

Logische Programmierung und deduktive Datenbanken

Kapitel 0: Organisatorisches

Prof. Dr. Stefan Brass

Martin-Luther-Universität Halle-Wittenberg

Sommersemester 2023

<http://www.informatik.uni-halle.de/~brass/lp23/>

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches
- 4 Literatur
- 5 Software
- 6 Schlussbemerkungen

Themen

- Programmierung in der Sprache Prolog.
 - Ziel: Praktische Programmierfähigkeiten.
 - Semantik (min. Modell), Ausführung (SLD-Resolution).
- Implementierung der Prolog-Teilmenge „Datalog“ mit Datenbank-Techniken:
 - Bottom-Up Auswertung.
 - Optimierung mittels „Magischer Mengen“.
- Nichtmonotone Negation.
- Weitere Themen (sofern Zeit):
Constraint Logic Programming, Answer Set Programming, ...

Inhalt

- 1 Inhalte
- 2 Motivation**
- 3 Organisatorisches
- 4 Literatur
- 5 Software
- 6 Schlussbemerkungen

Motivation

Warum Prolog und Deduktive DBen lernen?

- Eventuell deutlich gesteigerte Produktivität.
- Horizont-Erweiterung, ganz andere Sprache:
Programming in Logic.
- Deklarative Programmierung (erfolgreich für DB-Abfragen in SQL) ist wichtig für Skalierbarkeit.
- Z.B. Rekursion in SQL-99: Hier in Reinform.
Datalog ist viel einfacher als SQL. Ideen so besser zu verstehen.
- Grundlage klassischer künstlicher Intelligenz.
- Forschungsthema meiner Gruppe: Mitwirkung in der Forschung möglich, z.B. über Master-Arbeiten.

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches**
- 4 Literatur
- 5 Software
- 6 Schlussbemerkungen

Zeit und Ort

Vorlesung (2 SWS):

- Donnerstags, 14¹⁵–15⁴⁵, Raum 3.04 oder Aufzeichnung.

Der Dozent wäre enttäuscht, wenn extrem wenig Studierende in Präsenz kommen. Nutzen Sie die Gelegenheit, Fragen zu stellen! Die Aufzeichnung ist hauptsächlich für Notfälle gedacht, nicht als Normalfall.

- Aufzeichnung in **StudIP**, Reiter „**Opencast**“.

Ab ca. 17–18 Uhr (Verarbeitungszeit der Rohdaten).

Übung (2 SWS):

- Donnerstags, 10¹⁵–11⁴⁵, Multimedia-Pool 3.02
- Beginn: Zweite Semesterwoche (14.04.2023).

Zur Video-Aufzeichnung

- Sie sind auf dem Video fast nicht zu hören, wenn Sie eine Frage stellen.

Deswegen sollte ich daran denken, die Frage zu wiederholen.

- Sie sind auf dem Video auch nicht zu sehen.

Außer, wenn Sie zu spät kommen und durch das Bild gehen.

- Betonung von besonders wichtigen Inhalten der Folien

Die Folien sind nicht wirklich zum Selbststudium gedacht (hauptsächlich als Gedächtnisstütze für den Dozenten). In der Vorlesung werden manche Folien überblättert (nur zum Nachschlagen), andere ausführlich behandelt (mit Hinweis auf Prüfungsrelevanz und ggf. zusätzlichen Erläuterungen/Beispielen).

- Regelmäßige Beschäftigung mit dem Thema ist wichtig!

Man bleibt so eher dran, und merkt ggf. noch rechtzeitig, wenn man etwas wiederholen muss (oder Fragen stellen sollte).

Zeitliche Belastung

- Für dieses Modul gibt es 5 Leistungspunkte (LP).
- Aufteilung von 150 Stunden Arbeitszeit:

Lernform	SWS	Std.
Vorlesung	2	30
Selbststudium (Buch in Semesterpause)	0	45
Übung	2	30
Hausaufgaben/direkte Nacharbeit	0	30
Spezielle Prüfungsvorbereitung	0	15

Falls Sie hoffen, mit weniger auszukommen: Es funktioniert nicht.

Falls deutlich mehr als zwei Stunden Aufwand für HA: Bitte melden.

Voraussichtlich müssen Sie einen kurzen Seminarvortrag halten (10–15 min, s.u.).

Die Vorbereitungszeit würde vom „Selbststudium“ abgehen.

