

Inhalt

Materialien zum Buch	12
Vorwort	13

1 Einleitung 15

1.1 Compiler und Sprache	15
1.2 Aufbau dieses Buches	19

2 Grundbegriffe der Programmiersprachen 29

2.1 Paradigmen	30
2.1.1 Prozedurale Programmierung	31
2.1.2 Funktionale Programmierung	33
2.1.3 Objektorientierte Programmierung	34
2.1.4 Logikbasierte Sprachen	35
2.2 Konzepte der Programmiersprachen	37
2.2.1 Programm	37
2.2.2 Literale	38
2.2.3 Operatoren und Trennzeichen	39
2.2.4 Schlüsselwörter (Keywords)	40
2.2.5 Bezeichner (Identifizier)	41
2.2.6 Gültigkeitsbereiche	43
2.2.7 Lebensdauer	46
2.2.8 Typen	47
2.2.9 Weitere Merkmale von Typsystemen	56
2.2.10 Typumwandlungen	56
2.2.11 Ausdrücke	59
2.2.12 Anweisungen	60
2.2.13 Unterprogramme	61
2.3 Die Beispielsprache SPL	63
2.3.1 Trennzeichen	65
2.3.2 Kommentare	66

2.3.3	Literale	66
2.3.4	Typen	67
2.3.5	Variablen	68
2.3.6	Ausdrücke	69
2.3.7	Prozeduren	71
2.3.8	Anweisungen	72
2.3.9	Das Programm	75
2.4	Zusammenfassung	76
2.5	Übungsaufgaben	77
2.5.1	Funktionales Paradigma	77
2.5.2	Logikorientiertes Paradigma	77
2.5.3	Prozedurales Paradigma	77
2.5.4	Gültigkeitsbereiche	78
2.5.5	SPL	78

3 Lexikalische Analyse 79

3.1	Einleitung	79
3.2	Lexikalische Elemente	80
3.3	Reguläre Ausdrücke	82
3.4	Endliche Automaten	90
3.4.1	Nichtdeterministische Automaten	93
3.4.2	Elimination von ϵ -Übergängen	99
3.4.3	Deterministische Automaten	103
3.4.4	Minimierung von DEAs	109
3.5	Scanner-Generatoren	114
3.5.1	Lex bzw. Flex	114
3.5.2	JFlex	124
3.6	Zusammenfassung	129
3.7	Übungen	129
3.7.1	Reguläre Ausdrücke	129
3.7.2	Reguläre Sprachen	130
3.7.3	Nichtdeterministische Automaten	130
3.7.4	Deterministische Automaten	130

3.7.5	Minimierung von endlichen Automaten	132
3.7.6	Vervollständigung des Codes	132
4	Syntaxanalyse	133
4.1	Einleitung	133
4.2	Grammatiken	135
4.3	Pumping-Lemma für reguläre Sprachen	143
4.4	Backus-Naur-Form	146
4.5	Ableitungsbäume	148
4.5.1	Ableitungsbäume	148
4.5.2	Mehrdeutigkeit	149
4.5.3	Präzedenzen	151
4.6	Top-Down-Parser	153
4.6.1	Rekursiver Abstiegs-Parser	155
4.6.2	Grammatiktransformationen	158
4.6.3	LL(1)-Parser	160
4.7	Bottom-Up-Parser	176
4.7.1	LR(0)-Parser	178
4.7.2	SLR(1)-Parser	190
4.7.3	LR(1)-Parser	194
4.7.4	LALR(1)-Parser	197
4.8	Fehlerbehandlung	200
4.9	Parsergeneratoren	201
4.9.1	Yacc/Bison	201
4.9.2	CUP	210
4.9.3	ANTLR	216
4.10	Zusammenfassung	220
4.11	Übungen	222
4.11.1	Grammatiken	222
4.11.2	First- und Follow-Mengen	223
4.11.3	LL(1)-Parser	223
4.11.4	LR(0)-Parser	224
4.11.5	SLR(1)-Parser	224

