Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

12.04.2018

Logische Programmierung & deduktive Datenbanken — Übungsblatt 2 (Vergleich mit SQL) —

Ihre Lösungen zu den Hausaufgaben e) bis i) schicken Sie bitte per EMail an den jeweiligen Übungsleiter, also `mario.wenzel@informatik.uni-halle.de` für die Dienstags-Gruppe und `brass@informatik.uni-halle.de` für die Donnerstags-Gruppe. Bitte schreiben Sie "`lp18`" mit in die Betreff-Zeile. Einsendeschluss ist Samstag, der 21. April (die Aufgaben werden in den Übungen am 24. April und 26. April besprochen). Aufgabe b) wird dort auch besprochen: Sie sollten vorher darüber nachdenken, müssen aber nichts abgeben.

Zum Selbstudium

a) Schauen Sie sich die folgenden Webseiten an. Sie brauchen für diese Aufgabe nichts abzugeben. Ziel ist, dass Sie einen Eindruck davon gewinnen, was es im WWW zum Thema dieser Vorlesung gibt, und dabei für sich nützliche Quellen entdecken.

- "Try Logic Programming! A Gentle Introduction to Prolog" von Bernardo Pires:

  [https://bernardopires.com/2013/10/]
  [try-logic-programming-a-gentle-introduction-to-prolog/]

  Versuchen Sie das Programm zum Färben der Deutschlandkarte zu verstehen: Es sollen dabei keine zwei benachbarten Bundesländer die gleiche Farbe zugewiesen bekommen. Sie können sich den Prolog-Code auch mit Copy&Paste in eine eigene Datei kopieren. (Eigentlich sind "Constraint Logic Programming" Systeme auf solche Aufgaben spezialisiert, aber kleinere Beispiele wie dieses kann auch ein klassisches Prolog-System lösen.)

- "Overview of Prolog Systems" von Ulrich Neumerkel:

  [http://www.complang.tuwien.ac.at/ulrich/prolog_misc/systems.html].

  Die Liste ist vermutlich ungefähr von 2004, obwohl die HTML Datei zuletzt 2015 geändert wurde.

- Auch die Wikipedia hat eine Liste von Prolog-Implementierungen (aktueller, aber mit etwas anderen Informationen):

  [https://en.wikipedia.org/wiki/Comparison_of_Prolog_implementations].

- Im PC-Pool sind SWI-Prolog und GNU Prolog installiert. Da Sie SWI-Prolog schon ausprobiert haben, informieren Sie sich etwas über GNU Prolog:

  [http://www.gprolog.org/]

  Z.B. könnten Sie sich einen ersten Eindruck vom Handbuch verschaffen.

  [http://www.gprolog.org/manual/html_node/index.html]

  Die Benutzerschnittstelle des Interpreters wird hier erklärt:

  [http://www.gprolog.org/manual/html_node/gprolog007.html#sec9]

- Hier zum Vergleich das Handbuch von SWI-Prolog:

  [http://www.swi-prolog.org/pldoc/doc_for?object=manual]

b) Was würden Sie in einer mündlichen Prüfung auf die folgenden Fragen antworten?

- Schreiben Sie das Programm zur Vorfahr-Berechnung auf. Allgemein berechnet dies die transitive Hülle einer zweistelligen Relation, in diesem Fall der "Elternteil"-Relation.
- Wo wird noch eine transitive Hülle gebraucht?
- Kann man die transitive Hülle auch in relationaler Algebra berechnen? Wie ist die Situation bei SQL?
- In der Vorlesung wurde gesagt, dass Prolog "Top-Down" vorgeht, deduktive Datenbanken dagegen "Bottom-Up". Was muss man sich dazu "oben" und "unten" vorstellen? Wie funktionieren die beiden Systeme also ganz grob gesprochen? (Die Details füllen später einen recht großen Teil der Vorlesung.)
- Kann Prolog in Endlosschleifen geraten? Geben Sie ggf. ein Beispiel.
- Wie läd man eine Datei mit Fakten und Regeln in Prolog?
- Wie verhält sich Prolog, wenn man eine Anfrage stellt, die mehr als eine Antwort hat?
- Wie beendet man eine Sitzung des Prolog-Interpreters?
- Wie unterscheiden sich in Prolog Variablen und Konstanten? Wann kann man die Anführungszeichen `'...'` weglassen?
- Wie sieht die anonyme Variable aus? Was unterscheidet sie von normalen Variablen? Welchen Schutz gibt es in Prolog gegen Tippfehler in Variablen?
- Welche Probleme gibt es an der Schnittstelle zwischen Programmiersprache und klassischer SQL-Datenbank? Welche Versuche gibt es, Datenbank und Programmiersprache zu integrieren? Was zeichnet in diesem Zusammenhang deduktive Datenbanken aus?

Präsenzaufgaben

c) Laden Sie sich die folgende Datei mit den klassischen Beispiel-Relationen von Oracle herunter (hier in einer etwas vereinfachten Version):

[http://users.informatik.uni-halle.de/~brass/lp18/empdept.pl]

Diese Datei enthält Fakten zu Abteilungen einer Firma ("departments") und Angestellten ("employees"). Die Tabelle `dept` hat folgende Spalten:

- `DEPTNO` (Department Number): Nummer der Abteilung
- `DNAME` (Department Name): Name der Abteilung
- `LOC` (Location): Sitz der Abteilung.

dept		
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Die Tabelle `emp` hat folgende Spalten:

- `EMPNO` ("employee number"): Angestellten-Nummer (Schlüssel, identifiziert Angestellten eindeutig).
- `ENAME` ("employee name"): Name des Angestellten.
- `JOB`: Beruf des Angestellten.
- `MGR` ("manager"): Angestellten-Nummer des direkten Vorgesetzten.
- `SAL` ("salary"): Gehalt des Angestellten.
- `DEPTNO` ("department number"): Abteilung, in der der Angestellte arbeitet.

