Aufgabe 3: Deduktive Datenbanken und Unifikation

Teil 1: Familien-Datenbank

Uwe Krause
HAW-Hamburg

Referat eingereicht im Rahmen der Vorlesung Logik und Berechenbarkeit

im Studiengang Angewandte Informatik (AI)
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. C. Klauck

Abgegeben am 22.11.2016

Fakultät Technik und Informatik Department Informatik	Faculty of Engineering and Computer Science Department of Computer Science

Inhaltsverzeichnis

1. Vorfahre / Nachkomme

> *"Definieren Sie zwei Prädikat vorfahre(X,Y) (descendant) und nachkomme(X,Y) (ancestor), die alle Vorfahren bzw. Nachkommen von Y (nacheinander) berechnen können."*

Vorfahre

Als Vorfahre muss jedes Elternteil erkannt werden und von diesen Elternteilen jeweils rekursiv wieder alle Vorfahren, solange bis keine weiteren Elternteile mehr in der Datenbank sind.

Einfachster Fall: Vorfahre ist direkt ein Elternteil der Person,
oder speziellerer Fall: Vorfahre ist ein Vorfahre eines Elternteils der Person.

Eine Person V ist Vorfahre der Person P,
wenn es einen Menschen E gibt, der ein Elternteil der Person P ist
und wenn V ein Vorfahre des Menschen E ist.

$$\forall\ V, P : [\ \text{vorfahre}(V, P) \leftarrow\ \exists\ E : [\ \text{elternteil}(E, P)\ \wedge\ \text{vorfahre}(V, E)\ ]\ ]$$

lässt sich umformen[1] zu:

$$\forall\ V, P, E : [\ \text{vorfahre}(V, P) \leftarrow \text{elternteil}(E, P)\ \wedge\ \text{vorfahre}(V, E)\ ]$$

Diese allquantisierte Formel kann, gemeinsam mit einer entsprechenden Wissensbasis die e/2 als Fakten bereitstellt, nun automatisiert von einem Programm, das nach dem "Beweisermodell" arbeitet, ausgewertet werden.[2]

Nachkomme

Ein Nachkomme N einer Person P ist jede Person N, die die Person P als Vorfahre hat.

$$\forall\ N, P : [\ \text{nachkomme}(N, P) \leftarrow \text{vorfahre}(P, N)\ ]$$

[1] Diese Behauptung wird in Aufgabe 2. ("Geschwister") bewiesen
[2] Dies wird in Aufgabe 6 detailliert gezeigt

2. Geschwister

> *"Definieren Sie ein Prädikat geschwister(X,Y) (siblings). Sie können die Bedingung X ungleich Y (in Prolog X \== Y) verwenden."*

Zwei Menschen sind Geschwister, wenn sie (mindestens) einen gemeinsamen Elternteil haben.

Etwas förmlicher lässt sich dies so formulieren:

- "Zwei Personen[3] sind Geschwister,
  wenn es eine andere Person gibt, die Elternteil beider Personen ist."

Formale Definition:

Seien X, Y, E ∈ des Universums aller Personen, X ≠ Y,
[und] Gültigkeitsbereich der Quantoren

	Für alle Personen X und Y gilt,	X und Y sind Geschwister	wenn	es eine Person E gibt, für die gilt	E ist Elternteil von X	und	E ist Elternteil von Y	
	∀ X, Y : [	g(X, Y)	←	∃ E : [	(e(E, X)	∧	e(E, Y))]]	// implikation
≡	∀ X, Y : [	g(X, Y)	∨	¬ ∃ E : [	(e(E, X)	∧	e(E, Y))]]	// ∀¬ ≡ ¬∃
≡	∀ X, Y : [	g(X, Y)	∨	∀ E : [¬	(e(E, X)	∧	e(E, Y))]]	// ∀ n. links
≡	∀ X, Y, E : [	g(X, Y)	∨	¬	(e(E, X)	∧	e(E, Y))]	// implikation
≡	∀ X, Y, E : [	g(X, Y)		←	(e(E, X)	∧	e(E, Y))]	

