

Ralf Hartmut Güting
Martin Erwig

Übersetzerbau

Techniken, Werkzeuge,
Anwendungen

Mit 103 Abbildungen



Springer

Inhaltsverzeichnis

Einführung	1
1.1 Anwendungsgebiete	1
1.2 Übersetzungsphasen	3
1.3 Die Systemumgebung des Compilers	11
1.4 Compiler und Interpreter, reale und abstrakte Maschinen	13
1.5 Werkzeuge	15
1.6 Struktur des Buches	17
1.7 Literaturhinweise	17
 Lexikalische Analyse	 19
2.1 Beschreibung von Token durch reguläre Ausdrücke	20
2.2 Beschreibung von Token durch Zustandsdiagramme	23
2.3 Direkte Implementierung eines Scanners	28
2.4 Implementierung eines Scanners mit Lex	33
2.5 Aufgaben	37
2.6 Literaturhinweise	40
 Syntaxanalyse	 41
3.1 Kontextfreie Grammatiken und Syntaxbäume	43
3.2 Top-down-Analyse	47
3.2.1 Das Prinzip der Top-down-Analyse	47
3.2.2 LL(fc)-Grammatiken	57
3.2.3 Berechnung von FIRST- und FOLLOW-Mengen, Modifikation von Grammatiken	61
3.2.4 Implementierung eines vorgeifenden Analysators mit Analysetabelle	69
3.2.5 Implementierung eines vorgeifenden Analysators durch rekursiven Abstieg	74

3.3	Bottom-up-Analyse	77
3.3.1	Das Prinzip der Bottom-up-Analyse	78
3.3.2	Operator-Vorranganalyse	83
3.3.3	LR-Analyse	88
3.3.4	Yacc: Ein Parsergenerator	109
3.4	Aufgaben	114
3.5	Literaturhinweise	117
4	Syntaxgesteuerte Übersetzung	119
4.1	Attributierte Grammatik, syntaxgesteuerte Definition	120
4.2	L-attributierte Definition, Übersetzungsschema	125
4.3	Top-down-Übersetzung	128
4.4	Bottom-up-Übersetzung	134
4.5	Aufgaben	136
4.6	Literaturhinweise	140
5	Übersetzung einer Dokument-Beschreibungssprache	141
5.1	Integration von Programmen und Dokumentation	141
5.2	Die Quellsprache: PD-Texte	143
5.2.1	Absatzformate	144
5.2.2	Zeichenformate	147
5.2.3	Sonderzeichen	148
5.3	Die Zielsprache: LATEX	149
5.4	Entwurf des Übersetzers	151
5.4.1	Systemfunktion und Umgebung	152
5.4.2	Text-Datenstrukturen	153
5.4.3	Datenstrukturen für benutzerdefinierte Formate und Zeichen	156
5.4.4	Lexikalische Symbole und Grammatik	156
5.5	Literaturhinweise	166
6	Übersetzung imperativer Programmiersprachen	167
6.1	Speicherorganisation und Laufzeitsystem	167
6.1.1	Anforderungen	167
6.1.2	Speicheraufteilung	172
6.2	3-Adreß-Code: Eine Zwischensprache	178
6.2.1	Zwischendarstellungen	178
6.2.2	Eine abstrakte Maschine für 3-Adreß-Code	182
6.3	Übersetzung in 3-Adreß-Code	191
6.3.1	Deklarationen	192
6.3.2	Zuweisungen und Ausdrücke	202
6.3.3	Kontrollstrukturen und boolesche Ausdrücke	209
6.3.4	Prozedur- und Funktionsaufrufe	213
6.4	Aufgaben	215
6.5	Literaturhinweise	216

7 Übersetzung funktionaler Programmiersprachen	219
7.1 ML	220
7.2 Polymorphe Typsysteme und Typinferenz	227
7.2.1 Bestimmung von Typen	228
7.2.2 Ein polymorphes Typinferenzsystem	230
7.2.3 Automatische Typinferenz	235
7.3 Implementierung durch Interpretation	240
7.4 Implementierung durch Übersetzung	243
7.4.1 Die SECD-Maschine	243
7.4.2 Übersetzung von ML in SECD-Code	246
7.4.3 Behandlung von Rekursion	248
7.5 Aufgaben	250
7.6 Literaturhinweise	252
8 Codeerzeugung und Optimierung	255
8.1 Ein Überblick über Optimierungsverfahren	256
8.1.1 Basisblöcke und Flußgraphen	257
8.1.2 Algebraische Optimierung	260
8.1.3 Maschinenunabhängige Optimierung	261
8.1.4 Maschinenabhängige Optimierung	267
8.2 Datenflußanalyse	270
8.2.1 Datenflußgleichungen	272
8.2.2 Das Lösen von Datenflußgleichungen	275
8.3 Codeerzeugung	280
8.3.1 Maschinenmodell	280
8.3.2 Codeerzeugung für Basisblöcke	281
8.4 Aufgaben	285
8.5 Literaturhinweise	288
Anhang A. Das PD-System	289
Anhang B. File PDNestedTexth	321
Anhang C. File PDNestedTexth.tex	323
Anhang D. Übersetzung von Parameterlisten in	
Prozedurdeklarationen	327
Lösungen zu den Aufgaben im Text	329
Literaturverzeichnis	355
Index	361