**3. Übung zur Vorlesung "Objektorientierte und Deduktive Datenbanken"**
**im Wintersemester 2004**
**– mit Musterlösungen –**

Prof. Dr. Gerd Stumme, Dipl.-Inform. Christoph Schmitz

22. November 2004

## Aufgabe 1 – Prädikatenlogik, Resolutionskalkül

Gegeben sei die folgende Menge $\Phi$ von Formeln:

$$\begin{gathered} next(odeon, stmichel) \\ next(stmichel, chatelet) \\ \forall x.\forall y.(next(x,y) \rightarrow connected(x,y)) \\ \forall x.\forall y.((\exists z.(next(x,z) \land next(z,y))) \rightarrow connected(x,y)) \end{gathered}$$

1. Zeigen Sie mit Hilfe des Resolutionskalküls (sollte aus Theoretische Informatik II bekannt sein), daß gilt:

   $$\Phi \vdash connected(odeon, chatelet).$$

   Zur Erinnerung:

   - Die Matrix der Formelmenge $\Phi$ in Skolemform lautet wie folgt:

     $$\begin{gathered} \{\neg next(x,y), conn(x,y)\} \\ \{\neg conn(x,z), \neg next(z,y), conn(x,y)\} \\ \{next(odeon, stmichel)\} \\ \{next(stmichel, chatelet)\} \end{gathered}$$

   - Um $\Phi \vdash \psi$ zu zeigen, muß man aus $\Phi \cup \{\neg\psi\}$ die leere Klausel $\square$ resolvieren können.

- Ein Resolutionsschritt macht im Wesentlichen Folgendes:
  Seien $F_1 = \{L_1, L_2, \ldots, L_k\}, F_2 = \{\neg L_1, L_{k+1}, \ldots, L_n\}$. Dann ist $\{L_2, \ldots, L_k, L_{k+1}, \ldots, L_n\}$ der Resolvent aus $F_1$ und $F_2$ (nach geeigneter Substitution).

Wir beginnen mit den gegebenen Klauseln plus der Negation des zu Zeigenden:

$$
\begin{aligned}
&\{\neg next(x,y), conn(x,y)\} && (1)\\
&\{\neg conn(x,z), \neg next(z,y), conn(x,y)\} && (2)\\
&\{next(odeon, stmichel)\} && (3)\\
&\{next(stmichel, chatelet)\} && (4)\\
&\{\neg conn(odeon, chatelet)\} && (5)\\
&\{conn(odeon, stmichel)\} && (6)\\
&\text{aus } 1,3 \text{ mit } [x/odeon], [y/stmichel] && \\
&\{\neg next(stmichel, y), conn(odeon, y)\} && (7)\\
&\text{aus } 2,6 \text{ mit } [x/odeon], [z/stmichel] && \\
&\{conn(odeon, chatelet)\} && (8)\\
&\text{aus } 4,7 \text{ mit } [y/chatelet] && \\
&\square \text{ aus } 8,5 && (9)
\end{aligned}
$$

2. Geben Sie ein Programm im Pseudocode angelehnt an eine Programmiersprache Ihrer Wahl an, das für zwei Stationen $x$ und $y$ berechnet, ob $y$ mit $x$ verbunden ist. *next* sei dabei z. B. als zweidimensionales Feld `next` repräsentiert mit `next[von][nach]=1` gdw. $next(von, nach)$.

   Ein einfaches Programm könnte lauten:

   ```
   conn ← next
   repeat
     for all (i, j) mit conn[i, j] = 1 do
       for all k mit next[j, k] = 1 do
         conn[i,k] ← 1
       end for
     end for
   until keine Änderungen mehr in conn
   ```

   Effizienter wird es, wenn man in jeder Iteration nur die neu dazugekommenen `conn[i,k]` berücksichtigt.

   Dieses Programm entspricht einem einfachen Auswertungsalgorithmus für die Fixpunktsemantik von Datalog-Programmen (→ später, Kap. 12).

3. Welche Variante (Resolutionskalkül oder Ihr Programm) ist Ihres Erachtens eleganter?

Natürlich lassen sich für die meisten Probleme elegante Lösungen programmieren. Der Vorteil der Anwendung der Prädikatenlogik liegt in der deklarativen Beschreibung von Problemen und der allgemeinen Anwendbarkeit der entsprechenden Kalküle.