Diese Umformung führt zu einer allquantisierten Formel, die wir als Regel gemeinsam mit einer entsprechenden Wissensbasis, die als Fakten e/2 bereitstellt, nun automatisiert von einem Programm, das nach dem "Beweisermodell" arbeitet, auswerten lassen können.[4]

Eine beispielhafte Implementierung dieser Regel in Prolog sieht folgendermaßen aus:

```
geschwister(X, Y) :- elternteil(E, X), elternteil(E, Y), X \== Y.
```

[3] Im normalen Sprachgebrauch geht man davon aus, dass zwei unterschiedliche Personen gemeint sind
[4] Dies wird in Aufgabe 6 gezeigt

3. Bruder / Schwester

> *“Definieren Sie unter Verwendung des Prädikates geschwister(X,Y) die beiden Prädikate bruder(X,Y) und schwester(X,Y).”*

Bruder

Jede Person B, die männlich und ein Geschwisterteil der Person P ist, ist Bruder von P.

∀ B, P : [bruder(B, P) ← maennlich(B) ∧ geschwister(B, P)]

Schwester

Jede Person S, die weiblich und ein Geschwisterteil der Person P ist, ist Schwester von P.

∀ S, P : [schwester(S, P) ← weiblich(S) ∧ geschwister(S, P)]

4. Eheleute

> *“Definieren Sie unter Verwendung des Prädikates verheiratet(X,Y) das symmetrische Prädikat eheleute(X,Y).”*

Das Prädikat verheiratet(X,Y) trifft nur die Aussage, dass X mit Y verheiratet ist.
Im normalen Sprachgebrauch impliziert die Aussage “X ist mit Y verheiratet”, dass Y ebenfalls mit X verheiratet ist.

Eine Lösungsmöglichkeit hierfür wäre, in der Wissensdatenbank für jedes verheiratete Paar jeweils zwei Fakten zu hinterlegen.

Das Prädikat verheiratet(X, Y) lässt sich auch lesen wie der Satz “X und Y sind miteinander verheiratet.”, auf welchen sich die Regeln der (natürlichsprachigen) Grammatik anwenden lassen.

“Es geht hier vielmehr um Vorgänge/Zustände, in die die beiden Aktanten in gleicher Weise involviert sind; sie können daher jeweils einmal als Subjekt und einmal als Objekt des Verbums gesehen werden, also: Bert [...] gleicht Klausi und Klausi [...] gleicht Bert.”[5]

In Anlehnung an die Aussage, dass “verbales Argumentieren ein ungeprüftes Verfahren” und “im Allgemeinen nicht gut geeignet, besonders [...] für Neulinge auf dem Gebiet absolut ungeeignet”[6] sei, lässt sich alternativ und einfacher ein zusätzliches Prädikat eheleute(X, Y) definieren, welches die Reihenfolge der Parameter nebensächlich macht, indem es für beide Personen abfragt, ob X mit Y verheiratet ist oder Y mit X.

---

[5] U. Wandruszka: Syntax und Morphosyntax, Eine kategorialgrammatische Darstellung anhand romanischer und deutscher Fakten, 1. Auflage, Gunter Narr Verlag, Tübingen, 1997
[6] C. Klauck: Logik, 4. Auflage 2014, Vorlesungsskript, HAW-Hamburg

5. Uropa / Uroma

> *"Definieren Sie unter Verwendung von Verwandtschaftsstufen (oma,opa) die Prädikat uropa(X,Y) und uroma(X,Y)."*

Oma / Opa

Eine Oma (/ Opa) ist der weibliche (/ männliche) Elternteil eines Elternteils einer Person.
Jede Person (mit mindestens zwei vorhergehenden Generationen[7]) wird dementsprechend zwei Omas (/ Opas) haben.

