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

06.06.2014

Logische Programmierung & deduktive Datenbanken — Hausaufgabe 1 —

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/lp14/cd.pro`

Formulieren Sie die folgenden Anfragen in Prolog, indem Sie passende Prädikate definieren (s.u.). Sie können natürlich bei Bedarf beliebige Hilfsprädikate einführen.

a) Geben Sie alle Stücke in C-dur oder a-moll aus. Gesucht sind jeweils der Nachname des Komponisten und der Titel des Stückes. Schreiben Sie bitte zwei Prädikate: `h1a/2` mit Argumenten für Komponistname und Stücktitel, und `h1a/0` ohne Argumente, das alle Antworten ausgibt (eine Zeile pro Antwort) und am Ende erfolgreich zurückkehrt.

b) Welche CDs (Nummer und Name) enthalten Stücke von mindestens zwei verschiedenen Komponisten? Definieren Sie ein Prädikat `h1b/2` mit Argumenten für die gesuchten Daten. Definieren Sie ausserdem ein Prädikat `h1b/1`, das eine Liste (ohne Duplikate) von Termen der Form `cd(Nr,Name)` mit den gesuchten Daten liefert.

   Sie können `\=` zum Test auf Ungleichheit verwenden. Ggf. ist `setof(A, B, C)` hilfreich, das `C` auf die Liste aller `A`s setzt, für die `B` gilt, also `C` = {`A` | `B`}. Sie dürfen auch `once(G)` verwenden, was nur eine Lösung für das Beweisziel `G` berechnet. Wenn unbedingt nötig, dürfen Sie auch einen Cut verwenden.

c) Welcher Komponist (oder welche Komponisten) sind am ältesten geworden, d.h. die Differenz von Todesjahr und Geburtsjahr ist maximal? Schreiben Sie ein Prädikat `h1c/3` mit Argumenten für Name, Vorname, und das so berechnete Alter.

   Wie bekannt können Sie arithmetische Ausdrücke mit dem Prädikat `is` auswerten, also z.B. `X is Y-Z`. Definieren Sie ein Hilfsprädikat für die Nicht-Ältesten Komponisten und verwenden Sie dann die Negation `\+`.

d) Definieren Sie ein Prädikat `h1d/2`, das als erstes Argument einen Suchstring wie z.B. `'M%a_t'` nimmt, und dann im zweiten Argument jeweils einen Nachnamen eines Komponisten liefert, der die entsprechende `LIKE`-Bedingung aus SQL erfüllen würde, also z.B. `Mozart`. Das Prozentzeichen soll auf eine beliebige Folge beliebiger Zeichen passen, der Unterstrich auf ein einzelnes beliebiges Zeichen. Alle anderen Zeichen müssen genau passen.

   Sie können das Prädikat `name/2` verwenden, dass ein Atom in eine Liste von ASCII-Codes umwandelt. Das Prozentzeichen `%` hat den Code `37`, der Unterstrich `_` den Code `95`. Sie können auch `0'%` und `0'_` für die ASCII-Codes dieser Zeichen schreiben.

Termin: Diese Aufgaben werden in der Übung am 12.06.2014 besprochen.