

Deduktive Datenbanksysteme

Problem: Transitiver Abschluss ist in PL1 nicht formulierbar (mit zustandsabhängiger Formulierung möglich)

Diskussion:

Typ 5: $q_1(\dots), \dots, q_n(\dots) : -.$

Typ 6: $q_1(\dots), \dots, q_n(\dots) : -p_1(\dots), \dots, p_m(\dots).$

\Rightarrow Übungsaufgabe

Nur Typ 1 und Typ 4: $q(\dots) : -p_1(\dots), \dots, p_n(\dots), n \geq 0$ (ist die Hornklauselform und wird bei definiten Datenbanken genutzt)

Definite Datenbanken

$$\begin{aligned} q(\dots) &: -.\text{(Fakt)} \\ q(\dots) &: -p_1(\dots), \dots, p_n(\dots).\text{(deduktive Regel, } p_{1-n} \text{ Teilziele)} \end{aligned} \quad (1)$$

- Mit IBen (Integritätsbedingungen) (+ Typ2, Typ3) ($: -p_1(\dots), \dots, p_n(\dots)$)
- Typ5 + Typ6 \Rightarrow **Disjunktive Datenbank**
- Definite Datenbank + negative Atome im Rumpf von Hornklauseln erlaubt \Rightarrow Volles Datalog

Formulierung von Anfragen

Klauseln vom Typ: $: -p_1(\dots), \dots, p_n(\dots)$, geschrieben $? - p_1(\dots), \dots, p_n(\dots)$

Beispiele:

- $? - ag(X, m).$
 - Bedeutung: Welche Kurse bietet 'm' an?
 - DRC: (x) / ANGEBOT(X, m)
- $? - ag(a3, m).$
 - Bedeutung: Bietet 'm' den Kurs 'a3' an?
 - DRC: () / ANGEBOT(a3, m)
- $? - ag(X, m), bl(X, s, j).$

- Bedeutung: Gib alle von 'm' angebotene Kurse, die 's' als Wiederholer belegt hat
- DRC: $(x) / \text{ANGEBOT}(x, 'm') \wedge \text{BELEGUNG}(x, 's', 'y')$
- Wie ist $(x) / \text{ANGEBOT}(x, 'm') \wedge (\exists y) \text{BELEGUNG}(x, 's', y)$ formulierbar?
 - Bedeutung: Gib die Dozenten der von s als Wiederholer belegte Kurse
 - Formulierung: $? - Ksm(X)$.
 $Ksm(X) : -ag(X, m), bl(X, s, y)$
 - Bequemer: $? - ag(X, Y*), bl(X, s, y)$., * Kennzeichnet die Ausgabevariable

In Anfragesprachen werden Vergleichsausdrücke benötigt. Dazu sind in Datalog spezielle vordefinierte Prädikate vorhanden. Für jeden Vergleichsoperator wird die Existenz eines solchen Prädikates angenommen.

Zunächst: Beschränkte Variablen in Regeln. Sei eine Regel r gegeben:

- Jede Variable, die als Argument in einem gewöhnlichen Prädikat im Rumpf von r vorkommt ist beschränkt.
- Jede Variable, die in einem Teilziel $X = c$ oder $c = X$ von r vorkommt, ist beschränkt.
- Eine Variable X ist beschränkt, wenn sie in einem Teilziel $X = Y$ oder $Y = X$ von r vorkommt mit Y ist schon als beschränkt bekannt.

Definition: sicher

Eine Regel heißt sicher, wenn alle in ihr vorkommenden Variablen beschränkt sind.

Beispiele:

- $Kls(X, Y) : -bl(Z, s, j), ag(Z, Y), X = Z$. **sicher**
- $vsj(X, Y) : -bl(Y, s, j)$. **nicht sicher** (X ist nicht beschränkt)
- $vs(X, Y) : -vs(X, Z), kp(Z, Y)$. **sicher, wenn vs terminiert**
- $kla(Z, Y) : -bl(Z, V, j), ag(Z, Y), V \neq s$. **sicher**

Bemerkung: Falls keine Build-in Prädikate erlaubt sind (/vorkommen):
 Eine Regel ist sicher genau dann wenn jede Variable im Kopf der Regel auch im Rumpf der Regel vorkommt.

