

Armin B. Cremers
Ulrike Griefahn
Ralf Hinze

Deduktive Datenbanken

Artificial Intelligence

Künstliche Intelligenz

herausgegeben von Wolfgang Bibel und Walther von Hahn

Künstliche Intelligenz steht hier für das Bemühen um ein Verständnis und um die technische Realisierung intelligenten Verhaltens.

Die Bücher dieser Reihe sollen Wissen aus den Gebieten der Wissensverarbeitung, Wissensrepräsentation, Expertensysteme, Wissenskommunikation (Sprache, Bild, Klang, etc.), Spezialmaschinen und -sprachen sowie Modelle biologischer Systeme und kognitive Modellierung vermitteln.

Auswahl der bisher erschienenen Titel:

Automated Theorem Proving

von Wolfgang Bibel

Prolog

von Ralf Cordes, Rudolf Kruse, Horst Langendörfer, Heinrich Rust

Wissensbasierte Systeme

von Doris Altenkrüger und Wilfried Büttner

Logische Grundlagen der Künstlichen Intelligenz

von Michael R. Genesereth und Nils J. Nilsson

Wissensbasierte Echtzeitplanung

von Jürgen Dorn

Modulare Regelprogrammierung

von Siegfried Bocionek

Logische und Funktionale Programmierung

von Ulrich Furbach

Parallelism in Logic

von Franz Kurfeß

Schließen bei unsicherem Wissen in der Künstlichen Intelligenz

von der Gruppe Léa Sombé

Wissensrepräsentation und Inferenz

von Wolfgang Bibel, zusammen mit St. Hölldobler und T. Schaub

Deduktive Datenbanken

von Armin B. Cremers, Ulrike Griefahn und Ralf Hinze

Armin B. Cremers
Ulrike Griefahn
Ralf Hinze

Deduktive Datenbanken

Eine Einführung aus der Sicht
der logischen Programmierung



Das in diesem Buch enthaltene Programm-Material ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Die Autoren und der Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials oder Teilen davon entsteht.

Alle Rechte vorbehalten

© Springer Fachmedien Wiesbaden 1994

Ursprünglich erschienen bei Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig 1994



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Gedruckt auf säurefreiem Papier

ISBN 978-3-528-04700-9 ISBN 978-3-663-09572-9 (eBook)
DOI 10.1007/978-3-663-09572-9

Vorwort

Der Begriff „deduktive Datenbank“ wurde vor etwa zwanzig Jahren geprägt, um eine neue Forschungsrichtung zu bezeichnen, die sich im Grenzbereich zwischen Künstlicher Intelligenz und Datenbankforschung zu entwickeln begann. Das wesentliche Anliegen der Wissenschaftler, die sich seither diesem Thema gewidmet haben, ist die Realisierung effizienter, automatischer Inferenzmethoden, die es erlauben, große Faktenmengen durch einzelne definierende Ausdrücke statt durch explizite Aufzählung darzustellen.

Erste Experimente in den späten sechziger und frühen siebziger Jahren mit Kopplungen von automatischen Beweissystemen und Datenbanksystemen können als Ausgangspunkt der Forschungen angesehen werden.¹ Nach einer intensiven Auseinandersetzung mit den theoretischen Grundlagen deduktiver Datenbanken richtete sich das Forschungsinteresse Mitte der achtziger Jahre zunehmend auf praktische Implementierungstechniken, vorwiegend im Kontext des relationalen Datenmodells. In jüngster Zeit werden regelbasierte Wissensdarstellung und automatische Inferenz von Daten auch im Rahmen anderer Datenmodelle, insbesondere objektorientierter Systeme, vorgeschlagen und untersucht.

In der Künstlichen Intelligenz entwickelte sich ungefähr zur gleichen Zeit die logische Programmierung. Sie basiert ebenfalls auf Methoden zum automatischen Beweisen, die zwischen 1960 und 1970 vorgeschlagen wurden. Die Verwendung von Logik als Programmiersprache zielt auf eine stärker deskriptive und weniger präskriptive Formulierung von Algorithmen. Populär wurde diese Forschungsrichtung in erster Linie durch die starke Verbreitung der Programmiersprache Prolog.

