Prof. Dr. Stefan Brass
Institut für Informatik
Universität Halle

19.06.2006

Deduktive Datenbanken und Logische Programmierung — Blatt 7: Typprüfer, Fortsetzung —

Aufgabe 9 — 6 Punkte

Erweitern Sie Ihren Typprüfer für Prolog/Datalog (von Blatt 5) so, daß er das zu prüfende Programm aus einer Datei liest. Man soll ihn als Prolog-Prädikat mit dem Dateinamen aufrufen können, z.B.

```
type_check('bsp.pro').
```

Die Prädikat-Deklarationen stehen jetzt in dem zu prüfenden Programm selbst (vor der Verwendung des jeweiligen Prädikates, aber nicht notwendigerweise ganz am Anfang: Sie können auch zwischen Regeln stehen). Bitte verwenden Sie eine Operator-Deklaration, die folgende Syntax erlaubt:

```
pred komponist(int, atom, atom, int, int).
```

Falls eine Regel einen Fehler enthält, geben Sie bitte (mindestens) eine Fehlermeldung der folgenden Form aus:

```
Typfehler in: mozart(X1, X2) :- komponist('Mozart', X1, X2, X3, X4).
```

Die Variablennamen brauchen dabei nicht zu stimmen (Sie geben einfach die eingelesene Regel mit `write` aus). Selbstverständlich dürfen Sie auch bessere Fehlermeldungen erzeugen.

Hinweise: Eine Datei können Sie z.B. mit folgendem Aufruf öffnen:

```
open('bsp.pro', read, S).
```

Dabei wird `S` an ein Objekt vom Typ “Stream” gebunden (offene Datei, Filepointer). Das Stream-Objekt können Sie z.B. beim Aufruf von `read` verwenden:

```
read(S, X).
```

Am Dateiende ist `X = end_of_file`. Sie sollten dann `close(S)` aufrufen.

Sie müssen die eingelesenen Prädikat-Deklarationen jetzt in die dynamische Datenbasis eintragen (mit `asserta/assertz` — GNU Prolog hat kein `assert`). Dazu ist die Deklaration

```
:- dynamic(pred/1).
```

in Ihrem Quellprogramm erforderlich. Am Anfang oder am Ende der Typprüfung sollten Sie die `pred`-Fakten wieder löschen (mit `retract` bzw. `retractall`). So könnte man Ihren Typprüfer in einer Prolog-Sitzung nacheinander für mehrere Eingabedateien aufrufen.

Abgabetermin:

Bitte geben Sie Ihre Lösung bis zum Montag, den **26. Juni 2006**, ab.