Definition: Datalog Programm

Ein Datalog-Programm P (ohne IBen(Integritätsbedingungen)) ist eine endliche Menge von Horn-Klauseln mit Jedes $d \in P$ ist entweder

- ein Fakt $q(\dots)$. ohne Variable
- eine sichere Regel $q(\dots) : \neg p_1(\dots), \dots, p_n(\dots)$. mit $q \in iPraedikat$

Ein $d \in P$ heißt auch **Datalog-Klausel** Alle Fakten zu extensionalen Prädikaten sind als in DB-Relationen gespeichert zu denken.

Beispiel Datenbankzustand:

$$\begin{aligned}
 &ag(a1, m). \\
 &kp(c2, a0). \\
 &\dots \\
 &rb(a1, r1, t1) (Kurs\ a1\ im\ Raum\ r1\ zu\ t1) \\
 &rb(a3, r2, t4)
 \end{aligned} \tag{2}$$

Angebot:

Kursnummer	Dozent
a1	m
...	...

Kursplan:

Kursnummer	Voraussetzung
c2	a0
...	...

Raumbelegung:

Kursnummer	Raum	Zeit
a1	r1	t1
...

Belegung:

Kursnummer	Teilnehmer	Wiederholer
a1	s	j
...	...	
	...	

$$vs(X, Y) : \neg Kp(X, Y).$$

$$vs(X, Y) : \neg vs(X, Z), Kp(Z, Y).$$

$$stdpl(W, X, Y, Z) : \neg bl(X, W, V), rb(X, Y, Z).$$

$$ueberschneidungen(X, Y) : \neg Kp(Z, X), Kp(Z, Y), rb(X, V, T), rb(Y, W, T), X \neq Y. \quad (3)$$

Deklarative Semantik

Extensionale Prädikate eines Programms (ext. Rel, ext. DB): EDB

Intentionale Prädikate eines Programms (int. Rel, int. DB): IDB

Bedeutung Bedeutung eines Datalog-Programms P: Menge derjenigen Grundatome zu den intentionalen Prädikaten von P, die logisch aus P gefolgert werden können. (Jedes Modell von P ist auch ein Modell von $f \in F$). Mit Zielklausel $g(\dots)$, $g \in Praed$: Aus P logisch folgbare Grundatome zu g, die von $g(\dots)$ subsummiert

(überdeckt) werden. $\frac{g(a, X)}{g(a, b)}$
 $\frac{g(a, X)}{g(a, c)}$

In P werden Werte aus Wertebereichen verwendet, ebenso in Darstellung der extensionalen Prädikate als DB-Relation. Daher können wir $Kost_A$ und Dom identifizieren. Mithilfe der Herbrand Interpretation kann die Semantik festgelegt werden (ist möglich).

Herbrand-Interpretation /-Modelle Gewöhnliche Interpretation:

$$\begin{aligned} Konst_A &= \{a, b\}, Dom = \{\circ, \square\} \\ k(a) &= \circ \\ k(b) &= \square \\ ext(p(\cdot, \cdot)) &= \{(\circ, \square), (\square, \square)\} \end{aligned} \quad (4)$$

eine mögliche Herbrand-Interpretation (passt dazu)

$$\begin{aligned}
 Konst_A &= Dom = \{a, b\} \\
 k(a) &= a \\
 k(b) &= b \\
 ext(p(.,.)) &= \{(a, b), (b, b)\}
 \end{aligned} \tag{5}$$

Entsprechende Herbrand-Interpretation. Betrachte alle Paare zu $p(.,.)$, teste gemäß gegebener (gewöhnlicher) Interpretation in $ext(p(.,.))$.

$$\begin{aligned}
 Konst_A &= \{a, b\}, Dom = \{\circ, \square\} \\
 k(a) &= \square \\
 k(b) &= \square \\
 ext(p(.,.)) &= \{(\circ, \square), (\square, \square)\}
 \end{aligned} \tag{6}$$

(a, a) wird zu $(\square, \square) \in ext(p(.,.))$

Herbrand-Interpretation

$$\begin{aligned}
 Konst_A &= Dom = \{a, b\} \\
 k(a) &= a \\
 k(b) &= b \\
 ext(p(.,.)) &= \{(a, a), (a, b), (b, a), (b, b)\}
 \end{aligned} \tag{7}$$

Bei beiden Interpretationen sind die gleichen Formeln gültig bei Beschränkung auf quantorenfreie Formeln ohne Variablen (und ohne Funktionen).