Studienleistung

- Voraussetzung für erfolgreichen Abschluss des Moduls (nicht für Teilnahme an Prüfung). Kriterien:
 - Hausaufgaben (50% der Punkte, teils 0 oder 1)
 - Eventuell größere Programmieraufgabe am Semesterende.
 - Auch: Lesen von Literatur, ggf. mit kurzem Bericht/Vortrag.
 - Aufgaben müssen einzeln bearbeitet werden (keine Gruppen)!
 - Zu ähnliche Lösungen führen zu Ärger (u.a. 0 Punkte für alle Beteiligten). Jeder muss seine Lösungen erklären können.
 - Aktive Mitarbeit in den Übungen
 - Z.B. Vorrechnen von Haus- und Präsenzaufgaben, Beantwortung von Fragen zum Umfeld der Aufgabe, Stellen von Fragen (!).
 - Ggf. kurze Präsentation über Literatur, System, Projektaufgabe.
 - Sie dürfen nicht zu häufig fehlen (3 Mal sind ok).
- Implizite HA zu jeder Übung: Mindestens eine Frage überlegen.

Modulleistung/Prüfung

- Da sich nur 11 Teilnehmer zur Vorlesung angemeldet haben, wird die Prüfung voraussichtlich mündlich sein (ggf. per Videokonferenz).
- Geplante Daten: 30./31.08.2023 und 11.–13.10.2023.

Achten Sie auf Ankündigungen wegen möglicher Änderungen (z.B. StudIP EMail-Verteiler). Teilen Sie mir ggf. Terminkonflikte möglichst frühzeitig mit. Eventuell liegen einige Prüfungen auch am Folgetag.
- Beispiele für mündliche Prüfungsfragen werden in der Vorlesung genannt (siehe auch Übungszettel).

Eine kurze Prolog Programmieraufgabe (auf Papier) gehört sicher dazu. Schauen Sie sich auch die alten Klausuren auf der Webseite an.

Seminarvortrag

- Offiziell ist diese Veranstaltung ein Seminar.

Das hat mit der Kapazitätsrechnung zu tun, die Vorlesungen immer mit 100 Teilnehmern ansetzt (d.h. hier nur 11% der SWS anrechnet).

- Ich werde einige von meinen Vorträgen bei Konferenzen einbauen (dies ist ja eine Lehrveranstaltung zu meinem Haupt-Forschungsthema).
- Sie müssen aber auch einen kleinen Seminarvortrag halten (10–15 min):
 - Entweder als Einstieg in die Prüfung, oder
 - als Hausaufgabe mit 45% der Punkte (d.h. Pflicht-HA).
- Außerdem stellen Sie natürlich Ihre Hausaufgaben in Kurz-Vorträgen vor.

Ansprechpartner (1)

Dozent (Vorlesung): Prof. Dr. Stefan Brass

- Email: brass@informatik.uni-halle.de

Betreff-Zeile sollte mit [1p23] beginnen, möglichst aussagefähig.

- Büro: Von-Seckendorff-Platz 1, Raum 313
- Telefon: 0345/55-24740
- Sprechstunde: Montags, 12⁰⁰–13⁰⁰

In den Semesterferien nur nach Vereinbarung.

- Promotion: „Defaults in deduktiven Datenbanken“.
- Habilitation: „Bottom-Up Query Evaluation in Extended Deductive Databases“.

Ansprechpartner (2)

Weitere Mitarbeiter der Gruppe:

- MSc. Mario Wenzel

Raum 315, Telefon: 0345/55-24776, Email: mario.wenzel@...

Forscht im Bereich deduktiver Datenbanken!

- PD Dr. Alexander Hinneburg

Raum 314, Telefon: 0345/55-24732, Email: hinneburg@...

Sekretärin: Ramona Vahrenhold

- Büro: Von-Seckendorff-Platz 1, Raum 324

- Telefon: 0345/55-24750, Fax: 0345/55-27333

- Email: vahrenhold@informatik.uni-halle.de

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches
- 4 Literatur**
- 5 Software
- 6 Schlussbemerkungen

Vorlesungs-Materialien

Folien:

- ca. 300–500 Folien auf Englisch.
- Jeweils nach Fertigstellung auf der Web-Seite:
[\[http://www.informatik.uni-halle.de/~brass/lp23/\]](http://www.informatik.uni-halle.de/~brass/lp23/)
- Es gibt notfalls (ganz) alte Skripte auf Deutsch.

Zusätzlich:

- Webseite: Viele nützliche Links.
Wenn Sie weitere nützliche Quellen finden, schicken Sie mir eine EMail.
- Forum in StudIP

Prolog:

- Springer, 2003, 5th Ed., 299 Seiten, ISBN 3540006788, 46.24 €.

