

Inhalt

0. Einführung	15
0.1 Programmiersprachen	15
0.1.1 Begriff	15
0.1.2 Maschinensprachen	16
0.1.3 Maschinenorientierte Programmiersprachen	16
0.1.4 Problemorientierte Programmiersprachen	17
0.1.5 Einfluß des Kontextes bei problemorientierten Programmiersprachen	18
0.2 Kompilation und Interpretation	19
0.2.1 Unterschied von Kompilation und Interpretation	19
0.2.2 Kompilation	19
0.2.3 Aufgaben des Compilers	20
0.2.4 Interpretation	21
0.3 Arbeitsphasen eines Compilers	21
0.3.1 Zerlegung in Phasen	21
0.3.2 Lexikalische Aufbereitung	22
0.3.3 Syntaktische Analyse	23
0.3.4 Weitergehende Zerlegung der syntaktischen Analyse	24
0.3.5 Abschließende Phasen	24
0.4 Weiterführende Konzepte	25
0.4.1 Maschinenunabhängige Compiler-Konzepte	25
0.4.2 Bootstrapping	25
0.4.3 Inkrementelle Compiler	27
0.4.4 Weitere Möglichkeiten	28
0.5 Literaturhinweise	28
0.5.1 Lehrbücher und Übersichtsartikel zur Programmierung	28
0.5.2 Entwicklungstendenzen bei Programmiersprachen	28
0.5.3 Spezielle Programmiersprachen	29
0.5.4 Lehrbücher und Übersichtsartikel über Compiler	30
0.5.5 Maschinenunabhängige Konzepte	30
0.5.6 Weitere Möglichkeiten	30
1. Problemstellung und grundlegende Übersetzungstechniken	32
1.1 Problemstellung	32
1.1.1 Ein Beispiel	32
1.1.2 Zerlegung einer Anweisung	32
1.1.3 Maschinenorientierte Fassung des Beispiels	33
1.1.4 Übersetzung von links nach rechts	34
1.1.5 Compiler-Erstellung	35
1.2 Definition des Sprachumfangs	35
1.2.1 Syntax und Semantik	35
1.2.2 Zur Definition der Syntax	36
1.2.3 Metasprache nach van Wijngaarden	37
1.2.4 Definition der arithmetischen Wertzuweisung	37
1.2.5 Darstellung einzelner Symbole	38

1.3	Sequentielle Formelübersetzung	38
1.3.1	Einführung des Kellerungsprinzips	38
1.3.2	Die Fälle in einer Übergangsmatrix	39
1.3.3	Die Übersetzung mittels Übergangsmatrix	40
1.3.4	Die Ausgänge der Fallunterscheidung	41
1.3.5	Die Übersetzung des Beispiels	43
1.3.6	Zusammenfassung der Keller	43
1.3.7	Programm zum Übersetzungsverfahren	44
1.4	Vorrangverfahren	45
1.4.1	Ausnutzung des Operatorenvorrangs	45
1.4.2	Vorrangalgorithmus zur Übersetzung von Wertzuweisungen	46
1.4.3	Weiterführende Vorrangkonzepte	48
1.4.4	Konstruktion von Vorrangfunktionen	49
1.5	Verfahren des rekursiven Abstiegs	49
1.5.1	Grundgedanke	49
1.5.2	Erkennung von Primärausdrücken	50
1.5.3	Erkennung von Faktoren, Termen und Ausdrücken	51
1.5.4	Erkennung der Wertzuweisung	52
1.5.5	Compiler-Compiler	53
1.6	Die Produktionssprache von Floyd und Evans	54
1.6.1	Grundgedanke	54
1.6.2	Floyd-Evans-Befehle	55
1.6.3	Beispiel	57
1.6.4	Abkürzungen	58
1.6.5	Übersetzung der Wertzuweisungen mit einem FE-Programm	59
2.	Grundlegende Techniken auf der maschinenorientierten Seite	61
2.1	Kellerung der Zwischenergebnisse	61
2.1.1	Zerlegung einer Anweisung	61
2.1.2	Zwischenergebniskeller	61
2.1.3	Maschinenbefehle	62
2.1.4	Beispiel (Ein-Adreß-Version)	63
2.1.5	Beispiel (Zwei-Adreß-Version)	64
2.2	Die Code-Generierung	65
2.2.1	Aufgaben der Routine GENERATE	65
2.2.2	Globale Information für GENERATE	65
2.2.3	Programm für GENERATE	67
2.3	Realisierung des Zwischenergebniskellers	68
2.3.1	Grundgedanke	68
2.3.2	Vor- und Nachteile beider Verfahren	69
2.3.3	Indirekte und modifizierte Adressen	70
2.3.4	Indexregister	71
2.3.5	Lösung für beide Verfahren	72
2.4	Operandenarten und Artanpassung	73
2.4.1	Verfügbare Operatoren	73
2.4.2	Artanpassungen in höheren Programmiersprachen	74
2.4.3	Generierung der Artanpassungen	74
2.4.4	Berücksichtigung der Artanpassungen in der Symbolliste	75
2.4.5	Dereferenzierung	75
2.5	Informationsvektoren	76
2.5.0	Indizierte Variable	76

