

# Inhalt

Materialien zum Buch .....	12
Vorwort .....	13
<b>1 Einleitung</b> .....	<b>15</b>
<hr/>	
1.1 Compiler und Sprache .....	15
1.2 Aufbau dieses Buches .....	19
<b>2 Grundbegriffe der Programmiersprachen</b> .....	<b>29</b>
<hr/>	
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
<b>2.4</b>	<b>Zusammenfassung</b> .....	76
<b>2.5</b>	<b>Übungsaufgaben</b> .....	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
<b>3</b>	<b>Lexikalische Analyse</b> .....	79
<b>3.1</b>	<b>Einleitung</b> .....	79
<b>3.2</b>	<b>Lexikalische Elemente</b> .....	80
<b>3.3</b>	<b>Reguläre Ausdrücke</b> .....	82
<b>3.4</b>	<b>Endliche Automaten</b> .....	90
3.4.1	Nichtdeterministische Automaten .....	93
3.4.2	Elimination von $\epsilon$ -Übergängen .....	99
3.4.3	Deterministische Automaten .....	103
3.4.4	Minimierung von DEAs .....	109
<b>3.5</b>	<b>Scanner-Generatoren</b> .....	114
3.5.1	Lex bzw. Flex .....	114
3.5.2	JFlex .....	124
<b>3.6</b>	<b>Zusammenfassung</b> .....	129
<b>3.7</b>	<b>Übungen</b> .....	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
<b>4</b>	<b>Syntaxanalyse</b> .....	<b>133</b>
<b>4.1</b>	<b>Einleitung</b> .....	<b>133</b>
<b>4.2</b>	<b>Grammatiken</b> .....	<b>135</b>
<b>4.3</b>	<b>Pumping-Lemma für reguläre Sprachen</b> .....	<b>143</b>
<b>4.4</b>	<b>Backus-Naur-Form</b> .....	<b>146</b>
<b>4.5</b>	<b>Ableitungsbäume</b> .....	<b>148</b>
4.5.1	Ableitungsbäume .....	148
4.5.2	Mehrdeutigkeit .....	149
4.5.3	Präzedenzen .....	151
<b>4.6</b>	<b>Top-Down-Parser</b> .....	<b>153</b>
4.6.1	Rekursiver Abstiegs-Parser .....	155
4.6.2	Grammatiktransformationen .....	158
4.6.3	LL(1)-Parser .....	160
<b>4.7</b>	<b>Bottom-Up-Parser</b> .....	<b>176</b>
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
<b>4.8</b>	<b>Fehlerbehandlung</b> .....	<b>200</b>
<b>4.9</b>	<b>Parsergeneratoren</b> .....	<b>201</b>
4.9.1	Yacc/Bison .....	201
4.9.2	CUP .....	210
4.9.3	ANTLR .....	216
<b>4.10</b>	<b>Zusammenfassung</b> .....	<b>220</b>
<b>4.11</b>	<b>Übungen</b> .....	<b>222</b>
4.11.1	Grammatiken .....	222
4.11.2	First- und Follow-Mengen .....	223
4.11.3	LL(1)-Parser .....	223
4.11.4	SLR(1)-Parser .....	224
4.11.5	LR(1)-Parser .....	224

4.11.6	LALR(1)-Parser .....	224
4.11.7	Parsergeneratoren .....	224

## **5 Abstrakter Syntaxbaum** 225

---

<b>5.1</b>	<b>Einleitung</b> .....	225
<b>5.2</b>	<b>Attributierte Grammatiken</b> .....	227
<b>5.3</b>	<b>Erzeugung des AST für SPL</b> .....	235
<b>5.4</b>	<b>Zusammenfassung</b> .....	250
<b>5.5</b>	<b>Übungen</b> .....	251
5.5.1	Erweiterungen .....	251
5.5.2	ANTLR .....	251

## **6 Semantische Analyse** 253

---

<b>6.1</b>	<b>Einleitung</b> .....	253
<b>6.2</b>	<b>Namensanalyse</b> .....	255
6.2.1	Symboltabellen .....	256
6.2.2	Das Visitor-Pattern .....	265
6.2.3	Type Patterns in Switch-Anweisungen .....	271
6.2.4	Typdeklarationen .....	272
6.2.5	Variablendeklarationen .....	279
6.2.6	Prozedurdeklarationen .....	280
<b>6.3</b>	<b>Typanalyse</b> .....	283
6.3.1	Typanalyse für Ausdrücke .....	285
6.3.2	Typanalyse für Anweisungen .....	291
<b>6.4</b>	<b>Semantische Analyse komplett</b> .....	295
<b>6.5</b>	<b>Vorgehen</b> .....	296
<b>6.6</b>	<b>Zusammenfassung</b> .....	297
<b>6.7</b>	<b>Übungen</b> .....	299
6.7.1	Typen .....	299
6.7.2	Symboltabelle .....	300
6.7.3	Typanalyse .....	300