Lehrbücher (2)

Prolog (Forts.):

- Richard O'Keefe:
The Craft of Prolog.
MIT Press, 2009 (1990?), 416 Seiten, ISBN 978-0262512275,
Taschenbuch, 32.49 €.
- Ulf Nilson, Jan Małuszyński:
Logic, Programming, and Prolog (2nd Ed.).
1995, [<http://www.ida.liu.se/~ulfni/lpp>]

Theorie der Logischen Programmierung:

- John W. Lloyd: Foundations of Logic Programming.
2nd Edition 1987, ISBN 3642831915, 224 pages.
[<https://link.springer.com/book/10.1007/978-3-642-83189-8>]

Lehrbücher (3)

Deduktive Datenbanken:

- Sergio Greco, Cristian Molinaro:
Datalog and Logic Databases.

Morgan & Claypool, 2016, ISBN 1-62705-113-9, 170 Seiten, 61.98 €.

- Armin B. Cremers, Ulrike Griefahn, Ralf Hinze:
Deduktive Datenbanken.

Vieweg, 1994, 480 Seiten, ISBN 3528047003, 54.99 €.

- Robert M. Colomb:
Deductive Databases and Their Applications.

Taylor&Francis, 1998, 288 Seiten, ISBN 0748407979, 100.31 €.
Gebundene Ausgabe: ISBN 0748407960.

Lehrbücher (4)

Constraint Logic Programming:

- Thom Frühwirth, Slim Abdennadher: Constraint Programmierung. Grundlagen und Anwendungen.

Springer, 1997, 165 Seiten, ISBN 354060670X, 22.99 Euro.

- Krzysztof R. Apt, Mark Wallace: Constraint Logic Programming using ECLiPSe.

Cambridge Univ. Press, 2006, ISBN 0521866286, 348 Seiten, 32.89 €.

[https://www.researchgate.net/publication/220693610-Constraint_logic_programming_using-ECLiPSe]

- Petra Hofstedt, Armin Wolf: Einführung in die Constraint-Programmierung.

Springer, 2007, ISBN 3540231846, 388 Seiten, 32.99 €.

Lehrbücher (5)

DB-Lehrbücher mit Berücksichtigung Deduktiver DBen:

- Serge Abiteboul, Richard Hull, Victor Vianu: Foundations of Databases.

[<http://webdam.inria.fr/Alice/>]

- Raghu Ramakrishnan, Johannes Gehrke: Database Management Systems, 3rd Ed.

[<https://github.com/Premiumlab/Database-Management-System-Ramakrishnan-Gehrke-3rd-Edition>]

- Alfons Kemper, André Eickler:
Datenbanksysteme, Eine Einführung, 10. Auflage.

Übersichtsartikel etc. (1)

- Francois Bancilhon / Raghu Ramakrishnan: An Amateur's Introduction to Recursive Query Processing Strategies.

Proc. ACM SIGMOD Int. Conf. on Management of Data, 1986.

[<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.133.8870>]

- François Bry / Dietmar Seipel:
Deduktive Datenbanken — das aktuelle Schlagwort.

Informatik Spektrum, Vol. 19, 1996, 214–215.

[<http://www.pms.ifi.lmu.de/publikationen/sonstiges/aktuelles-schlagwort-deduktive-datenbanken.ps>]

Übersichtsartikel etc. (2)

- Stefano Ceri / Georg Gottlob / Letizia Tanca:
What you always wanted to know about Datalog
(and never dared to ask).

IEEE Transactions on Knowledge and Data Engineering, Vol. 1, 1989, 146–166.

[<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.210.1118>]

- Evgeny Dantsin, Thomas Eiter, Georg Gottlob, Andrei Voronkov: Complexity and expressive power of logic programming.

ACM Computing Surveys, Sep. 2001, Vol. 33, No. 3, 374–425.

[<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.8271>]

Übersichtsartikel etc. (3)

- Mengchi Liu: Deductive database languages: problems and solutions.

ACM Computing Surveys, March 1999, Vol. 31, No. 1, 27–62.

[<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.42.793>]

- John Grant / Jack Minker:
The Impact of Logic Programming on Databases.

Communications of the ACM 35 (3), March 1992, 67–81.

[<https://dl.acm.org/citation.cfm?id=131297>]

- Werner Kießling / U. Güntzer: Deduktive Datenbanksysteme auf dem Weg zur Praxis.