2.5.1	Speicherung von Feldern	77
2.5.2	Speicherabbildungsfunktion	78
2.5.3	Versorgung der Speicherabbildungsfunktion mit den Indexwerten	79
2.5.4	Beispiel	80
2.5.5	Die Übersetzung der indizierten Variablen	81
2.5.6	Die Übersetzung der Deklarationen	82
2.6	Dynamische Speicherverwaltung	82
2.6.1	Charakterisierung der verschiedenen Speicherbereiche	82
2.6.2	Beispiel	83
2.6.3	Die Verwaltung des freien Speichers	84
2.6.4	Beispiel	85
2.6.5	Die Übersetzung der Blockstruktur	86
3.	Lexikalische Analyse	87
3.1	Die Erkennung von Identifikatoren und Konstanten	87
3.1.0	Aufgabenstellung	87
3.1.1	Verwaltung des Eingabetextes	87
3.1.2	Identifikatoren	88
3.1.3	Ganzzahlige Konstante	89
3.1.4	Übergang zu reellwertigen Konstanten	89
3.2	Ein automatentheoretisches Modell	90
3.2.1	Grundgedanke	90
3.2.2	Automat zur Erkennung von Identifikatoren und Konstanten	90
3.2.3	Automatentafel	92
3.2.4	Bearbeitung der Identifikatoren und ganzzahligen Konstanten	94
3.2.5	Bearbeitung der reellwertigen Konstanten	95
3.3	Begrenzungssymbole	96
3.3.1	Problemstellung	96
3.3.2	Bearbeitung der Begrenzungssymbole (Lochkartendarstellung)	97
3.3.3	Zeichenketten und Kommentare	98
3.4	Aufbau der Symbolliste	99
3.4.1	Der Gültigkeitsbereich einzelner Symbole	99
3.4.2	Beispiel	100
3.4.3	Symbolliste bei Blockstruktur	103
3.4.4	Beispiel	103
3.4.5	Suchen in einer blockstrukturierten Symbolliste	104
4.	Ein Beispiel-Compiler	107
4.0	Überblick	107
4.0.1	Sprachumfang	107
4.0.2	Einteilung der Phasen	109
4.0.3	Implementierung	110
4.1	Phase I	111
4.1.0	Rechnerabhängige Prozeduren	111
4.1.1	Prozedur NEXT SYMBOL	113
4.1.2	Liste der Begrenzer	115
4.1.3	Hauptprogramm	116
4.1.4	Identifikatoren	118
4.1.5	Ganzzahlige Konstanten	118
4.1.6	Reellwertige Objekte	119