4.11.6	LR(1)-Parser	224
4.11.7	LALR(1)-Parser	224
4.11.8	Parsergeneratoren	225

5 Abstrakter Syntaxbaum 227

5.1	Einleitung	227
5.2	Attributierte Grammatiken	229
5.3	Erzeugung des AST für SPL	237
5.4	Zusammenfassung	252
5.5	Übungen	253
5.5.1	Erweiterungen	253
5.5.2	ANTLR	253

6 Semantische Analyse 255

6.1	Einleitung	255
6.2	Namensanalyse	257
6.2.1	Symboltabellen	258
6.2.2	Das Visitor-Pattern	267
6.2.3	Typdeklarationen	273
6.2.4	Variablendeklarationen	280
6.2.5	Prozedurdeklarationen	281
6.3	Typanalyse	284
6.3.1	Typanalyse für Ausdrücke	286
6.3.2	Typanalyse für Anweisungen	292
6.4	Semantische Analyse komplett	296
6.5	Vorgehen	297
6.6	Zusammenfassung	298
6.7	Übungen	300
6.7.1	Typen	300
6.7.2	Symboltabelle	301
6.7.3	Typanalyse	301

7	Variablenallokation	303
7.1	Einleitung	303
7.2	Aktivierungsrahmen	305
7.2.1	Aufrufargumente	311
7.2.2	Lokale Variablen	315
7.2.3	Sichern der Register	317
7.2.4	Beispiel für Speicherallokation	318
7.3	Umsetzung im SPL-Compiler	320
7.4	Dynamische Speicherverwaltung	322
7.4.1	Explizite Deallokation	324
7.4.2	Implizite Deallokation	325
7.5	Erweiterungen für andere Sprachen	328
7.5.1	Zugriff auf Variablen eines umgebenden Gültigkeitsbereichs	328
7.5.2	Funktionen	331
7.5.3	Weitere Datentypen	333
7.6	Zusammenfassung	333
7.7	Übungen	334
7.7.1	AllocatorVisitor	334
7.7.2	Aktivierungsrahmen	335
7.7.3	Implementierung	335
8	Codegenerierung	337
8.1	Einleitung	337
8.2	Ziel-Hardware	338
8.2.1	RISC versus CISC	339
8.3	ECO32	339
8.3.1	Unbedingte Sprungbefehle	341
8.3.2	Befehle zum Speicherzugriff	342
8.3.3	Rechenbefehle	343
8.3.4	Sprungmarken (Labels)	344
8.3.5	Bedingte Sprünge	344

8.4	Codemuster	346
8.4.1	Ausdrücke	347
8.4.2	Zuweisungen	353
8.4.3	If-Anweisung	355
8.4.4	While-Schleifen	357
8.4.5	Zusammengesetzte Anweisung	357
8.4.6	Prozeduren	358
8.4.7	Prozeduraufrufe	359
8.4.8	Beispiel	360
8.4.9	Andere Anweisungstypen	363
8.4.10	Assembler-Direktiven	363
8.4.11	Post-Processing	364
8.5	Umsetzung im SPL-Compiler	365
8.6	Zusammenfassung	366
8.7	Übungen	368
8.7.1	Weitere Sprachkonstrukte	368
8.7.2	Auswertung von Ausdrücken	368
8.7.3	Codegenerierung für Anweisungen	369
8.7.4	Implementierung	369

9 Optimierung 371

9.1	Einleitung	371
9.2	Grundlagen für die Optimierung	374
9.3	Kontrollflussanalyse	376
9.4	Datenflussanalyse	385
9.5	Lokale und globale Optimierungen	391
9.6	Schleifenoptimierungen	394
9.7	Sonstige Optimierungen	398
9.7.1	Elimination von Endrekursion	399
9.7.2	Inlining	400
9.7.3	Leaf Routine Optimization	401
9.7.4	Registeroptimierung	403
9.8	Zusammenfassung	407

9.9	Übungen	409
9.9.1	Kontrollflussanalyse	409
9.9.2	Datenflussanalyse	410
10	Ausblick	411
<hr/>		
10.1	AOT und JIT	411
10.2	Forschungsfelder im Compilerbau	412
	Literaturverzeichnis	415
	Index	423