Informatik Forschung u. Entwicklung 5, 1990, 177–187. (+Vol.7, S.41)

Übersichtsartikel etc. (4)

- Jack Minker:
Logic and Databases: A 20 Year Retrospective.
D. Pedreschi / C. Zaniolo (Ed.): Logic in Databases, Int. Workshop (LID'96), Springer LNCS 1154, 1996, 3–57.
[\[http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.2337\]](http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.2337)
- Raghu Ramakrishnan/Jeffrey D. Ullman: A Survey of Research in Deductive Database Systems.
The Journal of Logic Programming, Vol. 23, 1995, 125–149.
[\[http://dbpubs.stanford.edu:8090/pub/1995-14\]](http://dbpubs.stanford.edu:8090/pub/1995-14)

Übersichtsartikel etc. (5)

- Raghu Ramakrishnan (Ed.):
Applications of Logic Databases.

Kluwer Academic Publishers, 1995.

- Kotagiri Ramamohanarao (Ed.): Special Issue on
Prototypes of Deductive Database Systems.

The VLDB Journal, Vol. 3, No. 2, 1994.

[<http://www.informatik.uni-trier.de/~ley/db/journals/vldb/vldb3.html>]

- Shalom Tsur:
Deductive Databases in Action.

Proc. 10th ACM Symp. on Principles of Database Syst. (PODS'91), 1991,
205–218. [<https://dl.acm.org/citation.cfm?id=113426>]

Übersichtsartikel etc. (6)

- Laurent Vieille: From Data Independence to Knowledge Independence: An on-going Story.

VLDB'98, 650–654, [<http://www.vldb.org/conf/1998/p650.pdf>]

- Jeffrey D. Ullman, Carlo Zaniolo: Deductive databases: achievements and future directions.

ACM SIGMOD Record, Vol. 19, Issue 4, Dec. 1990, 75–82.

[<https://dl.acm.org/citation.cfm?id=122067>]

- Carlo Zaniolo: Intelligent Databases: Old Challenges and New Opportunities.

Journal of Intelligent Information Syst., Vol. 1, 1992, 271–292.

[<http://web.cs.ucla.edu/~zaniolo/papers/jiis92.ps>]

Andere Vorlesungen im Web

- Wolfgang May: Deductive Databases
[<https://www.dbis.informatik.uni-goettingen.de/Teaching/DD-SS21/>]
- Seipel: Deduktive Datenbanken
[<http://www1.pub.informatik.uni-wuerzburg.de/databases/courses/ddb/Folien.pdf>]
- Zaniolo: Advanced Data Bases & Knowledge Bases
[<http://web.cs.ucla.edu/classes/spring03/cs240B/>]
- Ullman: CS345
[<http://infolab.stanford.edu/~ullman/cs345-notes.html>]
- Abiteboul: Databases
[<http://abiteboul.com/TEACHING/DBCOURSE/>]

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches
- 4 Literatur
- 5 Software**
- 6 Schlussbemerkungen

Software: Prolog (1)

Prolog-Systeme (Open Source):

- SWI-Prolog (Linux, Windows, MacOS X, Quellen)

<http://www.swi-prolog.org/> (einfach, problemlos, nett)

- XSB (Unix, Windows, Quellen)

[<http://xsb.sourceforge.net/>]

- GNU Prolog (Unix, Windows, Quellen)

[<http://gnu-prolog.inria.fr/>]

- Ciao Prolog System (Unix, Windows, Quellen)

[<http://clip.dia.fi.upm.es/~clip/Software/Ciao/>]

Software (2)

Prolog-Systeme (kommerziell):

- ECLiPSe

<http://eclipseclp.org/> (War früher kommerziell, jetzt Open Source).

- SICStus Prolog

Personal License (nicht kommerziell): 132 Euro.

30 Tage Evaluation: kostenlos. [<http://www.sics.se/sicstus/>]

- Quintus Prolog (jetzt von SICS vertrieben)

- Visual Prolog: Nicht kompatibel, nur Windows.

[<https://www.visual-prolog.com/>]

Für Lern-/Lehrzwecke Personal Version kostenlos.

Bei mir funktionierte Version 6.1 nicht, aber Version 5.2.

