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

24.05.2018

Logische Programmierung & deduktive Datenbanken — Übungsblatt 7 (SLD-Resolution) —

Ihre Lösungen zu den Hausaufgaben g) und h) schicken Sie bitte per EMail an den jeweiligen Übungsleiter (mit "`[lp18]`" in der Betreff-Zeile). Einsendeschluss ist der 2. Juni.

Zum Selbststudium

a) 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 WWW zum Thema dieser Vorlesung gibt, und dabei für sich nützliche Quellen entdecken.

- Zu der Vorlesung "Deduktive Datenbanken" von Prof. Dr. Dietmar Seipel (Universität Würzburg) gibt es den Foliensatz online:

  [http://www1.pub.informatik.uni-wuerzburg.de/databases/courses/ddb_ss2015.html]

- Das Buch "Prolog Programming in Depth" von Michael A. Covington, Donald Nute und André Vellino (2. Auflage, Prentice Hall, 1997) gibt es hier als PDF:

  [http://www.covingtoninnovations.com/books/PPID.pdf]

  Siehe auch:

  [http://www.covingtoninnovations.com/books.html]

b) Was würden Sie in einer mündlichen Prüfung auf die folgenden Fragen antworten?

- Definieren Sie einen SLD-Ableitungsschritt.
- Welche Selektionsfunktion wird von Prolog benutzt? Warum vereinfacht diese Selektionsfunktion die Implementierung?
- Definieren Sie "erfolgreiche SLD-Ableitung". Was ist die dabei berechnete Antwort?
- Wie lautet der Satz über die Korrektheit der SLD-Resolution? Gilt die Korrektheit auch für Prolog?
- Wie lautet der Satz über die Vollständigkeit der SLD-Resolution? Gilt dies auch für Prolog?

- Wie wird der SLD-Ableitungsbaum konstruiert? Was ist die Ursache der Verzweigungen im SLD-Ableitungsbaum?
- Wie sucht Prolog den SLD-Ableitungsbaum ab? Was ist das Problem dabei?
- Zeichnen Sie den SLD-Ableitungsbaum für ein Beispiel auf.
- Wie heißen die vier Ports im "Box Model" des Prolog-Debuggers? An welchen Stellen im Kasten sind sie, und welchen Vorteil hat diese Anordnung?
- Mit welchem Kommando bekommt man in einen Prolog-System die Ausgaben zur Programm-Ausführung nach dem Vier-Port-Modell? Wie kann man sie wieder abschalten? Wie heißen "Breakpoints" in Prolog?
- Was unterscheidet Prolog-Prädikate von Prozeduren in klassischer Programmierung?

Präsenzaufgaben

c) Zeichnen Sie einen SLD-Baum für folgenden Programm und die Anfrage `p(X)` (nutzen Sie dabei die "First Literal" Selektionsfunktion wie in Prolog):

```
p(X) :- q(X), r(X).
q(a) :- s(Y), t(Y).
q(b).
r(a).
r(b).
s(c).
s(d).
t(c).
```

d) Angenommen, Regeln sind als "Fakten" (mit Variablen) über ein Prädikat `rule` repräsentiert mit

- dem Kopf der Regel als erstem Argument, und
- der Liste der Rumpf-Literale als zweitem Argument.

Z.B. würde die Regel `p(X) :- q(X), r(X)` so gespeichert:

```
rule(p(X), [q(X), r(X)]).
```

(Normal sind "Fakten" variablenfreie Regeln ohne Rumpf, deswegen ist die Bezeichnung hier nicht ganz korrekt.)

Man kann nun leicht einen Meta-Interpreter in Prolog schreiben, der sämtliche Knoten des SLD-Baums berechnet:

```
sld([]). % Empty Goal: Query proven.
sld([Lit|Rest]) :-
    rule(Lit, Body), % Unification happens here.
    append(Body, Rest, Child),
    sld(Child).
```

Man ruft das Prädikat mit der zu beweisenen Aussage auf, z.B.:

```
sld([p(X)]).
```

Erweitern Sie das Programm so, dass es jeden Knoten in einer Zeile ausgibt, und zwar so eingerückt, dass man die Baumstruktur erkennen kann: Jeder Kindknoten soll vier Leerzeichen weiter eingerückt sein als sein Elternknoten. Dazu müssen Sie das Prädikat `sld` um ein zweites Argument für die AUfruftiefe erweitern.

