Stefan Jähnichen
Christoph Oeters
Bruce Willis

Übersetzerbau

Skriptum für Informatiker
im Hauptstudium

Springer Fachmedien Wiesbaden GmbH

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Jähnichen, Stefan

Übersetzerbau: Skriptum für Informatiker im
Hauptstudium/Stefan Jähnichen; Christoph Oeters;
Bruce Willis. – 1 Aufl. – Braunschweig: Vieweg, 1978
ISBN 978-3-528-03331-6

NE: Oeters, Christoph.; Willis, Bruce:

Verlagsredaktion: *Alfred Schubert*

1978

Alle Rechte vorbehalten

© Springer Fachmedien Wiesbaden 1978

Ursprünglich erschienen bei Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig 1978

Die Vervielfältigung und Übertragung einzelner Textabschnitte, Zeichnungen oder Bilder, auch für Zwecke der Unterrichtsgestaltung, gestattet das Urheberrecht nur, wenn sie mit dem Verlag vorher vereinbart wurden. Im Einzelfall muß über die Zahlung einer Gebühr für die Nutzung fremden geistigen Eigentums entschieden werden. Das gilt für die Vervielfältigung durch alle Verfahren einschließlich Speicherung und jede Übertragung auf Papier, Transparente, Filme, Bänder, Platten und andere Medien.

ISBN 978-3-528-03331-6

ISBN 978-3-663-13897-6 (eBook)

DOI 10.1007/978-3-663-13897-6

Vorwort

Dieses Buch entstand aus Skripten, die zur Lehrveranstaltung ÜBERSETZERBAU im Fachbereich Informatik der Technischen Universität Berlin angefertigt wurden. Die Lehrveranstaltung wird seit 1972 durchgeführt und wurde als ein Beitrag zur Neuordnung des Informatik Studiums (NORIS) konzipiert. Sie basiert auf dem im Rahmen dieses Konzeptes entwickelten Grundstudium Informatik und ist für Studenten im Hauptstudium gedacht, die Interesse an Programmiersprachen und ihren Übersetzern haben.

Der Konzeption der Lehrveranstaltung ÜBERSETZERBAU liegt das Ziel zugrunde, die wesentlichen Methoden des Übersetzerbaus projektbezogen zu vermitteln. Deshalb wird der größte Teil der Lehrveranstaltung dazu benutzt, einen Übersetzer für eine kleine Sprache zu implementieren. Auf diese Weise werden nicht nur die Konzepte am wirkungsvollsten vermittelt, sondern die Teilnehmer sammeln auch praktische Erfahrungen an einem größeren Softwareprodukt, die ihnen für die spätere Berufspraxis sehr nützlich sind.

Wir danken allen, die im Rahmen von NORIS am Aufbau des Studienganges Informatik mitgearbeitet haben. Insbesondere danken wir auch denen, die in den vergangenen Jahren aktiv an der Gestaltung der Lehrveranstaltung mitgewirkt haben, aber auch den Studenten, die teilgenommen haben und durch ihre Kritik die Weiterentwicklung der Lehrveranstaltung unterstützt haben.

Vor allem aber möchten wir Frau Gabriele Ambach danken, die das Manuskript sehr sorgfältig angefertigt hat und trotz zahlreicher Änderungen alle Arbeiten mit viel Geduld erledigt hat.

In memorium NORIS
die Autoren, im Januar 1978

I N H A L T S V E R Z E I C H N I S

	Seite
1. <u>EINLEITUNG</u>	1
1.1. Der Kurs Übersetzerbau	1
1.2. Allgemeine Terminologie	2
2. <u>ANALYSE</u>	6
2.1. Sprachen und Grammatiken	6
2.1.1. Einführung und Begriffsbildung	6
2.1.2. Grammatiktypen	10
2.1.3. Analyseverfahren	13
2.2. Lexikalische Analyse	16
2.2.1. Die Aufgabe	16
2.2.2. Erkennung einer lexikalischen Einheit	17
2.2.3. Bestimmung der Grenzen lexikalischer Einheiten	19
2.2.4. Fehlererkennung	22
2.2.5. Behandlung gemeinsamer Anfänge	23
2.2.6. Aufbau lexikalischer Einträge	25
2.2.7. Ausnahmefälle	31
2.2.8. Behandlung lexikalischer Fehler	33
2.3. Syntaktische Analyse	36
2.3.1. Das Problem	36
2.3.2. Absteigende und aufsteigende Analyse	37
2.3.3. Absteigende Analyse (top-down)	40
2.3.3.1. Rekursiver Abstieg	40
2.3.3.2. Typische Probleme	43
2.3.4. Aufsteigende Analyse (bottom-up)	48
2.3.4.1. Vorranganalyse (simple precedence)	48
2.3.4.2. LR(k)-Analyse	56
2.3.5. Vergleich der Analyseverfahren	64
2.3.6. Behandlung von syntaktischen Fehlern	66