4.2 Phase II	120
4.2.0 Übersicht	120
4.2.1 Kellerprozeduren	121
4.2.2 Symbolliste	123
4.2.3 Programmübersicht und Blockstruktur	125
4.2.4 Einfache Deklarationen	127
4.2.5 Feldvereinbarungen	128
4.2.6 Marken	130
4.2.7 Anweisungen	130
4.3 Phase III	131
4.3.0 Übersicht	131
4.3.1 Kellerprozeduren	132
4.3.2 Identifizierung der Identifikatoren und Adressenzuweisung	133
4.3.3 Programmübersicht	135
4.3.4 Blockstruktur	137
4.3.5 Marken und Sprunganweisungen	139
4.3.6 Deklarationen	140
4.3.7 Feldvereinbarungen und indizierte Variable	142
4.3.8 Indexgrenzen und Indizes	144
4.3.9 Ausdrücke und Anweisungen	146
4.4 Phase IV	148
4.4.0 Übersicht	148
4.4.1 Kommandobearbeitung	149
4.4.2 Identifizierung der Operatoren	152
4.4.3 Programmübersicht	154
4.4.4 Wertzuweisungen	155
4.4.5 Aktionen	157
4.4.6 Indexgrenzen, Indizes und Klammern	160
4.4.7 Adressierung	161
5. Die Übersetzung verschiedener Datenarten	162
5.1 Aspekte der Optimierung bei arithmetischen Ausdrücken	162
5.1.1 Gemeinsame Teilausdrücke	162
5.1.2 Heuristische Optimierungen	163
5.1.3 Registereinsparung	164
5.1.4 Operatoren für zusammengesetzte Operanden	165
5.2 Behandlung von Präfix-Arten	166
5.2.1 Arten verschiedener Genauigkeit	166
5.2.2 Arten verschiedener Referenzstufen	167
5.2.3 Aufbau der Symbolliste	167
5.2.4 Übersetzung der Präfixarten	169
5.3 Behandlung weiterer zusammengesetzter Arten	170
5.3.1 Feldarten	170
5.3.2 Verbundarten	171
5.3.3 Erweiterung der Symbolliste	172
5.3.4 Vereinigung von Arten	173
5.4 Blockunabhängige Speicherverwaltung	174
5.4.0 Übersicht	174
5.4.1 COMMON und EXTERNAL	175
5.4.2 Programmierte Speicherverwaltung	175
5.4.3 Halde	176

6. Übersetzung der Kontrollstrukturen	177
6.1 Vergleiche und boolesche Formeln	177
6.1.1 Sprachumfang	177
6.1.2 Kaskadenübersetzung	178
6.1.3 Beispiel	179
6.1.4 Zielcode	180
6.1.5 Boolesche Wertzuweisung	180
6.2 Bedingte Anweisungen	181
6.2.1 Bedingungen auf Maschinenebene	181
6.2.2 Zielcode	182
6.2.3 Beispiel	183
6.2.4 Übersetzung des linken Operanden	186
6.2.5 Übersetzung des rechten Operanden	187
6.3 Lauschleifen (Zyklen)	188
6.3.1 Darstellung der Schleifen mit Abfragen	189
6.3.2 Zielcode	190
6.3.3 Lauf Listen	191
6.3.4 Rekursive Adressenberechnung	191
6.4 Fallunterscheidungen	193
6.4.1 Fallunterscheidungen	193
6.4.2 Verteiler	194
6.4.3 Markenvariable	195
6.5 Vorwärtssprünge	196
6.5.1 Vormerkliste	196
6.5.2 Verkettete Vormerkungen	196
7. Unterprogramm	199
7.1 Unterprogrammaufruf	199
7.1.1 Aktueller Parameter	199
7.1.2 Parameterruf über die Adresse	201
7.1.3 Beispiel	202
7.1.4 Parameterruf über den Wert	204
7.1.5 Parameterruf über den Namen	205
7.1.6 Parameterbehandlung bei ALGOL 60	206
7.2 Unterprogrammdeklarationen	208
7.2.1 Sprachumfang	208
7.2.2 Rückkehradresse	208
7.2.3 Spezielle Probleme	209
7.2.4 Speicherverwaltung	210
7.2.5 Speicherverwaltung bei rekursiven Prozeduren	211
8. Formale Sprachen	213
8.1 Grundbegriffe	213
8.1.0 Backus-Naur-Form	213
8.1.1 Beispiel	213
8.1.2 Ableitbarkeit	214
8.1.3 Abgekürzte Notation der Beispiele	215
8.1.4 Abstrakte Beispiele	215
8.1.5 Syntaxbaum	216
8.1.6 Mehrdeutige Produktionensysteme	218

