

Kapitel 6: Logikorientierte Anfragesprachen

6.1 Grundlagen aus der Logik

6.1.1 Aussagenlogik

6.1.2 Prädikatenlogik 1. Ordnung

6.2 Domain-Relationenkalkül (DRK)

6.3 Tupel-Relationenkalkül (TRK)

6.4 Mächtigkeit relationaler Anfragesprachen

6.5 Datalog: Deduktionsregeln als Anfragesprache

6.1 Überblick: Dualismus der Logik

Syntax

(beweistheoretische Sicht):

Formeln F_1, \dots, F_n
mit Aussagenvariablen,
Prädikat-/Funktionssymbolen

Ableitungsregeln
(Beweis-/Deduktionsregeln):
 $F_1, \dots, F_n \vdash G$

Semantik

(modelltheoretische Sicht):

Interpretation ψ in
(mathematischer) Struktur S

Gültigkeit in S unter ψ
(G ist in S wahr):
 $\psi \models G$

idealerweise:

wenn G ableitbar, dann G wahr (Korrektheit)

wenn G wahr, dann G ableitbar (Vollständigkeit)

Aussagenlogik: Syntax

Definition:

Eine *atomare Formel* der Aussagenlogik ist eine Aussagenkonstante True oder False oder eine Aussagevariable der Form A_i ($i=1, 2, \dots$).

Die Menge der *Formeln* der Aussagenlogik ist:

- (i) Jede atomare Formel ist eine Formel.
- (ii) Wenn F, G Formeln sind, dann auch (F) , $\neg F$, $F \wedge G$, $F \vee G$ sowie $F \Rightarrow G$ (kurz für $\neg F \vee G$) und $F \Leftrightarrow G$ (kurz für $(F \wedge G) \vee (\neg F \wedge \neg G)$).

Notation und Präzedenzen:

Statt $\neg F$ schreibt man auch \overline{F} .

\neg bindet stärker als \wedge und \vee , und \wedge und \vee binden stärker als \Rightarrow , \Leftrightarrow .

Aussagenlogik: Semantik

Definition:

Sei F eine Formel der Aussagenlogik mit atomarer Formelmengemenge D .

Eine *passende Struktur S zu F* ist eine Menge P von Aussagen, so daß jede Variable in D einer dieser Aussagen zugeordnet werden kann.

Eine *Interpretation von F* ist eine Abbildung $\psi: D \rightarrow P$, für die gilt $\psi(\text{True})=1$, $\psi(\text{False})=0$, $\psi(A_i)=1$ (0), falls $\psi(A_i)$ in S wahr (falsch) ist.

ψ wird auf beliebige Formeln F , G , usw. über D fortgesetzt:

(i) $\psi((F)) = \psi(F)$

(ii) $\psi(F \wedge G) = 1$ falls $\psi(F) = \psi(G) = 1$, 0 sonst

(iii) $\psi(F \vee G) = 1$ falls $\psi(F) = 1$ oder $\psi(G) = 1$, 0 sonst

(iv) $\psi(\neg F) = 1$ falls $\psi(F) = 0$, 0 sonst.

Aussagenlogik: Modelle und Folgerungen

Definition:

Sei F eine Formel und ψ eine Interpretation von F .

Wenn $\psi(F)=1$ ist, heißt ψ *Modell von F* ($\psi \models F$ oder $\models_{\psi} F$).

F heißt *erfüllbar*, falls F mindestens ein Modell hat, sonst *unerfüllbar*.

F heißt *(allgemein-)gültig* oder *Tautologie*, falls jede Interpretation in einer zu F passenden Struktur ein Modell von F ist.

Satz: F ist Tautologie genau dann, wenn $\neg F$ unerfüllbar ist

Definition:

G heißt *Folgerung* von F_1, \dots, F_k ($F_1, \dots, F_k \models G$), wenn für jedes ψ gilt: falls ψ Modell von F_1, \dots, F_k ist, dann ist ψ auch Modell von G .

F und G heißen *äquivalent* ($F \equiv G$), falls für jedes ψ gilt: $\psi(F) = \psi(G)$.

Satz: G ist Folgerung von F_1, \dots, F_k g.d.w. $(F_1 \wedge F_2 \wedge \dots \wedge F_k) \Rightarrow G$ eine Tautologie ist. F und G sind äquivalent g.d.w. F Folgerung von G ist und umgekehrt.