<b>7</b>	<b>Variablenallokation</b>	301
<b>7.1</b>	<b>Einleitung</b>	301
<b>7.2</b>	<b>Aktivierungsrahmen</b>	303
7.2.1	Aufrufargumente	309
7.2.2	Lokale Variablen	313
7.2.3	Sichern der Register	315
7.2.4	Beispiel für Speicherallokation	316
<b>7.3</b>	<b>Umsetzung im SPL-Compiler</b>	318
<b>7.4</b>	<b>Dynamische Speicherverwaltung</b>	320
7.4.1	Explizite Deallokation	322
7.4.2	Implizite Deallokation	323
<b>7.5</b>	<b>Erweiterungen für andere Sprachen</b>	326
7.5.1	Zugriff auf Variablen eines umgebenden Gültigkeitsbereichs	326
7.5.2	Funktionen	329
7.5.3	Weitere Datentypen	331
<b>7.6</b>	<b>Zusammenfassung</b>	331
<b>7.7</b>	<b>Übungen</b>	332
7.7.1	AllocatorVisitor	332
7.7.2	Aktivierungsrahmen	333
7.7.3	Implementierung	333
<b>8</b>	<b>Codegenerierung</b>	335
<b>8.1</b>	<b>Einleitung</b>	335
<b>8.2</b>	<b>Ziel-Hardware</b>	336
8.2.1	RISC versus CISC	337
<b>8.3</b>	<b>ECO32</b>	337
8.3.1	Unbedingte Sprungbefehle	339
8.3.2	Befehle zum Speicherzugriff	340
8.3.3	Rechenbefehle	341
8.3.4	Sprungmarken (Labels)	342
8.3.5	Bedingte Sprünge	342

<b>8.4</b>	<b>Codemuster</b> .....	344
8.4.1	Ausdrücke .....	345
8.4.2	Zuweisungen .....	351
8.4.3	If-Anweisung .....	353
8.4.4	While-Schleifen .....	355
8.4.5	Zusammengesetzte Anweisung .....	355
8.4.6	Prozeduren .....	356
8.4.7	Prozeduraufrufe .....	357
8.4.8	Beispiel .....	358
8.4.9	Andere Anweisungstypen .....	361
8.4.10	Assembler-Direktiven .....	361
8.4.11	Post-Processing .....	362
<b>8.5</b>	<b>Umsetzung im SPL-Compiler</b> .....	363
<b>8.6</b>	<b>Zusammenfassung</b> .....	364
<b>8.7</b>	<b>Übungen</b> .....	366
8.7.1	Weitere Sprachkonstrukte .....	366
8.7.2	Auswertung von Ausdrücken .....	366
8.7.3	Codegenerierung für Anweisungen .....	367
8.7.4	Implementierung .....	367

## **9 Optimierung** 369

---

<b>9.1</b>	<b>Einleitung</b> .....	369
<b>9.2</b>	<b>Grundlagen für die Optimierung</b> .....	372
<b>9.3</b>	<b>Kontrollflussanalyse</b> .....	374
<b>9.4</b>	<b>Datenflussanalyse</b> .....	383
<b>9.5</b>	<b>Lokale und globale Optimierungen</b> .....	389
<b>9.6</b>	<b>Schleifenoptimierungen</b> .....	392
<b>9.7</b>	<b>Sonstige Optimierungen</b> .....	396
9.7.1	Elimination von Endrekursion .....	397
9.7.2	Inlining .....	398
9.7.3	Leaf Routine Optimization .....	399
9.7.4	Registeroptimierung .....	401
<b>9.8</b>	<b>Static-Single-Assignment</b> .....	405

<b>9.9</b>	<b>Zusammenfassung</b> .....	409
<b>9.10</b>	<b>Übungen</b> .....	411
9.10.1	Kontrollflussanalyse .....	411
9.10.2	Datenflussanalyse .....	412
<b>10</b>	<b>Ausblick</b> .....	413
<hr/>		
<b>10.1</b>	<b>AOT und JIT</b> .....	413
<b>10.2</b>	<b>Forschungsfelder im Compilerbau</b> .....	414
	Literaturverzeichnis .....	417
	Index .....	425