Die theoretischen Grundlagen beider Forschungsgebiete wurden nach 1970 praktisch gleichzeitig entwickelt. Seither hat sich die Entwicklung der beiden Richtungen durchaus unabhängig voneinander vollzogen. Dementsprechend kann man heute, nach ungefähr zwanzig Jahren intensiver Forschung, ein weites Spektrum von Ansätzen und Methoden antreffen, die zum Teil nur noch schwer zu überschauen sind. Innerhalb dieses Spektrums läßt sich das vorliegende Buch in sehr enger Nachbarschaft zur logischen Programmierung einordnen. Wesentliches Leitmotiv ist die Verwendung von Logik als Datenbanksprache. Logische Formeln erlauben in sehr einheitlicher Weise die Beschreibung von Daten, Anfragen, Sichten, Integritätsbedingungen und Programmen.

¹Mit der bisherigen Geschichte dieses Forschungsgebietes beschäftigt sich der Artikel „Perspectives in deductive databases“ von Jack Minker [Minker 88b].

Das Buch ist wie folgt gegliedert. Zunächst werden die Grundlagen der logischen Programmierung erarbeitet und ein spezielles Grundmodell deduktiver Datenbanken definiert. Die Syntax der Datenbanksprache ist dabei im Gegensatz zu vielen in der Literatur aufgeführten Ansätzen sehr weit gefaßt: Objekte eines zu modellierenden Weltausschnitts können durch komplex strukturierte Terme beschrieben werden, Beziehungen durch beliebige prädikatenlogische Formeln. Die schrittweise Erweiterung dieses Grundmodells um Datenbankkonzepte wie Anfrageauswertung, Integritätsprüfung und Änderungsbearbeitung bildet einen Schwerpunkt des Buches. Breiten Raum nimmt daneben die Diskussion von Typ- und Modusssystemen ein, die in der logischen Programmierung zunehmend diskutiert werden. Das Buch schließt mit einem praktischen Teil, in dem die prototypische Realisierung des vorgestellten deduktiven Datenbanksystems in Prolog erörtert wird.

Die Idee zu diesem Buch entstand auf der Grundlage dreier Diplomarbeiten, die in der zweiten Hälfte der achtziger Jahre an der Universität Dortmund durchgeführt wurden. Diese Arbeiten umfassen eine allgemeine Abhandlung theoretischer und praktischer Aspekte von deduktiven Datenbanken, aus der auch eine erste Implementierung hervorging [Lüttringhaus 87], die Überprüfung von Integritätsbedingungen in deduktiven Datenbanken [Griefahn 89] und Typsysteme in der funktionalen Programmierung [Hinze 89].

Durch die Beteiligung an europäischen Kooperationsprojekten, wie dem ESPRIT-Projekt 530 Epsilon (Advanced Knowledge Base Management System Based on the Integration of Logic Programming and Databases) [Rohen 88] und dem EUREKA-Projekt 56 PROTOS (Prolog Tools for Building Expert Systems) [PROTOS 89, PROTOS 90], lagen wesentliche Forschungsschwerpunkte der Universität Dortmund in der logischen Programmierung. Vor diesem Hintergrund ergab sich der erste Kontakt mit deduktiven Datenbanken durch die Arbeiten von John W. Lloyd und Rodney W. Topor [Lloyd 85, Lloyd 86]. Das von ihnen vorgeschlagene Konzept deduktiver Datenbanken aus der Sicht der logischen Programmierung bildete den Ausgangspunkt für die ersten beiden der oben genannten Diplomarbeiten. Dieser Ansatz wird aufgrund seiner Allgemeinheit gegenüber den heute verbreiteten Vorschlägen als recht außergewöhnlich angesehen. In dem vorliegenden Buch wird dieses interessante Modell weiter ausgebaut und seine Vorteile, aber auch seine Grenzen aufgedeckt.

Adressaten

Das Buch richtet sich sowohl an Lernende als auch an Lehrende der Informatik. Voraussetzung für das Studium des Buches sind elementare Informatik- und Mathematikkenntnisse, wie sie im universitären Grundstudium vermittelt werden. Kenntnisse der logischen Programmierung (etwa durch den Umgang mit der Programmiersprache Prolog) sowie Erfahrungen mit (relationalen) Datenbanken erweisen sich als hilfreich, sind aber nicht unbedingt notwendig. Übungsaufgaben geben dem Leser Gelegenheit, den besprochenen Stoff zu vertiefen. Das Buch ist somit für das Selbststudium geeignet, bietet sich aber auch als Grundlage für eine Spezialvorlesung an.