Wenn Sie wollen, können Sie zusätzlich auch die Regel ausgeben, die zu diesem Knoten geführt hat (bzw. "`(QUERY)`", falls es der Wurzelknoten ist).

e) Schreiben Sie ein Prädikat `perm(N,L)`, das alle Listen `L`, die sich durch Permutation der Liste `[1,2,...,N]` ergeben, liefert. Z.B. soll `perm(2,L)` die beiden Lösungen `L=[1,2]` und `L=[2,1]` haben. Sie können sich z.B. rekursiv die Permutationen für die Listen bis `N-1` generieren lassen, und dann die Zahl `N` an jeder möglichen Position einfügen (von ganz hinten bis ganz vorne).

f) Schreiben Sie ein Prädikat `simplify(A,B)`, das einen arithmetischen Ausdruck `A` vereinfacht, und in `B` den vereinfachten Ausdruck liefert. Der arithmetische Ausdruck kann das Atom `x` enthalten, sonst nur Zahlkonstanten und die Operatoren `+`, `-`, `*`, `/` und `^`. Mindestens sollen Sie Summen mit `0` auf einer Seite, Produkte mit `0` und `1`, und `^0` (hoch 0) und `^1` (hoch 1) vereinfachen. Der arithmetische Ausdruck kann z.B. durch Anwendung der Ableitungsregeln auf ein Polynom (siehe Aufgabenblatt 4) berechnet worden sein.

Mit `number(Term)` können Sie testen, ob `Term` eine Zahlkonstante ist.

Wenn Sie gleiche Teile für verschiedene Operatoren nur einmal aufschreiben wollen, ist es nötig, Terme in Operator und Argumente zu zerlegen. Dazu können Sie z.B. das eingebaute Prädikat `=..` Verwenden, z.B. gilt `(2*x) =.. [*,2,x]`. Man erhält also eine Liste aus Operator und Argument-Termen. Das Prädikat funktioniert auch in umgekehrter Richtung, um einen Term aufzubauen. Wenn Sie die Stelligkeit des Operators kennen, können Sie auch mit `functor(Term, Funktor, Stelligkeit)` und `arg(ArgNo, Term, Value)` den Term in seine Bestandteile zerlegen.

Hausaufgabe

g) Lösen Sie das "`N`-Damen-Problem": Auf einem Schachbrett der Größe `N*N` sollen `N` Damen so plaziert werden, dass sie sich nach den Schach-Regeln nicht gegenseitig bedrohen, d.h.

- Keine Zeile darf zwei Damen enthalten (da insgesamt `N` Damen aufgestellt werden sollen, muss jede Zeile genau eine Dame enthalten).
- Keine Spalte darf zwei Damen enthalten (auch hier muss jede Spalte genau eine Dame enthalten).
- Keine zwei Damen dürfen in einer Diagonale stehen, d.h. für je zwei Damen an den Positionen (x_1, y_1) und (x_2, y_2) muss $|x_2 - x_1| \neq |y_2 - y_1|$ sein.

Ihr Prädikat `nqueens(N,L)` soll in `L` die Liste der Spaltennummern für die Damen in den Zeilen `1` bis `N` liefern (die Liste muss also die Länge `N` haben). Mehr Informationen finden Sie in der Wikipedia:

[https://en.wikipedia.org/wiki/Eight_queens_puzzle]

Z.B. gibt es für `N=4` zwei Lösungen: `[3,1,4,2]` und `[2,4,1,3]`.

Natürlich gibt es für dieses bekannte Problem Lösungen im Netz. Falls Sie eine Lösung verwenden, die nicht von Ihnen stammt, müssen Sie die Quelle offenlegen, und in der Übung die Lösung genau erklären. In jedem Fall muss der Dozent ohne größere Anstrengungen verstehen können, wie Ihre Lösung arbeitet. Verwenden Sie ausreichend Kommentare!

h) Schreiben Sie ein Prädikat `print_queens(L)`, das Listen aus g) z.B. in folgender Form druckt:

```
..*.
*...
...*
.*..
```

Dies würde der Liste `[3,1,4,2]` entsprechen. Wenn Sie wollen, können Sie es auch schöner ausgeben. Man könnte z.B. einen Rahmen zeichnen:

```
+-+-+-+-+
| | |*| |
|*| | | |
| | | |*|
| |*| | |
+-+-+-+-+
```

Ideen sind willkommen.