Prof. Dr. Stefan Brass
Institut für Informatik
Universität Halle

14.04.2011

Deduktive Datenbanken und Logische Programmierung — Präsenzübung 0: Benutzung von Prolog —

Aufgabe 0 (0 Punkte)

Unter der URI `http://www.informatik.uni-halle.de/~brass/lp11/family.pro` können Sie die Daten des Stammbaums als Prolog-Fakten herunterladen.

parents

Child	Father	Mother
emil	arno	birgit
frida	chris	doris
gerd	chris	doris
ian	emil	frida
julia	emil	frida
klaus	gerd	helga

couple

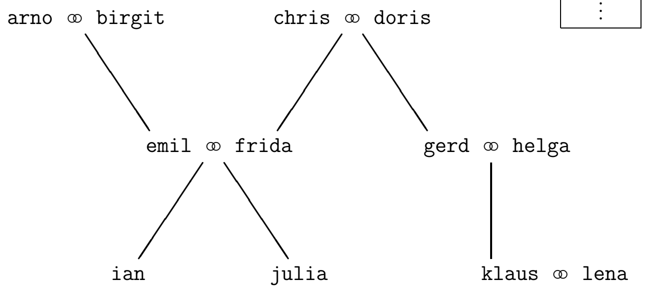
Man	Woman
arno	birgit
chris	doris
emil	frida
gerd	helga
klaus	lena

man

Name
arno
⋮

woman

Name
birgit
⋮



a) Legen Sie eine Datei an, in die Sie die aus der Vorlesung bekannten Regeln schreiben:

$$
\begin{array}{lll}
\texttt{father}(\texttt{X},\texttt{Y}) & \leftarrow & \texttt{parents}(\texttt{X},\texttt{Y},\texttt{Z}).\\
\texttt{mother}(\texttt{X},\texttt{Z}) & \leftarrow & \texttt{parents}(\texttt{X},\texttt{Y},\texttt{Z}).\\
\texttt{parent}(\texttt{X},\texttt{Y}) & \leftarrow & \texttt{father}(\texttt{X},\texttt{Y}).\\
\texttt{parent}(\texttt{X},\texttt{Y}) & \leftarrow & \texttt{mother}(\texttt{X},\texttt{Y}).\\
\texttt{grandparent}(\texttt{X},\texttt{Z}) & \leftarrow & \texttt{parent}(\texttt{X},\texttt{Y}) \land \texttt{parent}(\texttt{Y},\texttt{Z}).\\
\texttt{person}(\texttt{X},\texttt{m}) & \leftarrow & \texttt{man}(\texttt{X}).\\
\texttt{person}(\texttt{X},\texttt{f}) & \leftarrow & \texttt{woman}(\texttt{X}).
\end{array}
$$

Achten Sie dabei auf die Prolog-Syntax (`:-` statt $\leftarrow$ und `,` statt $\land$).

b) Starten Sie SWI-Prolog mit dem Befehl `prolog` (mindestens auf `anubis`). Laden Sie die Datei `family.pro` mit dem Befehl "`['family.pro'].`". Falls Sie die Datei in `family.pl` umbenennen, ghet auch "`[family].`". Laden Sie anschließend Ihre Datei mit den Regeln. Wenn es Syntaxfehler geben sollte, korrigieren Sie diese, und laden Sie die Datei neu: Die neuen Definitionen der Prädikate überschreiben die alten.

c) Probieren Sie die in der Vorlesung gegebenen Anfragen aus (in Prolog müssen Sie nicht das Fragezeichen schreiben, da der Interpreter automatisch im Anfragemodus ist):

- ? $\texttt{grandparent}(\texttt{ian},\texttt{birgit})$.
  $\longrightarrow$ Yes.
- ? $\texttt{grandparent}(\texttt{klaus},\texttt{arno})$.
  $\longrightarrow$ No.
- ? $\texttt{mother}(\texttt{frida},\texttt{X})$.
  $\longrightarrow$ $\texttt{X} = \texttt{doris}$.
- ? $\texttt{mother}(\texttt{X},\texttt{doris})$.
  $\longrightarrow$ $\texttt{X} = \texttt{frida}$.
  $\texttt{X} = \texttt{gerd}$.
- ? $\texttt{mother}(\texttt{X},\texttt{Y})$.
  $\longrightarrow$ $\texttt{X} = \texttt{emil}$, $\texttt{Y} = \texttt{birgit}$.
  $\texttt{X} = \texttt{frida}$, $\texttt{Y} = \texttt{doris}$.
  $\texttt{X} = \texttt{gerd}$, $\texttt{Y} = \texttt{doris}$.
  ⋮ ⋮
- ? $\texttt{father}(\texttt{emil},\texttt{X}) \land \texttt{mother}(\texttt{emil},\texttt{Y})$.
  $\longrightarrow$ $\texttt{X} = \texttt{arno}$, $\texttt{Y} = \texttt{birgit}$.

d) Definieren Sie ein Prädikat `married_with` ("verheiratet mit"): Dies verwendet die Information aus der Datenbank-Relation `couple`, aber soll symmetrisch sein: Wenn `X` mit `Y` verheiratet ist, ist natürlich auch `Y` mit `X` verheiratet. Versuchen Sie Endlosschleifen in Prolog zu vermeiden. Ggf. können Sie die Anfrageauswertung mit `Crtl+C` unterbrechen. Wenn Sie dann das Kommando `a` ("abort") eingeben, wird die Ausführung der Anfrage abgebrochen.

e) Definieren Sie ein Prädikat `siblings` ("Geschwister", Vater und Mutter gemeinsam). Sie können die Bedingung $\texttt{X} \neq \texttt{Y}$ (in Prolog `X \= Y`) im Rumpf verwenden (in Prolog sollten beide Variablen vorher, also weiter links, an einen Wert gebunden werden).

f) Definieren Sie ein Prädikat `uncle` (ein Bruder eines Elternteils).

g) Definieren Sie ein Prädikat für Konsistenztests: Gibt es eine Person, die gleichzeitig männlich und weiblich ist? Das Prädikat soll `inconsistent` heissen und herleitbar sein, wenn es ein Problem gibt. Es soll keine Argumente haben. Natürlich können Sie bei allen Aufgaben auch weitere Hilfsprädikate deklarieren. Wenn Sie alles ausprobiert haben, können Sie Prolog mit "`halt.`" beenden. Probieren Sie aber z.B. auch "`help(halt/0).`" (0 ist die Anzahl Argumente).