Aussagenlogik: Beispiele

1) F1: $T \Rightarrow \neg P$, F2: $P \wedge \neg T$ bzw. F: $(T \Rightarrow \neg P) \wedge (P \wedge \neg T)$
 $\psi(T) = \text{„7 ist durch 2 teilbar“}$, $\psi(P) = \text{„7 ist eine Primzahl“}$

2) F: $(S \Rightarrow \neg R) \wedge (\neg R \Rightarrow S) \wedge (A \Rightarrow \neg S) \wedge (A)$
 $\psi(S) = \text{„die Sonne scheint“}$, $\psi(R) = \text{„es regnet“}$ (wahr),
 $\psi(A) = \text{„es ist April“}$

3) a) $\neg(\neg F) \Leftrightarrow F$
b) $(X \wedge Y) \vee (\neg X \wedge Z)$
c) $F \wedge \neg F$

Aussagenlogik: Deduktionskalkül

Ein einfacher Deduktionskalkül:

- (i) $\vdash F \Rightarrow (F \Rightarrow F)$
- (ii) $\vdash (F \Rightarrow (G \Rightarrow H)) \Rightarrow ((F \Rightarrow G) \Rightarrow (F \Rightarrow H))$
- (iii) $\vdash (\neg F \Rightarrow \neg G) \Rightarrow (G \Rightarrow F)$
- (iv) $F \Leftrightarrow G, H \quad \vdash H[F/G]$
- (v) $(v) F \Rightarrow G, F \quad \vdash G$

Definition:

Sei H eine endl. Formelmeng. F heißt aus H *ableitbar* ($H \vdash F$), wenn es eine endl. Sequenz F_0, F_1, \dots, F_n von Formeln gibt mit $F_n = F$, so daß für alle F_i gilt: F_i ist Element von H , F_i ist eine der Formeln (i) bis (iii) oder F_i ist aus F_0, \dots, F_{i-1} mittels (iv) oder (v) abgeleitet. Für $H = \emptyset$ heißt F *Theorem* des Deduktionskalküls.

Satz: Der einfache Deduktionskalkül ist korrekt und vollständig, d.h.:

$H \vdash F$ genau dann, wenn $H \models F$
(und $\vdash F$ genau dann, wenn $\models F$)

Aussagenlogik: alternative Kalküle bzw. Äquivalenzregeln

$\neg\neg F \Leftrightarrow F$	(Doppelnegation)
$F \wedge F \Leftrightarrow F$	(Idempotenz)
$F \vee F \Leftrightarrow F$	
$F \wedge G \Leftrightarrow G \wedge F$	(Kommutativität)
$F \vee G \Leftrightarrow G \vee F$	
$F \wedge (G \wedge H) \Leftrightarrow (F \wedge G) \wedge H$	(Assoziativität)
$F \vee (G \vee H) \Leftrightarrow (F \vee G) \vee H$	
$F \wedge (F \vee G) \Leftrightarrow F$	(Absorption)
$F \vee (F \wedge G) \Leftrightarrow F$	
$F \wedge (G \vee H) \Leftrightarrow (F \wedge G) \vee (F \wedge H)$	(Distributivität)
$F \vee (G \wedge H) \Leftrightarrow (F \vee G) \wedge (F \vee H)$	
$\neg (F \wedge G) \Leftrightarrow (\neg F \vee \neg G)$	(Gesetz von de Morgan)
$\neg (F \vee G) \Leftrightarrow (\neg F \wedge \neg G)$	
$F \vee 1 \Leftrightarrow 1$	(Tautologieregeln)
$F \wedge 1 \Leftrightarrow F$	
$F \vee 0 \Leftrightarrow F$	(Unerfüllbarkeitsregeln)
$F \wedge 0 \Leftrightarrow 0$	
$F \vee (\neg F) \Leftrightarrow 1$	(Tertium non datur)
$F \wedge (\neg F) \Leftrightarrow 0$	(Kontradiktion)
$((F \Leftrightarrow G) \wedge H) \Leftrightarrow ((F \Leftrightarrow G) \wedge H[F/G])$	(Substitution)

Beispiel für aussagenlogische Deduktion

„Worin besteht das Geheimnis Ihres Lebens?“:

Wenn ich kein Bier zu einer Mahlzeit trinke, dann habe ich immer Fisch.