Logikprogrammierung bietet einen Mittelweg zwischen (prozeduraler) Programmierung und Prädikatenlogik + Kalkül, indem höhere Effizienz gegenüber voller Prädikatenlogik durch eingeschränkte Ausdrucksstärke erkauft wird.

## Aufgabe 2 – Prädikatenlogik

Vervollständigen Sie die rekursive Definition für $I \models \phi[\mu]$ aus der Vorlesung:

- Wenn $\phi = \psi_1 \wedge \psi_2$, dann $I \models \phi[\mu]$ gdw. ...
  ... $I \models \psi_1[\mu]$ und $I \models \psi_2[\mu]$
- Wenn $\phi = \psi_1 \rightarrow \psi_2$, dann $I \models \phi[\mu]$ gdw. ...
  ... $I \not\models \psi_1[\mu]$ oder $I \models \psi_2[\mu]$
- Wenn $\phi = \psi_1 \leftrightarrow \psi_2$, dann $I \models \phi[\mu]$ gdw. ...
  ... $I \models \psi_1[\mu]$ und $I \models \psi_2[\mu]$ oder $I \not\models \psi_1[\mu]$ und $I \not\models \psi_2[\mu]$
- Wenn $\phi = \neg\psi$, dann $I \models \phi[\mu]$ gdw. ...
  ... $I \not\models \psi[\mu]$

## Aufgabe 3 – Herbrand-Interpretationen, -modelle

Sei $\Psi = \{p(a), q(a, f(b)), \forall x.(p(x) \rightarrow q(x,x))\}$. Mit $a, b, \ldots$ als Konstantensymbolen, $f, g, \ldots$ als Funktionssymbolen, $p, q, \ldots$ als Prädikatensymbolen und $x, y, \ldots$ als Variablensymbolen:

1. Wie müßte eine Signatur $\Sigma$ aussehen, damit $\Psi$ syntaktisch korrekt ist?
   $\Sigma = (\{a, b\}, \{f\}, \{p, q\}, d)$ mit $d(f) = 1, d(p) = 1, d(q) = 2$.
2. Skizzieren Sie das entsprechende Herbrand-Universum und die Herbrand-Basis, sowie die entsprechende Herbrand Präinterpretation!
   Universum: $T_\Sigma(\emptyset) = \{a, b, f(a), f(b), f(f(a)), f(f(b)), \ldots\}$
   Basis: $\{p(a), p(b), q(a,a), \ldots\ q(b,b), p(f(a)), \ldots, q(f(a), a), \ldots\}$
   Präinterpretation: $(U, \mathcal{C}, \mathcal{F})$ mit
   $U := T_\Sigma(\emptyset)$
   $\mathcal{C}(c) := c$
   $\mathcal{F}(f)(\text{“}t_1\text{”}, \ldots \text{“}t_m\text{”}) := \text{“}f(t_1, \ldots, t_m)\text{”}$

3. Mit welcher der folgenden Abbildungen $P_i$ wird aus dieser Präinterpretation eine Herbrand-Interpretation?

   Mit jeder, da alle Prädikate mit den richtigen Stelligkeiten interpretieren.

4. Welche dieser Interpretationen sind Herbrand-Modelle für $\Psi$?

   - $P_1$: nein, da $(a, a) \notin P_1(q)$.
   - $P_2$: ja.
   - $P_3$: nein, da $(b, b), (f(f(a)), f(f(a)) \notin P_2(q)$.
   - $P_4$: ja.
   - $P_5$: nein, da $(a, a), (b, b) \notin P_1(q)$.

- $P_1(p) := \{(a), (b)\}, P_1(q) := \{(a, b), (b, b)\}$
- $P_2(p) := \{(a)\}, P_2(q) := \{(a, a), (a, f(b))\}$
- $P_3(p) := \{(f(f(a))), (b)\}, P_3(q) := \{(a, a), (a, f(b))\}$
- $P_4(p) := \{(a), (b)\}, P_4(q) := \{(a, a), (b, b), (a, f(b))\}$
- $P_5(p) := \{(a), (b)\}, P_5(q) := \{(a, b)\}, P_5(r) := \{(b, b)\}$