Formal definiert ist eine Person O dann eine Oma der Person P, wenn sie weiblich ist und es eine Person E gibt, dessen Elternteil O ist, wobei E aber selbst Elternteil von P ist.

$$\forall\ O, P : [\ oma(O, P) \leftarrow (\ weiblich(O) \land \exists\ E: [\ (\ elternteil(O, E) \land elternteil(E, P)\ )\ ]\ )\ ]$$

Diese Formel lässt sich wie in Aufgabe 2 umformen und anschließend mit einem Programm, das nach dem Beweisermodell arbeitet, beweisen.

Uroma / Uropa

Uromas (/ Uropas) sind also die weiblichen (/ männlichen) Elternteile der Omas **und** Opas einer Person.
Jede Person (mit mindestens drei vorhergehenden Generationen) wird also vier Uromas (/ Uropas) haben.
Jede Person U ist Uroma von P, wenn sie weiblich und Elternteil einer Oma oder eines Opas der Person P ist.

$$
\begin{array}{l}
\forall\ U, P : [\\
\quad uroma(U, P) \leftarrow (\\
\quad\quad weiblich(U) \land \\
\quad\quad \exists\ Oma, Opa : [\ (\\
\quad\quad\quad (\ elternteil(U, Oma) \lor elternteil(U, Opa)\ ) \\
\quad\quad\quad \land\ (\ oma(Oma, P) \lor opa(Opa, P) \\
\quad\quad\quad)\ ] \\
\quad\quad) \\
\quad]
\end{array}
$$

Auch diese Formel lässt sich wie in Aufgabe 2 umformen.

Mutter / Vater

Zusätzlich würden sich hier die zwei weiteren Beziehungen "Mutter" und "Vater" anbieten, mit deren Hilfe sich Uromas (/ Uropas) zu den Müttern (/ Vätern) der Omas und Opas einer Person vereinfachen lassen.

---

[7] Einschränkung hier durch den Umfang des vorgegebenen Stammbaumes

6. Konsistenzprüfung

> *“Definieren Sie Prädikate zur Konsistenzprüfung der Datenbank (maenUweibl/1, verhKor/1, elterVoll/1, wurzel/1) und geben Sie ggf. die Fehleinträge als Liste zurück. In diesen Listen sollen keine Mehrfacheinträge vorkommen. Sie können das Prädikat findall/3 verwenden.”*

Allgemein

Alle Konsistenzprüfungsprädikate nutzen das Prädikat findall/3, welches eine Liste aller Belegungen (Substitutionen) von Variablen erstellt, die eine gegebene Anfrage erfolgreich auswerten lassen.
Für die Auswertung der Anfragen ist im ersten Argument des Prädikats zu definieren, welche Variablen genutzt werden sollen (∃ Var : [Anfrage]) und im zweiten Argument die Anfrage selbst.
Das dritte Argument ist die Ausgabeliste.

Anfragen (auch Zielklauseln / engl. Goal genannt) können formuliert werden als “stimmt es, dass … ?”, wobei anstelle der “…” die Frage gestellt wird.
Um die Erfüllbarkeit dieser Anfragen zu beweisen, kann man nach dem Resolutionsverfahren vorgehen:
Zuerst wird behauptet, die Anfrage sei unter allen Umständen (Belegungen) wahr (Tautologie).
Da mit dem Resolutionsverfahren aber nur Widersprüche bewiesen werden können, wird diese Behauptung zu einem Widerspruch negiert.
Gemeinsam mit den vorher definierten Fakten und Regeln (in allquantifizierter Form, siehe Aufgabe 2) kann der Widerspruch bewiesen werden, was im Umkehrschluss bedeutet, dass die Tautologie der Behauptung bewiesen wurde.

Beispielhaft wird dies einmal komplett an der Aufgabe 6.1 detailliert erklärt,
für die Aufgaben 6.2, 6.3 und 6.4 werden die gegebenen Anfragen als Behauptungen formuliert und in eine Form gebracht, die ein Programm, das automatisch nach der Beschreibung in Aufgabe 6.1 verarbeiten kann.