Wenn ich Fisch und Bier zur selben Mahlzeit habe, verzichte ich auf Eiscreme.

Wenn ich Eiscreme habe oder Bier meide, dann rühre ich Fisch nicht an.

Modellierung:

Formel G: $(\neg B \Rightarrow F) \wedge ((F \wedge B) \Rightarrow \neg E) \wedge ((E \vee \neg B) \Rightarrow \neg F)$

Vereinfachung, sprich Deduktion (mit Unterstreichung als Negation):

$$\begin{aligned} G &\Leftrightarrow (B \vee F) \wedge (\underline{F} \wedge \underline{B} \vee \underline{E}) \wedge ((\underline{E} \vee \neg B) \vee \underline{F}) && \text{Def. Implikation} \\ &\Leftrightarrow (B \vee F) \wedge (\underline{F} \vee \underline{B} \vee \underline{E}) \wedge ((\underline{E} \wedge B) \vee \underline{F}) && \text{de Morgan} \\ &\Leftrightarrow (B \vee F) \wedge (\underline{F} \vee \underline{B} \vee \underline{E}) \wedge (\underline{E} \vee \underline{F}) \wedge (B \vee \underline{F}) && \text{Distributivität} \\ &\Leftrightarrow ((B \vee F) \wedge (B \vee \underline{F})) \wedge ((\underline{F} \vee \underline{B} \vee \underline{E}) \wedge (\underline{E} \vee \underline{F})) && \text{Komm., Assoz.} \\ &\Leftrightarrow ((B \vee (F \wedge \underline{F})) \wedge ((\underline{F} \vee \underline{E}) \vee (\underline{B} \wedge 0))) && \text{Distrib., Tautologie} \\ &\Leftrightarrow ((B \vee 0) \wedge ((\underline{F} \vee \underline{E}) \vee 0)) && \text{Kontradikt., Unerf.} \\ &\Leftrightarrow B \wedge (\underline{F} \vee \underline{E}) && \text{Unerfüllbarkeit} \end{aligned}$$

Fazit: Trinke zu jeder Mahlzeit Bier, und iss niemals Fisch mit Eiscreme!

Prädikatenlogik: Syntax

Definition:

Gegeben seien *Variablen* x_i , *Prädikatsymbole* P_i der Stelligkeit k_i und *Funktionssymbole* f_i der Stelligkeit l_i . *Terme* sind:

- (i) Jede Variable ist ein Term
- (ii) Wenn t_1, \dots, t_k Terme sind und f_i ein k -stelliges Funktionssymbol ist, dann ist auch $f_i(t_1, \dots, t_k)$ ein Term.

Eine *atomare Formel* hat die Form $P_i(t_1, \dots, t_k)$ mit einem k -stelligem Prädikatsymbol P_i und Termen t_1, \dots, t_k .

Formeln der Prädikatenlogik 1. Ordnung sind:

- (i) Jede atomare Formel ist eine Formel.
- (ii) Für Formeln F und G sind auch $\neg F$, $F \wedge G$, $F \vee G$ und (F) Formeln.
- (iii) Für eine Variable x_i und eine Formel F sind auch $\forall x_i (F)$ und $\exists x_i (F)$ Formeln (mit *gebundener* Variable x_i).

Notation und Präzedenzen:

Statt $\forall x \forall y \forall z (F)$ schreibt man auch $\forall x,y,z (F)$.

Junktoren binden stärker als Quantoren.

Alle gebundenen Variablen sind paarweise verschieden.

Prädikatenlogik: Semantik (1)

Definition:

Geg. sei eine Menge von Funktions- und Prädikatsymbolen mit entspr. Stelligkeiten. Eine passende *Struktur* ist ein Tripel $S = (U, fun, pred)$ mit

- einem *Universum* (einer *Trägermenge*) U von Individuen,
- einer Menge fun von *Funktionen* $f_i: U \times \dots \times U \rightarrow U$ der Stelligkeit l_i , so daß für jedes l_i -stellige Symbol eine entspr. Funktion existiert
- einer Menge $pred$ von *Prädikaten* $P_i: U \times \dots \times U \rightarrow \{0,1\}$ der Stelligkeit k_i , so daß für jedes k_i -stellige Symbol ein entspr. Prädikat existiert.