Hinweise an die Leser

Die Kapitel des Buches müssen nicht notwendigerweise in der vorgegebenen Reihenfolge gelesen werden. Die folgenden Hinweise zeigen kurz die Abhängigkeiten der einzelnen Kapitel auf und geben dem Leser einen Leitfaden zum Studium des Buches an die Hand.

Kapitel 2 und 3 schaffen die zum Verständnis der übrigen Kapitel notwendigen Voraussetzungen. Kapitel 2 gibt einen Einblick in die allgemeinen Grundlagen der Prädikatenlogik und Kapitel 3 führt darauf aufbauend ein spezielles Grundmodell deduktiver Datenbanken ein. Die nachfolgenden Kapitel beschäftigen sich jeweils mit der Erweiterung dieses Grundmodells um ein ausgewähltes Konzept. Sie basieren auf den Grundlagen der ersten beiden Kapitel und können im Prinzip in einer beliebigen Reihenfolge durchgearbeitet werden.

Der praktische Teil des Buches setzt sich aus den Anhängen A bis C zusammen. Anhang A führt zunächst die Programmiersprache Prolog ein. Bei ausreichenden Prolog-Kenntnissen kann auf das Studium dieses Anhangs verzichtet werden. Anhang B spezifiziert die äußere Form eines einfachen Datenbanksystems, dessen Realisierung Gegenstand von Anhang C ist. Die Kenntnis der Anhänge A und B sind somit eine wesentliche Voraussetzung für das Verständnis von Anhang C.

Anhang D stellt eine Formelsammlung dar und dient als Nachschlagewerk für wesentliche mathematische Begriffe, die in den übrigen Kapiteln zwar verwendet, aber nicht eingeführt werden. Mit Hilfe dieses Anhangs haben wir versucht, das Buch so weit wie möglich in sich geschlossen zu halten.

Der teilweise formale und enzyklopädische Charakter des Buches sollte keinen Leser abschrecken. Viele der Beweise sind der Vollständigkeit halber aufgenommen und können häufig ohne Verständnisverlust übersprungen werden.

Konventionen und Notationen

Im Buch werden zwei Arten von Beispielen unterschieden, für die jeweils ein anderer Schriftsatz verwendet wird. Akademische Beispiele, die auf keine spezielle Anwendung Bezug nehmen, sind *kursiv* geschrieben. An realistische Applikationen angelehnte Beispiele werden hingegen in Schreibmaschinenschrift gesetzt. Um Beispielprädikate und -funktionen im laufenden Text hervorzuheben, sind die entsprechenden Bezeichner ausnahmslos der englischen Sprache entnommen.

Theoremartige Strukturen wie Definitionen, Sätze und Beispiele werden ebenso wie Beweise mit einem ■ abgeschlossen. Die Numerierung der verschiedenen Strukturen (mit Ausnahme der Aufgaben) erfolgt fortlaufend innerhalb eines Kapitels. Aufgaben erhalten eine eigene kapitelweise Numerierung. Damit sie im Text leicht auffindbar sind, wird ihre erste Zeile am linken Rand mit einem ► gekennzeichnet.

Wir haben uns bemüht, soweit wie möglich Anglizismen zu vermeiden. Viele in der englischsprachigen Fachliteratur verbreitete Begriffe wurden ins Deutsche übersetzt. Wenn es sich um keine direkte Übersetzung handelt (wie z. B. der Begriff *blockieren* für das englische Wort *to flounder*), geben wir bei der ersten

Verwendung zusätzlich das englische Original an. Schwer zu übersetzende Fachtermini, wie z. B. Narrowing, top-down oder bottom-up, die sich teilweise im deutschen Sprachgebrauch eingebürgert haben, werden beibehalten. Für diese Begriffe gibt es nur unzureichende oder unklare Übersetzungen, die eher verwirrend als hilfreich sind. Auf diese Weise hoffen wir, den Lesern die spätere Lektüre englischsprachiger Literatur zu erleichtern.

Zum Schluß möchten wir noch eine Anmerkungen zu weiblichen und männlichen Formen der Anrede machen: Wenn von „Lesern“ bzw. „Anwendern“ gesprochen wird, meinen wir stets Leserinnen und Leser bzw. Anwenderinnen und Anwender.