Beispiel Erste Interpretation: $p(a, b) \wedge p(b, a) \Rightarrow (\square, \square) \in ext(p(.,.)) \wedge \dots$ bzw. $(a, b) \in ext(p(.,.)) \wedge (b, a) \in ext(p(.,.))$

Menge von Konstanten und Prädikatensymbolen ist endlich, daher ist die Anzahl der möglichen Herbrand-Interpretationen endlich.

Satz von Gödel / Skolem (Herbrand, 1930)

Eine Klauselmengen P hat ein Modell genau dann wenn P hat ein Herbrand-Modell. Daraus folgt, dass ein Verfahren analog zu Wahrheitstabellen in der Aussagenlogik möglich ist.

Beispiel $F = \{p(a) \Rightarrow q(b), p(a) \wedge q(b)\}, q(b)?$

p	q	$p(a) \Rightarrow q(b)$ erfüllt?	$p(a) \wedge q(b)$ erfüllt?	$p(a) \Rightarrow q(b)$ und $p(a) \wedge q(b)$ erfüllt?
$\{\}$	$\{\}$	✓	-	-
$\{\}$	$\{a\}$	✓	-	-
$\{\}$	$\{b\}$	✓	✓	✓
$\{\}$	$\{a, b\}$	✓	✓	✓
$\{a\}$	$\{\}$	-	✓	-
...
$\{b\}$	$\{b\}$	✓	✓	✓

Jedes Modell von F ist auch ein Modell von $q(b)$, d.h. $q(b)$ kann aus F logisch gefolgert werden. Gilt bei Klauselmengen, aber **Vorsicht bei allgemeinen Formeln**.

Beispiel: $\{p(a), (\exists X)(\neg p(X))\}$ Formelmenge, keine Klauselmenge

Modell (vgl. Übung):

$$\begin{aligned} Dom &= \{0, 1\} \\ k(a) &= 0 \\ ext(p(.)) &= \{(0)\} \end{aligned} \tag{8}$$

Aber: Es gibt kein durch **ext** bestimmtes Herbrand-Modell:

1. $ext(p(.)) = \{(a)\}, Konst = Dom = \{a\}$
2. $ext(p(.)) = \{\}$

Herbrand-Modell muss genügend viele Elemente enthalten, damit der Satz von Gödel / Skolem gelten kann. **Skolemisierung** bedeutet, dass man alle Existenzquantoren durch Funktionen ersetzt:

$$(\forall x_1, \dots, x_n)(\exists y)(F) \rightsquigarrow (\forall x_1, \dots, x_n)(F[f(x_1, \dots, x_n)/y]) \tag{9}$$

Bemerkung: Skolemisierung

Jede Formel der PL1 Logik kann man in einer erfüllbarkeitsäquivalente Formel in Skolemform umformen:

1. Pränexnormalform

2. Umformungen à la $(\forall x_1, \dots, x_n)(\exists y)(F) \rightsquigarrow (\forall x_1, \dots, x_n)(F[f(x_1, \dots, x_n)/y])$
mit jeweils einem neuen Funktionssymbol

Dies ist eine Art “Materialisierung” der durch den Existenzquantor gebundenen Variablen.

Beispiel (von oben) $\{p(a), \neg p(y)^1\}$ erfüllbar $\iff \{p(a), (\exists X)(\neg p(X))\}$ erfüllbar

Vorsicht: Semantische Äquivalenz von Formeln und ihren Skolem-Normalformen im Allgemeinen nicht gegeben.

Skolem-NF: $(\exists X)(p(X)) : p(a)$ Interpretation I mit

$$\begin{aligned} Dom &= \{1, 2\} \\ k(a) &= 1 \\ ext(p(.)) &= \{(2)\} \end{aligned} \tag{10}$$

$\rightsquigarrow \models_I (\exists X)(p(X))$ Belegung von X mit allen Elementen aus Dom, d.h. auch mit 2.
Aber $\not\models_I p(a)$, da $(1) \notin ext(p(.))$.

Im Kontext von Datalog

- Herbrand-Universum: Konst
- Herbrand-Basis (HB): Menge aller Grundatome
($\rightsquigarrow EDB^2, IDB^3$ (Fakten in der Datenbank und solche die Ableitbar sind) $\subset HB$)
- Herbrand-Interpretation: (Konst, id, ext), d.h. jedes Konstantensymbol wird als es selbst interpretiert. (Verglichen mit relationaler Interpretation, dort k ist Bijektion)
- Jede Herbrand-Interpretation ist eindeutig bestimmt durch ext (Extension, Ausprägung), da Konst und id unveränderlich sind
- Jedes Prädikat ist eindeutig bestimmt durch die Angabe der Grundatome, für die es “wahr” liefert.
- extentional: genaue Antwort in Tupel
- intentional: Beispielsweise Formeln als Antworten

¹neue Variable

²gegeben durch Datenbankzustand

³muss ausgerechnet / gefolgert werden

Definition: Herbrand-Interpretation

Einfache Definition: Eine Herbrand-Interpretation ist eine Teilmenge der Herbrand-Basis.