	Seite
3, <u>ZWISCHENSPRACHEN</u>	75
3.1. Baumsprache	76
3.2. Quadrupelsprache	81
3.3. Postfixsprache	83
4, <u>SYNTHESE</u>	85
4.1. Übersetzung bei rekursivem Abstieg	86
4.2. Übersetzung mit einfachem Vorrang	90
4.3. Meldungen	93
5, <u>IDENTIFIZIERUNG</u>	99
5.1. Prinzipien	99
5.2. Definitionen und Anwendungen	99
5.3. Block-orientierte Identifizierung	101
5.4. Inhalt der Definitionstabelle	106
5.5. Andere Identifizierungsbegriffe	107
6, <u>SPEICHERVERWALTUNG</u>	109
6.1. Statische Adressierung	109
6.1.1. Speicherplatzzuordnung für Objekte eines Blockes	109
6.1.2. Geschachtelte Blöcke	111
6.1.3. Nichtrekursive Prozeduraufrufe	113
6.1.4. Prozedurresultate	114
6.1.5. Prozedurparameter	115
6.1.6. Statische Bestandteile des Datenraums einer Prozedur (DSA)	117
6.2. Dynamische Adressierung	117
6.2.1. Dynamische Verwaltung von Datenräumen	117
6.2.2. Rekursive Prozeduraufrufe	120
6.2.3. Geschachtelte Prozeduren	121
6.2.4. Werte von Prozeduren und Marken	127
6.2.5. Felder (Reihen)	130
6.2.5.1. Deklaration von Feldern	130
6.2.5.2. Zugriff auf Feldelemente	137
6.2.5.3. Spezielle Probleme	140
6.2.6. Beispiel zum Aufbau eines Datenraums	142

	Seite
6.3. Parameterübergabe	143
6.3.1. Call by value	144
6.3.2. Call by reference	144
6.3.3. Call by name	145
6.3.4. Call by value-result	148
6.4. Speicherverwaltung für Datenobjekte ohne kellerbedingte Lebensdauer	150
6.4.1. Sprachelemente für Objekte mit unbegrenzter Lebensdauer	152
6.4.2. Methoden zur Speicherrückgewinnung	154
6.5. Parallele Prozesse	162
6.5.1. Kellerverwaltung	163
6.5.2. Kachelketten	167
6.5.2.1. Benutzung von Kachelketten	167
6.5.2.2. Platzverlust	168
6.5.2.3. Speicherbereinigung	169
6.5.2.4. Große Datenräume	169
6.5.2.5. Sprünge	169
6.5.3. Auslagerung von Datenräumen	171
7. <u>OPTIMIERUNG</u>	177
7.1. Zweckmäßigkeit von Optimierungen	177
7.2. Optimierungstechniken	179
7.2.1. Maschinenunabhängige Optimierungen	180
7.2.1.1. Folding und Propagation	180
7.2.1.2. Strength reduction	180
7.2.1.3. Code motion	181
7.2.1.4. Common subexpression elimination	182
7.2.1.5. Redundant code elimination	183
7.2.2. Maschinenabhängige Optimierungen	183
7.2.2.1. Registerzuordnung	183
7.2.2.2. Benutzung spezieller Maschineninstruktionen	184
7.2.2.3. Parallelverarbeitung	184
7.3. Durchführung der Optimierung in der Synthese	185
7.3.1. Lokale Optimierungen	186

	Seite
7.3.2. Globale Optimierungen	195
7.3.2.1. Schleifenoptimierung	195
7.3.2.2. Optimierung der Prozedurbehandlung	196
7.4. Peep-hole-Optimierung	198
8. <u>CODEERZEUGUNG</u>	199
8.1. Arten des Objekt-Codes	199
8.2. Verfahren bei der Codeerzeugung	201
8.2.1. Definition der Zielmaschine	202
8.2.2. Adressierung und Zugriff auf Objekte	203
8.2.3. Codegenerierung für arithmetische Ausdrücke	205
8.2.3.1. Tabellengesteuerte Codeerzeugung	206
8.2.3.2. Codeerzeugung mit Hilfe von Objekt-deskriptoren	210
ANHANG A : <u>DIE SPRACHEN MINI UND MAXI</u>	218
A.1. Vorbemerkung	218
A.2. Beschreibung der Sprachen MINI und MAXI	218
A.2.1. Program	219
A.2.2. Clauses	220
A.2.3. Declarations und Declarer	225
A.2.4. Units	230
A.2.5. Definition und Application	238
A.2.6. Denoter	239
A.2.7. Token Symbole	240
A.2.8. Comment und Pragmat	242
A.2.9. Anpassungen	243
ANHANG B : <u>DER MINI-COMPILER</u>	244
B.1. Pass-Struktur	244
B.2. Schichtenstruktur	245
B.3. Der erste Pass: Analyse	246
B.3.1. Lexikalische Analyse	246
B.3.2. Syntaxanalyse	250
B.3.3. Die Schnittstelle zwischen dem ersten und dem zweiten Pass	254

	Seite
B.4. Der zweite Pass: semantische Analyse und Codeerzeugung	254
B.4.1. Semantische Analyse	255
B.4.2. Codeerzeugung	256
ANHANG C : <u>ÜBUNGEN</u>	266
ANHANG D : <u>EINFÜHRUNG IN DIE IMPLEMENTIERUNGSSPRACHE CDL2</u>	275
D.1. CDL2 als Systemimplementierungssprache	275
D.2. Programmierung im Großen	276
D.2.1. Modularisierungskonzepte	276
D.2.2. Schnittstellen	277
D.3. Programmierung im Kleinen	278
D.3.1. Algorithmen	278
D.3.1.1. Der Typ eines Algorithmus	279
D.3.1.2. Die Affixe eines Algorithmus	280
D.3.1.3. Regeln	282
D.3.1.4. Makros	286
D.3.2. Daten	289
D.4. Steuerung von CDL2 Programmen	290
LITERATURHINWEISE	293
SACHWORTVERZEICHNIS	298