Danksagungen

Die Entstehung dieses Buches wäre nicht möglich gewesen ohne die tatkräftige Unterstützung von zahlreichen Freunden und Kollegen, denen wir an dieser Stelle unseren besonderen Dank aussprechen möchten.

An erster Stelle sei Jürgen Kalinski genannt, aus dessen Feder Abschnitt 3.4 über alternative Ansätze zur Semantikdefinition stammt. Darüber hinaus möchten wir allen Korrektoren danken, ohne deren konstruktive Kritik das Buch nicht in der jetzigen Form vorliegen würde: Holger Berse, Stefan Braß, Wolfram Burgard, Michael Hanus, Ullrich Hustadt, Katharina Just, Thomas Lemke, Rainer Manthey, Lutz Plümer, Heinz Rottmann sowie Stefan Lüttringhaus-Kappel, dessen Diplomarbeit den Ausgangspunkt zu diesem Buch bildete. Unser Dank gebührt auch Norbert Olligschläger, der die neuesten T_EX- und METAFONT-Versionen installiert hat, so daß dieses Manuskript aus technischer Sicht ohne Schwierigkeiten erstellt werden konnte. Schließlich möchten wir den Mitarbeitern des Vieweg-Verlages, insbesondere Reinald Klockenbusch, unseren Dank für die jahrelange, reibungslose Zusammenarbeit aussprechen.

An unsere Leser richten wir die Bitte, uns Reaktionen jeglicher Art (Hinweise auf Fehler, Verbesserungen und Anregungen) zukommen zu lassen (e-mail an `dabb@informatik.uni-bonn.de`).

Bonn, im Oktober 1993

Armin B. Cremers
Ulrike Griefahn
Ralf Hinze

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ein Bundesbahninformationssystem	6
1.2	Gliederung des Buches	15
2	Prädikatenlogik	17
2.1	Syntax	17
2.1.1	Signaturen	18
2.1.2	Terme	19
2.1.3	Formeln	24
2.1.4	Freie und gebundene Variablen	26
2.2	Semantik	28
2.2.1	Interpretationen und Modelle	28
2.2.2	Der semantische Folgerungsbegriff	34
2.2.3	Normalformen von Formeln	35
2.2.4	Herbrand-Interpretationen	38
2.3	Kalküle	41
2.3.1	Logik-Kalküle	41
2.3.2	Substitution und Unifikation	44
2.3.3	Der Resolutionskalkül	49
3	Deduktive Datenbanken	53
3.1	Datenbanken als logische Programme	54
3.1.1	Syntax deduktiver Datenbanken	54
3.1.2	Semantik deduktiver Datenbanken	57
3.1.3	Relationale Datenbanken	63
3.2	Normale Programme	68
3.2.1	Syntax normaler Programme	68
3.2.2	Transformation in normale Programme	69
3.3	Anfrageauswertung durch SLDNF-Resolution	73
3.3.1	SLDNF-Resolution	75
3.3.2	Korrektheit der SLDNF-Resolution	81
3.3.3	Relative Vollständigkeit der SLDNF-Resolution	84
3.3.4	Sichere Programme	96

3.4	Alternative Ansätze zur Semantikdefinition	101
3.4.1	Semantik definiter Programme	102
3.4.2	Stratifizierungs-Ansätze	104
3.4.3	Die „stable model“-Semantik	106
3.4.4	Vervollständigungs-Semantiken	107
3.4.5	Die „well-founded“-Semantik	108
4	Alternative Auswertungsmethoden	111
4.1	Datalog und Relationenalgebra	112
4.2	Bottom-up-Auswertung	116
4.2.1	Naive Auswertung	117
4.2.2	Semi-naive Auswertung	120
4.3	Magic Sets	123
4.4	OLD-Resolution mit Tabellarisierung	131
4.5	Bottom-up and Top-down im Vergleich	141
4.5.1	Reine Auswertungsverfahren	141
4.5.2	Gemischte Auswertungsstrategien	144
5	Typsysteme	147
5.1	Polymorphe Programme	149
5.1.1	Typsignaturen	149
5.1.2	Polymorphe Terme	150
5.1.3	Polymorphe Formeln	156
5.2	Polymorphe Substitution und Unifikation	161
5.3	Optimierung der Unifikation	164
5.3.1	Parametrisierte Form	165
5.3.2	Irrelevante Argumente	169
5.4	Typinferenz	176
5.4.1	Wohlgetypte Klauseln	176
5.4.2	Rekonstruktion der Typannotationen	178
6	Statische Transformationen	183
6.1	Modussysteme	184
6.1.1	Modusdeklarationen	186
6.1.2	Modusprüfung	192
6.1.3	Mehrfache Modusdeklarationen	198
6.1.4	Modusinferenz	202
6.1.5	Modusgesteuerte Transformation	208
6.1.6	Transformation polymorpher Programme	213
6.2	Auswertbare Funktionen	214
6.2.1	Syntax bedingter Gleichungen	216
6.2.2	Transformation bedingter Gleichungen	220
6.3	Listenbeschreibungen	228
6.3.1	Syntax von Listenbeschreibungen	231
6.3.2	Transformation von Listenbeschreibungen	236

