

Inhalt

Vorwort	9
A. Prolog- und Programmierumgebung	
<hr/>	
1. Aufruf und Verlassen des Systems	13
1.1 Aufruf	13
1.2 Editieren von Prolog-Dateien	15
1.3 Sichern und Wiederaufruf einer Prolog-Sitzung	15
1.4 Verlassen des Prolog-Systems	16
2. Benutzerdialog	17
2.1 Prolog-Prompt	17
2.2 Prolog-Anfragen	18
3. Steuerfluß	23
3.1 Steuerung durch den Prolog-Interpreter	23
3.2 Rekursion	23
3.3 Backtracking	26
3.4 Der Cut	27
3.5 Iteration	31
4. Ein- und Ausgabe-Prädikate	33
4.1 Umlenken des Ein- und Ausgabestroms	33
4.2 Lesen und Schreiben von Termen	35
4.3 Lesen und Schreiben von Zeichen	41
5. Konsultierung und Modularisierung	45
5.1 Konsultieren von Prolog-Prädikaten	45
5.2 Mischen von interpretiertem und kompiliertem Prolog-Code	47
5.3 Modularisierung	49
6. Grundlegende Datentypen	55
6.1 Terme	55
6.2 Listen	60
6.3 Operatoren	62

7.	Klassifizierende Prädikate	67
7.1	Abprüfung von Datentypen	67
7.2	Typkonvertierung	68
7.3	Datentypprüfung einzelner Zeichen	69
8.	Numerische Prädikate	71
8.1	Vergleich von Termen	72
8.2	Vergleich arithmetischer Ausdrücke	73
8.3	Negierte Vergleiche	73
9.	Listen- und Zeichenbearbeitung	75
9.1	Zugriff auf die Elemente einer Liste	75
9.2	Zeichenverarbeitung	79
10.	assert und retract	85
10.1	Dynamisches Verändern der Prolog-Datenbasis	85
10.2	Zwei Definitionen von consult/1 - ein Beispiel zum Prolog-Programmierstil	87
11.	Testmethoden und -hilfen	91
11.1	Grundlegende Testmethoden	91
11.2	Verfolgen eines Prolog-Prädikats mit Trace	95
11.3	Interaktives Testen	99
12.	Systemanschluß	115
12.1	Systemaufrufe	115
12.2	Komfortables Editieren mit edit	117
12.3	Kommunikation mit Systemprozessen	120

B. Programmiertechnik

1.	Datenstrukturen	125
1.1	Bäume als Graphen	125
1.2	Bäume als rekursive Relationen	127
1.3	Listen, Kellerspeicher und Warteschlangen	133
1.4	Offene Listen	136
1.5	Tabellen	139
1.6	Entscheidungstabellen und geschützte Klauseln	141

2.	Kontrollstrukturen	145
2.1	Zähler	145
2.2	Akkumulatoren	147
2.3	"Cut" und "not"	147
2.4	Breitensuche	151
3.	Endliche Automaten	155
3.1	Zustände und Zustandsübergänge	155
3.2	Beispiel: Verkehrsampel	158
3.3	Interpreter für endliche Automaten	160
3.4	Hilfen für Prototyping und Test	161
4.	Modularisierung von Wissensbasen	167
4.1	Datenabstraktion	167
4.2	Modularisierungsmittel des Compilers	169
4.3	Segmentierte Wissensbasen	170
4.4	Externe Texte	171
4.5	Ersetzungstexte	173
5.	Test- und Wartungshilfen	177
5.1	Ausnahmebehandlung	177
5.2	Termtypen und klassifizierende Prädikate	179
5.3	Parameter-Typisierung	182
5.4	Testrahmen	186
5.5	Testfälle und Testtreiber	188
5.6	Protokollschreibung	191
6.	Grammatiken	193
6.1	Die DCG-Darstellung von Syntax und Semantik	193
6.2	Verarbeitung von Zeichenketten	195
6.3	Analyse natürlicher Sprachen	197
6.4	Kontext-Abhängigkeiten	201
6.5	Semantische Terme und Einschübe	205
7.	Generatoren	213
7.1	Syntaxorientierte Texterzeugung	213
7.2	Generierung von Kommandofolgen	215
7.3	Generierung von Dokumenten	221

C. Anhang

1.	Literaturempfehlungen	231
2.	Literaturverzeichnis	233
3.	Register	237