emp					
EMPNO	ENAME	JOB	MGR	SAL	DEPTNO
7369	SMITH	CLERK	7902	800	20
7499	ALLEN	SALESMAN	7698	1600	30
7521	WARD	SALESMAN	7698	1250	30
7566	JONES	MANAGER	7839	2975	20
7654	MARTIN	SALESMAN	7698	1250	30
7698	BLAKE	MANAGER	7839	2850	30
7782	CLARK	MANAGER	7839	2450	10
7788	SCOTT	ANALYST	7566	3000	20
7839	KING	PRESIDENT		5000	10
7844	TURNER	SALESMAN	7698	1500	30
7876	ADAMS	CLERK	7788	1100	20
7900	JAMES	CLERK	7698	950	30
7902	FORD	ANALYST	7566	3000	20
7934	MILLER	CLERK	7782	1300	10

Formulieren Sie die folgenden Anfragen sowohl in SQL wie in Prolog:

- Gesucht sind Nummer und Name der Abteilung in Boston.
- Geben Sie Nummer und Name aller Angestellten in der Forschungsabteilung (Name: "RESEARCH') aus.
- Erstellen Sie eine Liste aller Angestellten (Nummer und Name), die "MANAGER" oder "PRESIDENT" der Firma sind.
- Geben Sie alle Angestellten aus, die mehr als ihr direkter Vorgesetzte verdienen.

  Sie können in Prolog im Regelrumpf auch Bedingungen wie `X > Y` verwenden. Die Variablen müssen in einer Bedingung weiter links in der Regel an einen Wert gebunden werden. Probieren Sie aus, welche Fehlermeldung Sie erhalten, wenn Sie die Rumpfliterale so umsortieren, dass mindestens eine der am Vergleich beteiligten Variablen noch keinen Wert hat, wenn der Vergleich ausgeführt wird. Beim "Constraint Logic Programming" können Bedingungen automatisch auf einen passenden späteren Zeitpunkt verschoben werden. Prolog wertet die Bedingungen dagegen strikt in der gegebenen Reihenfolge aus.
- Welche Angestellten sind direkte oder indirekte Untergebene von "JONES"? (Für die SQL-Version können Sie voraussetzen, dass es nur zwei Hierarchie-Ebenen unterhalb von "JONES" gibt.)

d) Vergleichen Sie SQL und Prolog/Datalog anhand der Beispiele aus c). Wenn Sie sich die Reihenfolge der Spalten von `emp` nicht merken können, wie könnten Ihnen Hilfsprädikate helfen?

Wenn Sie eine eigene logische Programmiersprache entwerfen würden, wie könnten Sie die Vorteile von Tupelkalkül (mit Spalten-Namen) und Bereichskalkül (mit Argument-Positionen) verbinden?

Hausaufgabe

Gegeben sei eine Datenbank mit Informationen über Komponisten, Stücke und CDs:

- `komponist(KNR, NAME, VORNAME, GEBOREN, GESTORBEN).`
- `stueck(SNR, KNR→komponist, TITEL, TONART, OPUS).`
  (`TONART` und `OPUS` können `NULL` sein, dargestellt als `'?'`.)
- `cd(CDNR, NAME, HERSTELLER, ANZ_CDs, GESAMTSPIELZEIT).`
- `aufnahme(CDNR→cd, SNR→stueck, ORCHESTER, LEITUNG).`
  (`ORCHESTER` und `LEITUNG` können `NULL` sein, dargestellt als `'?'`.)
- `solist((CDNR, SNR)→aufnahme, NAME, INSTRUMENT).`

Sie finden die Daten unter

[http://www.informatik.uni-halle.de/˜brass/lp18/cd.pl]

Formulieren Sie die folgenden Anfragen in Prolog, indem Sie passende Prädikate definieren, z.B. `h2e` für Teil e) auf diesem zweiten Übungsblatt. Wenn Ihnen bessere Namen für die Prädikate einfallen, dürfen Sie die natürlich auch verwenden. Außerdem können Sie bei Bedarf beliebige Hilfsprädikate einführen.

e) Was sind Geburts- und Todesjahr von Wolfgang Amadeus Mozart?

f) Geben Sie Vorname und Nachname aller Komponisten aus, die zwischen 1700 und 1799 geboren sind (jeweils einschliesslich der Grenzen).

   Vergleiche werden in Prolog `<`, `=<`, `>=`, `>` geschrieben. Beachten Sie, dass `=<` (für $\leq$) anders ist als in den meisten anderen Programmiersprachen. Die Motivation dahinter ist, dass man den Pfeil `<=` vermeiden wollte.

g) Geben Sie Name, Vorname und Alter aller Komponisten aus, d.h. jeweils die Differenz von Todesjahr und Geburtsjahr.

   Arithmetische Ausdrücke können in Prolog mit dem Prädikat `is` ausgewertet werden, z.B. `X is Y-Z`. Schreibt man dagegen `X = Y-Z`, so ist `X` anschließend ein Term wie etwa `1759-1685` für Georg Friedrich Händel.

h) Welcher Komponist (oder welche Komponisten) sind am ältesten geworden, d.h. die Differenz von Todesjahr und Geburtsjahr ist maximal? Gesucht sind Name, Vorname und das so berechnete Alter.

   Definieren Sie ein Hilfsprädikat für die Nicht-Ältesten Komponisten und verwenden Sie dann die Negation `\+`.

i) Geben Sie alle Stücke in C-dur oder a-moll aus. Gesucht sind jeweils der Nachname des Komponisten und der Titel des Stückes.