7	Integritätsprüfung	241
7.1	Grundlegende Begriffe	244
7.2	Vereinfachungsverfahren	245
7.2.1	Integritätsprüfung in relationalen Datenbanken	245
7.2.2	Integritätsprüfung in stratifizierten Datenbanken	249
7.3	Top-down-Integritätsprüfung	255
7.3.1	Konsequenzen von Datenbankänderungen	256
7.3.2	Markierte Und-Oder-Bäume	263
7.3.3	Die Integritätsprüfung	273
7.4	Integritätsprüfungen im Vergleich	280
8	Änderungen	285
8.1	Ein Überblick über Änderungssprachen	287
8.1.1	Prozedurale Änderungssprachen	287
8.1.2	Deklarative Änderungssprachen	290
8.2	Eine Änderungssprache für deduktive Datenbanken	294
8.2.1	Primitive Änderungsanweisungen	295
8.2.2	Komplexe Änderungsanweisungen	297
8.2.3	Änderungsprozeduren	303
8.2.4	Transaktionen	307
8.3	Änderung abgeleiteter Prädikate	308
8.3.1	Problematik der Sichtenänderung	310
8.3.2	Automatische Sichtenänderung	313
8.3.3	Änderungsprozeduren für Sichten	317
A	Prolog	321
A.1	Prolog-Interpreter	322
A.2	Korrektheit	326
A.3	Vollständigkeit	328
A.4	Effizienz	330
A.5	Negation	333
A.6	Kontrolle	335
A.7	Datenbankprädikate	336
A.8	Termmanipulation	338
A.9	„all solutions“-Prädikate	341
B	Benutzungsschnittstelle	345
B.1	Das Quellformat	345
B.1.1	Deklarationen	346
B.1.2	Klauseln und Änderungen	350
B.1.3	Die Standardumgebung	353
B.1.4	Bibliotheken	354
B.2	Anweisungen der Schnittstelle	355
B.3	Beispiel: Fahrraddatenbank	358
B.3.1	Die Datenbank	358

B.3.2	Anfragen	363
B.3.3	Änderungen	365
B.4	Beispiel: Graphen	368
B.4.1	Die Datenbank	368
B.4.2	Anfragen	375
B.4.3	Änderungen	377
C	Realisierung in Prolog	379
C.1	Übersetzung der Quelldatei	380
C.1.1	Syntaxanalyse	382
C.1.2	Meta-Programmierung	382
C.1.3	Deklarationen	385
C.1.4	Typinferenz	386
C.1.5	Normalisierung	389
C.1.6	Transformation in die relationale Form	393
C.1.7	Modusprüfung	397
C.1.8	Optimierung der Unifikation	402
C.2	Bearbeitung von Anfragen	404
C.3	Durchführung der Integritätsprüfung	407
C.4	Bearbeitung von Änderungsanweisungen	413
D	Mathematische Grundlagen	419
D.1	Logik	419
D.2	Mengen, Relationen und Abbildungen	420
D.3	Äquivalenzrelationen	423
D.4	Monoide	423
D.5	Natürliche Zahlen, Folgen und Bäume	424
D.6	Graphen	426
D.7	Induktive Definitionen	427
D.8	Transitionssysteme	430
D.9	Halbordnungen und vollständige Induktion	430
D.10	Verbände und Fixpunkte	433
D.11	Berechenbarkeit und Komplexität	436
	Literaturverzeichnis	439
	Sachwortverzeichnis	451