyse	238
7.2.5	Einige Begriffe und Fakten aus der Verbandstheorie	242
7.2.6	Effizientere Analyse auf dem Grammatikgraphen	246
7.2.7	Ein lineares Verfahren	249
7.2.8	<i>FIRST</i> und <i>FOLLOW</i>	250

7.2.9	Der Spezialfall $FIRST_1$ und $FOLLOW_1$	255
7.2.10	Reine Vereinigungsprobleme	257
7.3	Top down-Syntaxanalyse	261
7.3.1	Einführung	261
7.3.2	Top down-Syntaxanalyse mit Zurücksetzen in Prolog	262
7.3.3	LL(k): Definition, Beispiele, Eigenschaften	266
7.3.4	(Starke) LL(k)-Parser	274
7.3.5	LL-Parser für erweiterte kontextfreie Grammatiken	279
7.3.6	Fehlerbehandlung in LL(k)-Parsern	288
7.4	Bottom up-Syntaxanalyse	302
7.4.1	Einführung	302
7.4.2	Bottom up-Analyse mit Zurücksetzen in Prolog	303
7.4.3	LR(k)-Analysatoren	306
7.4.4	LR(k): Definition, Eigenschaften, Beispiele	317
7.4.5	LR(k)-Parser	320
7.4.6	Fehlerbehandlung in LR-Parsern	331
7.4.7	Scannergenerierung mit LR-Techniken	338
7.5	Übungen	342
7.6	Literaturhinweise	348
8	Semantische Analyse	351
8.1	Aufgabe der semantischen Analyse	351
8.1.1	Gültigkeits- und Sichtbarkeitsregeln	356
8.1.2	Überprüfung der Kontextbedingungen	358
8.1.3	Überladung von Bezeichnern	364
8.1.4	Polymorphismus	367
8.2	Attributgrammatiken	372
8.2.1	Die Semantik einer Attributgrammatik	374
8.2.2	Eine Notation für Attributgrammatiken	375
8.3	Einige Attributgrammatiken	376
8.4	Die Generierung von Attributauswertern	382
8.4.1	Attributabhängigkeiten	383
8.4.2	Attributauswertung	397
8.4.3	Besuchsorientierte Auswerter	402
8.4.4	l-geordnete Attributgrammatiken	409
8.4.5	Absolut zyklensfreie Attributgrammatiken	410
8.4.6	Parsergesteuerte Attributauswertung	412
8.5	Übungen	421
8.6	Literaturhinweise	423
9	Abstrakte Interpretation	425
9.1	Einführung	425
9.1.1	Beispiel 1: Rechnen mit Resten	426
9.1.2	Beispiel 2: Neunerprobe	427

9.1.3	Beispiel 3:Vorzeichenregeln	428
9.1.4	Bestandteile einer abstrakten Interpretation	429
9.1.5	Denotationell basierte abstrakte Interpretation	430
9.1.6	Operationell basierte abstrakte Interpretation	431
9.2	Grundlagen	432
9.2.1	Algebren	432
9.2.2	Partielle Ordnungen, Verbände	438
9.3	Abstrakte Interpretation (denotationelle Semantik)	444
9.3.1	Die denotationelle Methode	444
9.3.2	Grundprinzip der abstrakten Interpretation	447
9.3.3	Konstruktion abstrakter Interpretationen	449
9.3.4	Verwendung von Hilfssemantiken	453
9.3.5	Fallbeispiel: Striktheitsanalyse	461
9.4	Abstrakte Interpretation (operationelle Semantik)	472
9.4.1	Die operationelle Methode	472
9.4.2	Grundprinzip der abstrakten Interpretation	476
9.4.3	Konstruktion abstrakter Interpretationen	481
9.4.4	Verwendung von Hilfssemantiken	499
9.5	Übungen	506
9.6	Literaturhinweise	507
10	Bäume: Mustererkennung und Analyse	509
10.1	Programmtransformationen	509
10.1.1	Effizienzsteigernde Programmtransformationen	510
10.1.2	Standardisierende Transformationen	513
10.2	Codeselektion	514
10.3	Das Mustererkennungsproblem	519
10.4	Das Baumanalyseproblem	520
10.5	Endliche Baumautomaten	523
10.6	Die Generierung von Baumanalysatoren	529
10.7	Baumautomaten mit Kosten	531
10.8	Implementierung	536
10.9	Übungen	537
10.10	Literaturhinweise	537
11	Codeerzeugung	539
11.1	Abstrakte und reale Maschinen	539
11.1.1	Sprachspezifische abstrakte Maschinen	539
11.1.2	Universelle reale Maschinen	540
11.1.3	Codeerzeugung für abstrakte und reale Maschinen	543
11.2	Klassifikation von Architekturen	546
11.2.1	CISC (Complex Instruction Set Computer)	546
11.2.2	RISC (Reduced Instruction Set Computer)	547
11.2.3	Intraprozessorparallelität	548

11.3	Programmdarstellungen	550
11.4	Codeerzeugung, integrierte Verfahren	554
11.4.1	Optimale Auswertungsordnung	554
11.4.2	Dynamisches Programmieren	558
11.5	Registerzuteilung durch Graphfärbung	561
11.6	Instruktionsanordnung	563
11.6.1	Abhängigkeitsgraphen für Basisblöcke	563
11.6.2	Befehlsfließband	571
11.6.3	Lange Befehlswörter	573
11.6.4	Realistische VLIW-Rechner	578
11.7	Übungen	583
11.8	Literaturhinweise	584
	Literatur	587
	Index	599