Werden durch das Backtracking Anfragen mehrfach mit der gleichen Substitution der Variablen erfolgreich ausgewertet, so wird diese Belegung auch mehrmals in der Liste auftauchen.
Vor Rückgabe der Fehlerliste muss ein Weg gefunden werden, die generierte Liste von Mehrfacheinträgen zu bereinigen.
Ein möglicher Weg ist, die Originalliste komplett zu durchlaufen und parallel eine zweite Liste nur dann mit Elementen zu füllen, wenn diese Elemente noch nicht Teil der zweiten Liste sind.

Ein Element ist nicht Teil einer Liste, wenn das Element nicht gleich dem obersten Element der Liste ist und auch nicht Teil des Rests der Liste ist.

6.1 maenUweibl/1

“Gibt es eine Person, die zugleich männlich und weiblich ist?”

Als Anfrage formuliert:

“Stimmt es dass, es eine Person X gibt, die zugleich männlich und weiblich ist?”

Als Behauptung:

“Es gibt eine Person X, die zugleich männlich und weiblich ist”

$\exists\ X : [\ m(X) \land w(X)\ ]$

Als Negation der Behauptung:

$\neg \exists\ X : [\ m(X) \land w(X)\ ]$

$\equiv$ $\forall\ X : [\ \neg\ (\ m(X) \land w(X)\ )\ ]$

“Alle Personen X sind nicht männlich und weiblich zugleich.”

KNF: $\neg\ (\ m(X) \land w(X)\ ) \equiv \neg m(X) \lor \neg w(X)$

Resolution

Darstellung in Mengenschreibweise:

{ Fakten, Regeln, negierte_Behauptung }

Fakten:

$\{m(Y)\}_1$, $\{m(A)\}_2$, $\{m(B)\}_3$, $\{m(X)\}_4$, …

$\{w(H)\}_5$, $\{w(K)\}_6$, $\{w(L)\}_7$, $\{w(X)\}_8$, …

Regeln:

(hier nicht relevant)

Negierte Behauptung:

$\{\ \neg m(X), \neg w(X)\ \}_{Behauptung}$

Resolution:

Behauptung mit K_4 mit $m(X) = \{\ \neg w(X)\ \}_1$

R_1 mit K_8 mit $w(X) = \varnothing$

Der Widerspruch wurde bewiesen, im Umkehrschluss wurde somit die Allgemeingültigkeit der Behauptung für X bewiesen.

6.2 verhKor/1

“Ist das Prädikat verheiratet korrekt abgespeichert? (links Mann, rechts Frau).”

Als Behauptung:
“Es gibt eine Person L, die an linker Stelle des Prädikats ‘verheiratet’ steht und nicht männlich ist oder eine Person R, die an rechter Stelle des Prädikats steht und nicht weiblich ist.”

$$\exists \ L, R : [\text{ verheiratet}(L, R) \land (\text{ weiblich}(L) \lor \text{ männlich}(R))]$$

Negation und Resolution gemäß Aufgabe 6.1.

6.3 elterVoll/1

“Sind alle Personen im Prädikat elternteil auch als männlich oder weiblich registriert?”

“Nicht männlich oder weiblich” ≡ “Nicht männlich und nicht weiblich”
$\neg(m \lor w) \equiv \neg m \land \neg w$

Als Behauptung:
“Es gibt eine Person E, die im Prädikat ‘elternteil’ genannt wird, aber nicht männlich und nicht weiblich ist, oder eine Person K, die im Prädikat ‘elternteil’ genannt wird, aber nicht männlich und nicht weiblich ist.”

$$\exists \ E, K : [\text{ elternteil}(E, K) \land ((\neg m(E) \land \neg w(E)) \lor (\neg m(K) \land \neg w(K))]$$

Negation und Resolution gemäß Aufgabe 6.1.

6.4 wurzel/1

“Gibt es jemand ohne Eltern?”

Um die Belegungen für “jemand” zu finden, nehmen wir alle uns bekannten Personen, indem wir alle weiblichen und anschließend alle männlichen Personen überprüfen.