Software (3)

Weitere Logische Programmiersprachen:

- Mercury

[<https://mercurylang.org/>]

- Gödel

[https://dennisdarland.com/my_sw_projects/goedel/v1_3_27/doc/book.pdf]

[https://dennisdarland.com/my_sw_projects/goedel/]

[[https://en.wikipedia.org/wiki/Gödel_\(programming_language\)](https://en.wikipedia.org/wiki/Gödel_(programming_language))]

Siehe auch:

- Logic Programming Repositories and Resources

[https://cliplab.org/logic_programming_repositories.html]

Software (5)

DOOD-Systeme (deduktiv, objektorientiert):

- FLORID/FloXML: OO, F-Logik (Solaris, Linux)

[<http://www.informatik.uni-freiburg.de/~dbis/florid/>]

- ROCK&ROLL

[<http://www.cee.hw.ac.uk/Databases/rnr.html>]

- ConceptBase

[<http://www-i5.informatik.rwth-aachen.de/CBdoc/>]

- Validity (kommerziell, nicht erhältlich?)

Software (6)

Neuere Projekte:

- DES — Datalog Educational System
 - [<http://sourceforge.net/projects/des/>]
 - [<http://www.fdi.ucm.es/profesor/fernan/des/>]
- bddbldb — BDD-Based Deductive DataBase
 - [<http://bddbdb.sourceforge.net/>]
- Lightweight Deductive Database System (MITRE Corporation)
 - [<http://www.ccs.neu.edu/home/ramsdell/tools/datalog/>]

Software (7)

Ganz neues Projekt:

- Soufflé: [<https://github.com/souffle-lang>]
- Datalog wird hier zur statischen Analyse für Java-Programme verwendet.

Das System kann aber beliebige Datalog-Programme ausführen.

- B. Scholz, H. Jordan, P. Subotić, T. Westmann: On Fast Large-Scale Program Analysis in Datalog.

Proc. of the 25th International Conf. on Compiler Construction, ACM, 2016, 196–206. [<https://dl.acm.org/citation.cfm?id=2892226>]

- Tutorial: [<http://souffle-lang.org/docs/datalog/>]

Software (8)

Kommerzielle Nutzung einer Datalog-Variante:

- LogicBlox [<http://www.logicblox.com/>]

„Cloud-delivered platform for actionable Big Data applications“

- Anspruch: Nur ein System und eine Sprache statt Wirrwarr verschiedener DBen, Sprachen und Werkzeuge wie in großen Firmen üblich.

Eventuell können wir das System über eine Kooperationsvereinbarung ausprobieren. Es gibt auch mehrere Veröffentlichungen, z.B.:

Aref, ten Cate, Green, Kimelfeld, Olteanu, Pasalic, Veldhuizen, Washburn: Design and Implementation of the LogicBlox system. In: Proc. of the 2015 ACM SIGMOD International Conf. on Management of Data, 1371–1382.

ACM (2015). [<https://developer.logicblox.com/wp-content/uploads/2016/01/logicblox-sigmod15.pdf>]

Inhalt

- 1 Inhalte
- 2 Motivation
- 3 Organisatorisches
- 4 Literatur
- 5 Software
- 6 Schlussbemerkungen**

Projekt meines Lehrstuhls

- Übersetzung von Datalog nach C++.
[\[http://users.informatik.uni-halle.de/~brass/push/\]](http://users.informatik.uni-halle.de/~brass/push/)
- Alternativ abstrakte Maschine.
- Sehr vielversprechende Benchmark-Ergebnisse!
[\[https://dbs.informatik.uni-halle.de/rbench/\]](https://dbs.informatik.uni-halle.de/rbench/)
- Es gibt noch viel zu tun (Herzliche Einladung!):
 - Parallele Ausführung.
 - Negation, Funktionssymbole/Arrays.
 - Programmanalyse, Optimierungen.
 - Spracherweiterungen.
- Weiteres Projekt: Arduino-Programmierung in Datalog.

Verbesserung der Lehre

- Die Vorlesung soll kein Monolog werden:
Fragen sind sehr willkommen.
Gerade bei einem Gebiet, das viel mit Forschung zu tun hat, kann es spannende Diskussionen geben. Ihre Ideen interessieren mich.
- Falls Sie Fehler in den Vorlesungsmaterialien finden, nützliche Links für die WWW-Seite entdecken, oder andere Vorschläge zur Verbesserung der Vorlesung haben:
Teilen Sie es mir bitte mit.
Falls Sie Probleme mit der Vorlesungs-Aufzeichnung entdecken (z.B. Teile, die man besser herausschneiden sollte), teilen Sie mir das auch bitte mit.
- Fehler in Programmen müssen aufgeklärt werden.
Nur so lange basteln, bis es zu funktionieren scheint, ist keine Lösung. Man muss auch verstehen, warum es vorher nicht funktioniert hat. Ich bin bereit, zu helfen (so weit ich kann).