

# Inhalt

Einleitung	1
<b>I Prolog als Sprache und Werkzeug</b>	<b>5</b>
1 Die Sprache Prolog	7
1.1 Grundkonzepte	7
1.2 Logik und Prolog	12
1.3 Sprachkonstrukte	17
1.4 Programmausführung	23
2 Eine Implementierung	31
2.1 Auswahlkriterien	31
2.2 Komponenten	34
2.3 Sprachelemente	35
2.4 Entwicklungsumgebung	35
2.5 Beispielsitzung	36
2.6 Eingebaute Prädikate	38
2.7 Spracherweiterungen	42
<b>II Von Prolog zum wissensbasierten System</b>	<b>45</b>
3 Problemlösung durch Logik	47
3.1 Generierung und Test	47
3.2 Planung	50
3.3 Entscheidung	52
3.4 Musteranpassung	55
3.5 Imperative Programmierung	62
3.6 Dynamische Relationen	67
3.7 Strategische Suche	72
4 Wissensbasierte Systeme und Expertensysteme	77
4.1 Wissensbasierung und Fachkenntnis	77
4.2 Problemlösungskomponente	79
4.3 Wissensbank	80
4.4 Wissenserwerb	81
4.5 Wissenserwerbskomponente	81

4.6	Erklärungskomponente . . . . .	82
<b>5</b>	<b>Ein Diagnosesystem</b>	<b>85</b>
5.1	Aufgabe und Voraussetzungen . . . . .	85
5.2	Inferenzkomponente . . . . .	86
5.3	Wissensdarstellung . . . . .	91
5.4	Wissensaufteilung . . . . .	92
5.5	Erklärungskomponente . . . . .	94
5.6	Wissenserwerbskomponente . . . . .	97
5.7	Dokumentationskomponente . . . . .	100
<b>6</b>	<b>Ein Beratungssystem</b>	<b>107</b>
6.1	Aufgabenstellung . . . . .	107
6.2	Konzeption . . . . .	108
6.3	Erinnernde Inferenzkomponente . . . . .	109
6.4	Attribut-Wert Darstellung . . . . .	110
6.5	Menüs . . . . .	111
6.6	Formulare . . . . .	113
6.7	Wissensdarstellung . . . . .	118
<b>7</b>	<b>Softwaretechnik</b>	<b>121</b>
7.1	Logisches Programmieren als Paradigma . . . . .	121
7.2	Werkzeuge, Bausteine und Schalen . . . . .	123
7.3	Rasche Prototypentwicklung . . . . .	125
7.4	Softwarelebenszyklus . . . . .	126
7.5	Programmieren im Großen . . . . .	130
7.6	Programmieren im Kleinen . . . . .	131
7.7	Prolog für konventionelle DV-Aufgaben . . . . .	138
7.8	Ein Prolog-Derivat für die Systemimplementierung . . . . .	139
<b>III</b>	<b>Ausgewählte Techniken und Methoden</b>	<b>143</b>
<b>8</b>	<b>Meta-Interpreter</b>	<b>145</b>
8.1	Einfache Interpretation . . . . .	145
8.2	Erklärungsfähigkeit . . . . .	147
8.3	Wissensbasis . . . . .	148
8.4	Unsicherheit . . . . .	153
8.5	Hierarchie und Vererbung . . . . .	157
8.6	Breitensuche . . . . .	164
8.7	Einfacher LISP-Interpreter . . . . .	166
8.8	Partielle Evaluation . . . . .	168
<b>9</b>	<b>Datenbanken</b>	<b>173</b>
9.1	Prolog als Datenbank . . . . .	173
9.2	Wissensaufnahme und Wissensassimilation . . . . .	185
9.3	Negation . . . . .	186
9.4	Konsistenz . . . . .	187

9.5	Ableitbarkeit . . . . .	192
9.6	Redundanz . . . . .	193
9.7	Sichten . . . . .	193
9.8	Nullwerte . . . . .	194
9.9	Effizienz . . . . .	196
<b>10</b>	<b>Produktionssysteme</b>	<b>203</b>
10.1	Aufbau und Arbeitsweise . . . . .	203
10.2	Kontrollstrategien . . . . .	205
10.3	Rückwärts Schließen . . . . .	207
10.4	Vorwärts Schließen . . . . .	209
10.5	Ein Beispiel . . . . .	212
10.6	Negierte Ausdrücke . . . . .	214
10.7	Vermeidung von Endlosschleifen . . . . .	216
10.8	Auswahlstrategien und Filter . . . . .	218
<b>11</b>	<b>Parallelität</b>	<b>227</b>
11.1	Prolog-Programme und Parallelität . . . . .	227
11.2	Ausdrucksmittel für Parallelität . . . . .	230
11.3	Interpreter für Konnektoren und Meta-Prädikate . . . . .	233
11.4	Interpreter für Variablen-Annotation . . . . .	240
11.5	Interpreter für flache geschützte Hornklauseln . . . . .	244
<b>12</b>	<b>Simulation</b>	<b>249</b>
12.1	Diskrete Simulation . . . . .	249
12.2	Simulation und Backtracking . . . . .	249
12.3	Simulation in reinem Prolog . . . . .	251
12.4	Ein Simulationssystem in Prolog . . . . .	254
	<b>Literatur</b>	<b>263</b>