1. Übung zur Vorlesung "Objektorientierte und Deduktive Datenbanken" im Wintersemester 2004 – mit Musterlösungen –

Prof. Dr. Gerd Stumme, Dipl.-Inform. Christoph Schmitz

1. November 2004

Aufgabe 1

Aus der Philosophenuni kennen Sie die Relation *voraussetzen(Vorgänger, Nachfolger)*, die die direkten Voraussetzungen von Vorlesungen beschreibt. Gesucht ist nun die transitive Hülle dieser Relation, d.h. auch die Vorgänger der Vorgänger, die Vorgänger der Vorgänger der Vorgänger usw.

In relationaler Algebra läßt sich diese transitive Hülle allerdings nicht beschreiben, wie Sie aus der Vorlesung wissen.

- Geben Sie einen Ausdruck in relationaler Algebra an, der wenigstens die Vorvorgänger jeder Vorlesung zurückgibt.

$$\pi_{v_1.Vorgaenger,v_2.Nachfolger}(\rho_{v_1}(voraussetzen) \underset{Nachfolger=Vorgaenger}{\bowtie} \rho_{v_2}(voraussetzen))$$

- Jemand weist Sie darauf hin, daß es nie mehr als 20 verschiedene Vorlesungen an der Philosophenuni gibt.

  Können Sie nun einen Ausdruck in relationaler Algebra angeben, der die transitive Hülle der Relation *voraussetzen* berechnet?

  Im Folgenden wird die Relation *voraussetzen(Vorgänger, Nachfolger)* mit $v(V, N)$ abgekürzt.

$$\begin{aligned} v_{Huelle} &= v \\ &\cup \pi_{v_1.V,v_2.N}(\rho_{v_1}(v) \underset{N=V}{\bowtie} \rho_{v_2}(v)) \\ &\cup \ldots \\ &\cup \pi_{v_1.V,v_{20}.N}(\rho_{v_1}(v) \underset{N=V}{\bowtie} \ldots \underset{N=V}{\bowtie} \rho_{v_{20}}(v)) \end{aligned}$$

Aufgabe 2: Relationale Algebra und Datalog

Beantworten Sie folgende Fragen an die Philosophenuni mit Hilfe von relationaler Algebra. Setzen Sie dann die Algebraausdrücke in Datalog um.

- Welche Vorlesungen werden überhaupt gehört?

  $\pi_{vorlnr}(Hoeren)$

  wirdgehört(Vorlesung) :- hören(Student, Vorlesung).

- Welche Vorlesungen (Titel ist gesucht) hält Professor 2125?

  $\pi_{Titel}(\sigma_{persnr=2125}(Vorlesungen))$

  gesucht(Titel) :- vorlesungen(Nr, Titel, SWS, 2125).

- Welche Vorlesungen (Titel) hält Sokrates?

  $\pi_{Titel}(\sigma_{name='Sokrates'}(Professoren) \bowtie Vorlesungen)$

  gesucht(Titel) :- professoren(Persnr, "Sokrates", Rang, Raum), vorlesungen(Nr, Titel, SWS, Persnr).

Aufgabe 3: Datalog

Betrachten Sie wieder die Philosophenuni.

- Geben Sie eine Datalog-Regel an, die berechnet, welche Studenten Kommilitonen sind. Kommilitonen sind diejenigen Studenten, die zusammen eine Vorlesung gehört haben.

  kommilitone(S1, S2) :- hören(S1, V), hören(S2, V), S1!=S2.

- Wenn ein Student ein Problem hat, fragt er Kommilitonen, ob diese ihrerseits eine Lösung wissen oder Kommilitonen haben, die weiterhelfen können. Formulieren Sie diese Relation *kannFragen* einmal unter Verwendung der Relation *kommilitone* aus der vorigen Aufgabe, einmal direkt unter Verwendung von *hören*.

  Variante 1:

  kannFragen(S1, S2) :- kommilitone(S1, S2).

  kannFragen(S1, S2) :- kommilitone(S1, S3), kommilitone(S3, S2), S1!=S2.

  Variante 2:

  kannFragen(S1, S2) :- hören(S1, V), hören(S2, V), S1!=S2.

  kannFragen(S1, S2) :- hören(S1, V1), hören(S3, V1), hören(S3, V2), hören(S2, V2), S1!=S3, S2!=S3, S1!=S3.

- Carnap hat ein Problem. Wen kann er fragen? Geben Sie den Rechenweg an.

  kannFragen(carnap, theophrastos) wegen hören(carnap, ethik), hören(theophrastos, ethik)

  kannFragen(carnap, schopenhauer) wegen hören(carnap, ethik), hören(theophrastos, ethik), hören(theophrastos, grundzüge), hören(schopenhauer, grundzüge)

  kannFragen(carnap, fichte) wegen hören(carnap, ethik), hören(theophrastos, ethik), hören(theophrastos, grundzüge), hören(fichte, grundzüge)

- Ein Student will Vorlesungen hören, die keine Voraussetzungen haben. Sie sollen nicht von C3-Professoren gehalten werden. Geben Sie einen Datalog-Ausdruck an, der die Titel dieser Vorlesungen zurückgibt.

  willHoeren(Titel) :- vorlesung(Nr, Titel, SWS, PersNr), ¬voraussetzen(VorNr, Nr), ¬professoren(PersNr, Name, c3, Raum).

- Man stellt fest, daß die Uni doch eine kleine Welt ist: man kann eigentlich jeden kennenlernen, der über eine Kette von "X kennt jemanden, der jemanden kennt, ..., der jemanden kennt, der Y kennt" verknüpft ist. Geben sie eine Regel *kannKennenlernen* basierend auf *hören* an.

  kannKennenlernen(P1,P2):-hören(P1,V),hören(P2,V).

  kannKennenlernen(P1,P2):-hören(P1,V),hören(P3,V),kannKennenlernen(P3,P2).