8.2 Chomsky-Sprachen	219
8.2.1 Konkatenation von Wörtern	219
8.2.2 Operationen mit Wortmengen	219
8.2.3 Iteration	220
8.2.4 Semi-Thue-Systeme	221
8.2.5 Chomsky-Grammatik	221
8.3 Chomsky-Hierarchie	222
8.3.1 Notwendigkeit von Einschränkungen	222
8.3.2 Kontextsensitive und monotone Grammatiken	223
8.3.3 Kontextfreie Grammatiken	223
8.3.4 Lineare Grammatiken	224
8.4 Kontextfreie Grammatiken	225
8.4.1 Kontextunabhängigkeit der Ableitungen	225
8.4.2 Kontextfreie Grammatiken mit Löschung	225
8.4.3 Reduzierte Grammatiken	226
8.4.4 Mengentheoretische Operationen	228
8.5 Zwei-Stufen-Grammatiken	229
8.5.1 Attribut-Grammatiken	229
8.5.2 Beispiel	230
8.5.3 Darstellung kontextsensitiver Sprachen	231
8.5.4 Grammatiken mit Kontrollmenge	232
8.5.5 Beispiel	233
9. Modelle zur Rekognition	234
9.1 Endliche Automaten	234
9.1.1 Definition des endlichen Automaten	234
9.1.2 Beispiel	235
9.1.3 Nichtdeterministische Automaten	236
9.2 Kellerautomaten	236
9.2.1 Definition des Kellerautomaten	236
9.2.2 Angenommene Wortmenge	238
9.2.3 Beispiele: Annahme durch leeren Keller	238
9.2.4 Beispiele: Annahme durch markierten Zustand	240
9.2.5 Weitere Resultate	241
9.3 Rekognitionsstrategien	241
9.3.1 Rekognitionsverfahren	241
9.3.2 Kanonische und antikanonische Ableitungen	242
9.3.3 Bottom-up- und Top-down-Strategie	243
9.3.4 Beschreibung der Top-down-Strategie	244
9.3.5 Beschreibung der Bottom-up-Strategie	246
9.3.6 Sackgassen	248
9.3.7 Left-corner-Strategie	249
10. Rekognitionsverfahren	252
10.1 Vorrangverfahren	252
10.1.1 Operatorenvorrang	252
10.1.2 Allgemeine Vorrangrelationen	253
10.1.3 Beschreibung des Vorrangverfahrens	254
10.1.4 Beispiel	256
10.1.5 Weitere Resultate	259

10.2 Verfahren mit beschränktem Kontext	259
10.2.1 (m,n)-beschränkter Kontext	259
10.2.2 Beispiel	260
10.2.3 Prüfung der Beschränktheitsbedingung	261
10.2.4 (1,1)-LR-beschränkter Kontext	262
10.2.5 Beispiel	263
10.2.6 LR(k)-Grammatiken	264
10.3 Verfahren des rekursiven Abstiegs	264
10.3.1 PM-Befehle	265
10.3.2 Beispiel	266
10.3.3 Konstruktion von PM-Programmen	267
10.3.4 Back-up-freie PM-Programme	268
10.3.5 Beseitigung linksrekursiver Symbole	270
Literatur	272
Namen- und Sachverzeichnis	281