

Logische Programmierung & deduktive Datenbanken — Übungsblatt 8 (Zahlenrätsel) —

Ihre Lösungen zu den Hausaufgaben a) und b) geben Sie bitte über die Übungs-Plattform in StudIP ab. Einsendeschluss ist der 16. Juni, 10⁰⁰ (die Aufgaben werden anschließend in der Übung besprochen).

Hausaufgaben

- a) Schauen Sie sich nochmal das Programm zur Berechnung von Knoten eines SLD-Baums an:

[<http://www.informatik.uni-halle.de/~brass/lp22/prolog/sld1.pl>]

Ändern Sie den Daten-Teil so, dass die Knoten für die Anfrage `grandfather(ian, G)` an das folgende Programm berechnet werden (Aufgabe d) vom letzten Übungsblatt):

```
[1] grandfather(X, Z) :- grandparent(X, Z), male(Z).
[2] grandparent(X, Z) :- parent(X, Y), parent(Y, Z).
[3] parent(eric, alan).
[4] parent(eric, bianca).
[5] parent(fiona, chris).
[6] parent(fiona, doris).
[7] parent(ian, eric).
[8] parent(ian, fiona).
[9] male(alan).
[10] male(chris).
[11] male(eric).
[12] male(ian).
```

Wenn Sie wollen, dürfen Sie das Programm auch verbessern, so dass die Ausgabe nützlicher wird. Das müssen Sie aber nicht tun. Vergleichen Sie die Programmausgabe mit dem von Ihnen gezeichneten SLD-Baum.

- b) Ein bekannter Typ von Rätsel ist:

```
SEND
+MORE
-----
MONEY
```

Man muss Ziffern 0 . . . 9 für die Buchstaben (Variablen) finden, so dass

- Die Additionsaufgabe korrekt ist,
- verschiedene Buchstaben mit unterschiedlichen Ziffern belegt werden,
- die ersten Buchstaben in jeder Zeile nicht mit 0 belegt werden.

Eine simple Formulierung dieses Rätsels in Prolog ist:

```
generate([S,E,N,D,M,O,R,Y]),
S \= 0,
M \= 0,
to_int([S,E,N,D], I1),
to_int([M,O,R,E], I2),
to_int([M,O,N,E,Y], I3),
I3 =:= I1 + I2.
```

Lösen Sie folgende Teilaufgaben:

- Definieren Sie zunächst das Hilfsprädikat `generate(L)` zum Einsatz im obigen Programmstück. Dieses Prädikat soll alle Elemente der Liste `L` mit unterschiedlichen Ziffern belegen (auf alle möglichen Arten). Die Ziffern sollen dabei als ganze Zahlen mit Wert zwischen 0 und 9 repräsentiert werden (nicht als Atome/Zeichen). Die Liste `L` besteht aus Variablen oder Ziffern.
- Als nächstes definieren Sie das Hilfsprädikat `to_int(L,I)`. Es soll den Zahlwert `I` einer Ziffernliste `L` berechnen.
- Nun definieren Sie ein Prädikat zur Rätsel-Lösung, das man so aufrufen kann:

```
raetsel([S,E,N,D], [M,O,R,E], [M,O,N,E,Y]).
```

Wenn Sie das obige Schema nutzen wollen, müssen Sie eine Liste der unterschiedlichen Variablen berechnen, die in den drei Eingabelisten vorkommen. Benutzen Sie `append` um die Listen zu einer Liste zu konkatenieren, und rufen Sie dann das folgende Prädikat auf, um doppelte Variablen zu eliminieren:

```
distinct([], []).
distinct([E|R], OUT) :-
    (var_member(E, R) -> OUT=L; OUT=[E|L]),
    distinct(R, L).

var_member(X, [Y|_]) :-
    X == Y.
var_member(X, [_|L]) :-
    var_member(X, L).
```

Wenn Sie wollen, können Sie prüfen, ob die folgenden Rätsel lösbar sind, die sich im Internet finden:

- `raetsel([B,I,L,L], [I,R,M,A], [L,I,E,B,E])`.
- `raetsel([A,A,L], [A,A,L], [F,A,N,G])`.
- `raetsel([G,A,U,S,S], [R,I,E,S,E], [E,U,K,L,I,D])`.
- `raetsel([V,A,T,E,R], [M,U,T,T,E,R], [E,L,T,E,R,N])`.

Zur Wiederholung

c) Was würden Sie in einer mündlichen Prüfung auf die folgenden Fragen zum minimalen Modell und zum T_P -Operator antworten?

- Definieren Sie den Begriff „Minimales Herbrandmodell“ eines logischen Programms P .
- Gegeben sei das folgende logische Programm:

$$\begin{aligned} & p(a). \\ & q(b). \\ & r(X) \text{ :- } p(X). \end{aligned}$$

Was ist das minimale Herbrandmodell dieses Programms? Geben Sie ein Beispiel für ein nicht-minimales Herbrandmodell dieses Programms. Warum gibt es nicht-minimale Modelle?

- Hat jedes logische Programm (Menge definiter Hornklauseln) ein Herbrandmodell? Gibt es immer ein minimales Modell? Kann es auch mehr als ein minimales Herbrandmodell geben?
- Wie kann man das minimale Herbrandmodell konstruieren?
- Gegeben sei ein logisches Programm P . Definieren Sie den Operator T_P .
- Was ist ein „Fixpunkt“ einer Abbildung, speziell des Operators T_P ? Gibt es immer einen Fixpunkt? Geben Sie ein Beispiel für ein logisches Programm, für das der T_P -Operator mehr als einen Fixpunkt hat. Wie ist „kleinster Fixpunkt“ definiert? Gibt es immer einen kleinsten Fixpunkt? Ist dieser eindeutig? (Es wäre schön, wenn Sie den entsprechenden Satz wiedergeben könnten.)
- Warum ist der kleinste Fixpunkt von T_P wichtig? Was ist der Zusammenhang zum minimalen Modell?
- Wie kann man das minimale Modell mit dem T_P -Operator berechnen? Schreiben Sie die Iteration formal auf.

- Angenommen, die Iteration des T_P -Operators terminiert nicht. Was kann man dennoch über die Beziehung zum minimalen Modell aussagen?
- Was ist ein kausales Modell („supported model“)? Welche Beziehung gibt es zum minimalen Modell?

Zum Selbststudium

d) 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 Internet zum Thema dieser Vorlesung gibt, und dabei für sich nützliche Quellen entdecken.

- Hier ist eine Sammlung von Aufgaben zu Prolog (von Karsten Hönig, einem Informatik-Lehrer):

[<http://www.hoenig.info/informatik/prolog/Word/InGN13%20-%20KI%20Aufgaben%20PROLOG%202.doc>]

- Ein kommerzieller Anbieter eines deduktiven Datenbanksystems ist LogicBlox:

[<https://www.logicblox.com/>]

Eine kurze Übersicht zum System und seiner Sprache findet sich im Artikel „LogicBlox, Platform as Language: a Tutorial“ von Todd J. Green, Molham Aref, and Grigoris Karvounarakis:

[<https://developer.logicblox.com/2012/08/logicblox-platform-and-language-a-tutorial/>]

Direkter Link zum PDF:

[<https://developer.logicblox.com/wp-content/uploads/2013/10/datalog2020tutorial-1.pdf>]

Eine etwas ausführlichere Erklärung zur Funktionsweise ist der Artikel „Design and Implementation of the LogicBlox System“ von Molham Aref et.al.:

[<http://www.cs.ox.ac.uk/dan.olteanu/papers/logicblox-sigmod15.pdf>]