Behauptung:
“Es gibt eine Person J, die weiblich oder männlich ist, und nicht Kind von jemanden ist.”

$$\exists \ J : [(\text{ weiblich}(J) \lor \text{ männlich}(J)) \land \neg \text{elternteil}(\_ , J)]$$

Negation und Resolution gemäß Aufgabe 6.1.

7. Exkurs: Vergleich von Implementierungen

Da für Aufgabe 5 für die Beziehung der Uroma (/ Uropa) irrelevant ist, von welchem Geschlecht die Zwischen-Generationen sind, wäre eine (wie in der Aufgabenstellung geforderte) Anfrage mit unnötig viel Rechenaufwand verbunden:

1. Uroma als weiblichen Vorfahren der Omas UND Opas einer Person

Unter Berücksichtigung der Anzahl der Aufrufe von (Unter-) Prädikaten ist es sinnvoller, eine solche Abfrage zu nutzen:

2. Uroma als *Mutter* eines Elternteils eines Elternteils der Person

Ein Vergleich mit time/1 (welches nicht die Anzahl der Schritte im Vierportmodell angibt, sondern die Anzahl der Inferenzen der durch Prolog implementierten Inferenzmaschine[8]) zeigt hier deutliche Unterschiede:
Ein zum Ausprobieren implementierter direkter Vergleich mit "eve" als Uroma von "alois" benötigte mit Implementierung (1) bis zu 50 Schritte für den Nachweis und mit Implementierung (2) maximal 16.

Dies ist ein kleines, aber wie ich finde interessantes, Beispiel dafür, wie man als Software-Entwickler mit wenig Vorüberlegungen Laufzeit-effizienter arbeiten kann. (Zumindest in Systemen, die ausreichend komplex sind, damit ein Unterschied spürbar ist.)

[8] http://www.swi-prolog.org/pldoc/man?predicate=time/1, abgerufen Nov. 2016

8. Modellierungsaufgabe

Aufgabe

Für die Informatikstudierenden Anton, Berta, Charlie und Dora steht demnächst die Klausur „Mathematische Grundlagen“ an. Eigentlich wollen sie die ins nächste Semester verschieben. Sie beschließen aber, dass mindestens einer von ihnen an dieser Klausur teilnehmen sollte, damit sie wissen, was drankommt.
Charlie weigert sich grundsätzlich, zu dieser Klausur zu gehen.
Da Anton und Dora auf ihre gemeinsame kleine Schwester aufpassen müssen, kann höchstens einer von beiden zur Klausur gehen.
Dora geht nur zur Klausur, wenn auch ihre beste Freundin Esmeralda hingeht.
Wenn Esmeralda an der Klausur teilnimmt, dann auch Anton.

a. *Stellen Sie eine aussagenlogische Formel auf, die die obige Situation bzgl. Teilnahme an dieser Klausur beschreibt.*
b. *Angenommen, Berta wird krank und kann an dieser Klausur nicht teilnehmen.*
   Zeigen Sie mittels Resolutions-Kalkül, dass dann Anton zur Klausur geht und Dora nicht.

Lösung

Modellierung

Aussagenlogische Variablen

A = “Anton geht zu der Klausur”
B = “Berta geht zu der Klausur”
C = “Charlie geht zu der Klausur”
D = “Dora geht zu der Klausur”
E = “Esmeralda geht zu der Klausur”

Aussagenlogische Formeln

- “[...] mindestens einer von ihnen an dieser Klausur teilnehmen sollte, [...]”
  $F1 = (A \lor B \lor C \lor D)$
- “Charlie weigert sich [...]”
  $F2 = \neg C$
- “Da Anton und Dora auf ihre gemeinsame kleine Schwester aufpassen müssen, kann höchstens einer von beiden zur Klausur gehen.”
  = Anton und Dora können nicht gleichzeitig gehen
  $F3 = \neg(A \land D) \equiv \neg A \lor \neg D$
- “Dora geht <u>nur</u> zur Klausur, <u>wenn</u> auch ihre beste Freundin Esmeralda hingeht.”
  $F4 = D \rightarrow E \equiv \neg D \lor E$
- “<u>Wenn</u> Esmeralda an der Klausur teilnimmt, <u>dann</u> auch Anton”
  $F5 = E \rightarrow A \equiv \neg E \lor A$