Beispiel Kursverwaltung

$$\begin{aligned} \text{Aus DB-Rel. KURSPLAN} \rightsquigarrow kp \in ePräd \\ vs(X, Y) : -kp(X, Y) \rightsquigarrow vs \in iPräd \end{aligned} \quad (11)$$

Gesucht: Durch Programm P bestimmte Grundatome (Fakten).

Definition: Grundatom

Ein Grundatom f ist eine logische folgerung einer Menge D von Datalog Klauseln (z.B. $D \models f$) \diamond_{Def} . Jedes Herbrand Modell von D ist auch ein Modell von f .

Da f ein Grundatom ist gilt $D \models f \implies f$ ist in jedem Herbrand-Modell von D enthalten. Das heißt $f \in \bigcap \{I \mid I \text{ Herbrand-Modell von } D\}$.

Sei $f \in \bigcap \{I \mid I \text{ Herbrand-Modell von } D\}$, dann ist f ein Grundatom und jedes Modell von D auch in Modell von f .

Definition: Menge aller Konsequenzen

$$cons(D) =_{def} \{f \in HB_D \mid D \models f\}$$

Satz 1.1

$$cons(D) = \bigcap \{I \mid I \text{ Herbrand-Modell von } D\}$$

Aufgrund der Eigenschaften unserer Regel ist der Schnitt / $cons(D)$ ein Modell, dies gilt es zu beweisen:

Da $cons(D) \subseteq HB_D$, ist $cons(D)$ eine Herbrand-Interpretation.

Satz 1.2

$cons(D)$ ist ein Herbrand-Modell von D .

Beweis

z.Z. Jedes $d \in D$ ist gültig in dieser Interpretation, also $cons(D) \models_{cons(D)} d$.

Beweis Falls d ein Grundatom ist gehört d zu jedem Herbrandmodell von D , daraus folgt $d \in cons(D)$.

Sei d eine Regel $q(\dots) : -p_1(\dots), \dots, p_m(\dots)$. Sei ϱ eine Belegung für die Variable von d . *Annahme:* $(\forall 1 \leq i \leq m)(\|p_i(\cdot)\|^{\varrho} \in cons(D))$, sonst d gültig unter $cons(D)$. Für jedes Herbrand-Modell I von D gilt, dann $\|p_i(\cdot)\|^{\varrho} \in I, i = 1, \dots, m$ und damit $\|q(\cdot)\|^{\varrho} \in cons(D)$, da d eine Horn-Klausel ist. (Der Schluss ist z.B. für $d = q_1(\dots), q_2(\dots) : -p_1(\dots), \dots, p_n(\dots)$ nicht möglich.)

Damit $D \models_{cons(D)} \|q(\dots)\|^{\varrho} \in cons(D)$. Wir erhalten, dass $cons(D)$ ein Herbrand-Modell von D ist. Da d beliebig gewählt wird folgt der Satz.

$cons(D)$ ist offensichtlich eindeutig bestimmt und das kleinste Herbrand-Modell von D . Damit: Semantik eines Datalog-Programms P ist gegeben durch das kleinste Herbrand-Modell von P oder (äquivalent) durch $cons(P) = \{f \in HB_P \mid O \models f\}$.

Beispiel $r \in ePräd, p, q \in iPräd, Konst = \{1, 2, 3\}$.

Fakten:

$$\begin{aligned} & r(1). \\ & p(X) : -q(X). \\ & q(X) : -r(X). \end{aligned} \tag{12}$$

Herbrand-Modelle:

$$\begin{aligned} & \{r(1), p(1), q(1)\} \\ & \{r(1), p(1), q(1), q(2), p(2)\} \\ & \{r(1), p(1), q(1), q(2), p(2), p(3)\} \end{aligned} \tag{13}$$

Fixpunkt-Semantik

Deklarative Semantik liefert keinen brauchbaren Algorithmus. Die Fixpunkt-Semantik führt direkt zu einem Algorithmus für die schrittweise Berechnung des kleinsten Herbrand-Modells.