- Zusammenfassung der Fakten
  $F_G = F1 \land F2 \land F3 \land F4 \land F5$

Behauptung

- Annahme: Berta wird Krank
  $F6 = \neg B$
- Behauptung: “[...] Anton zur Klausur geht und Dora nicht.”
  $F_B = A \land \neg D$

Zu zeigen ist, dass aus den Fakten (F_G) und der neuen Situation (Krankheit von Berta = F6) folgt, dass die Behauptung F_B unter allen Umständen wahr ist (Tautologie).

$F_G \land F6 \rightarrow F_B \equiv$ Tautologie

Resolution

Da sich mit dem Resolutionskalkül nur Widersprüche beweisen lassen, negieren wir die Tautologie zu einem Widerspruch und beweisen diesen.

$\neg((F_G \land F6) \rightarrow F_B) \equiv$ Widerspruch

$\equiv \quad \neg(\neg(F_G \land F6) \lor F_B)$

$\equiv \quad F_G \land F6 \land \neg F_B$

Darstellung in Mengenschreibweise

Besondere Achtung bei der Negation der Anfrage:

$\neg F_B = \neg(A \land \neg D) = (\neg A \lor D)$

$\{ \{A, B, C, D\}_1, \{\neg C\}_2, \{\neg A, \neg D\}_3, \{\neg D, E\}_4, \{\neg E, A\}_5, \{\neg B\}_6, \{\neg A, D\}_{Behauptung} \}$

Resolution:

$K_{Behauptung}$	mit	K_3	mit	D	=	$\{ \neg A1 \}_1$
R_1	mit	K_5	mit	A	=	$\{ \neg E2 \}_2$
R_2	mit	K_4	mit	E	=	$\{ \neg D \}_3$
R_3	mit	K_B	mit	D	=	$\{ \neg A \}_4$
R_4	mit	K_1	mit	A	=	$\{ B, C, D \}_5$
R_5	mit	K_4	mit	D	=	$\{ B, C, E \}_6$
R_6	mit	K_6	mit	B	=	$\{ C, E \}_7$
R_7	mit	K_2	mit	C	=	$\{ E \}_8$
R_8	mit	K_3	mit	E	=	$\varnothing$

Durch das Erreichen der leeren Menge wurde der Widerspruch der negierten Tautologie bewiesen. Die Behauptung dass Anton zur Klausur geht und Dora nicht, stimmt also unter der Voraussetzung, dass Berta krank wird.

Quellenangaben

- C. Klauck: Logik, 4. Auflage 2014, Vorlesungsskript, HAW-Hamburg
- W.F. Clocksin / C.S. Mellish: Programming in Prolog, 5. Auflage, Axel Springer Verlag, Berlin, Heidelberg 2003
- H. Göhner / B. Hafenbrak: Arbeitsbuch Prolog, 2. Auflage, Dümmler Verlag, Bonn, 1991
- J. Richter-Gebert, Skript zur Vorlesung "LOGIK", ETH Zürich http://www-hm.ma.tum.de/archiv/in2/ss02/literatur.html
- U. Wandruszka: Syntax und Morphosyntax, Eine kategorialgrammatische Darstellung anhand romanischer und deutscher Fakten, 1. Auflage, Gunter Narr Verlag, Tübingen, 1997

Erklärung zur schriftlichen Ausarbeitung des Referates

Hiermit erkläre ich, dass ich diese schriftliche Ausarbeitung meines Referates selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe sowie die aus fremden Quellen (dazu zählen auch Internetquellen) direkt oder indirekt übernommenen Gedanken oder Wortlaute als solche kenntlich gemacht habe. Die Arbeit habe ich bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher nicht veröffentlicht.