Eine *Interpretation* ψ einer Formel F in einer passenden Struktur $S = (U, fun, pred)$ ist eine Abbildung, die U festlegt und jedem Funktions- bzw. Prädikatsymbol in F eine Funktion aus fun bzw. ein Prädikat aus $pred$ zuordnet.

Prädikatenlogik: Semantik (2)

ψ wird auf beliebige Formeln F , G , usw. und Terme fortgesetzt:

(i) $\psi((F)) = \psi(F)$

(ii) $\psi(F \wedge G) = 1$ falls $\psi(F) = \psi(G) = 1$, 0 sonst

(iii) $\psi(F \vee G) = 1$ falls $\psi(F) = 1$ oder $\psi(G) = 1$, 0 sonst

(iv) $\psi(\neg F) = 1$ falls $\psi(F) = 0$, 0 sonst

(v) $\psi(f(t_1, \dots, t_k)) = \psi(f)(\psi(t_1), \dots, \psi(t_k))$

für ein Funktionssymbol f und Terme t_1, \dots, t_k ,

(vi) $\psi(x) = a$ mit einem beliebigen $a \in U$ für eine (freie) Variable x

(vii) $\psi(P(t_1, \dots, t_k)) = \psi(P)(\psi(t_1), \dots, \psi(t_k))$

für ein Prädikatsymbol P und Terme t_1, \dots, t_k ,

(viii) $\psi(\forall x (F)) = 1$ falls $\psi(F[x/a]) = 1$ für alle $a \in U$

(ix) $\psi(\exists x (F)) = 1$ falls $\psi(F[x/a]) = 1$ für ein beliebiges $a \in U$.

Prädikatenlogik: Beispiele (1)

1) $F: \forall q \exists p \forall x, y (\text{greater}(p, q) \wedge ((\text{greater}(x, 1) \wedge \text{greater}(y, 1)) \Rightarrow (\neg \text{equal}(\text{mult}(x, y), p))))$

Struktur $S = (\mathbb{N}_0, \{+, \cdot\}, \{>, =\})$.

Interpretation ψ :

$\text{add} \mapsto +, \text{mult} \mapsto \cdot, \text{greater} \mapsto >, \text{equal} \mapsto =,$
 $q \mapsto 1, p \mapsto 2, x \mapsto 3, y \mapsto 4$

2) $F: K(1, \text{Lauer}, \text{Merzig}) \wedge K(2, \text{Schneider}, \text{Homburg}) \wedge P(1, \text{Papier}) \wedge B(7, 16, 1, 1, 100)$

Struktur: Datenbank mit dem Schema

Kunden (KNr, Name, Stadt),

Produkte (PNr, Bez),

Bestellungen (Monat, Tag, KNr, PNr, Menge).

über dem Universum $\mathbb{N}_0 \cup \Sigma^*$

Interpretation $\psi (F) = 1$, falls es die vier Tupel in der Datenbank gibt,
0 sonst.

Erweiterungen der Prädikatenlogik

Mehrsortige Logik mit mehreren Trägermengen (Sorten) und einer Signatur für die Prädikate und Funktionen.

Beispiel: $U = \{U_1, U_2\}$ mit $U_1 = \mathbb{R}^n$, $U_2 = \mathbb{R}$ und

Addition: $U_1 \times U_1 \rightarrow U_1$, Multiplikation: $U_1 \times U_2 \rightarrow U_1$,

Skalarprodukt: $U_1 \times U_1 \rightarrow U_2$,

Vektorprodukt: $U_1 \times U_1 \rightarrow U_1$, Nullvektor: $\rightarrow U_1$,

Prädikaten Orthogonal: $U_1 \times U_1 \rightarrow \{0,1\}$ usw.

Prädikate als Argumente von Prädikaten oder Funktionen,

Quantifizierung von Prädikaten oder Funktionen

→ *Prädikatenlogik höherer Ordnung*

Beispiel:

$$\begin{aligned} \forall x,y (H(x,y) \Leftrightarrow \forall P ((R(x,y) \Rightarrow P(x,y)) \wedge \\ \forall u,w,z (P(u,w) \wedge P(w,z) \Rightarrow P(u,z))) \\ \Rightarrow (H(x,y) \Rightarrow P(x,y))) \end{aligned}$$

Prädikatenlogik: Modelle und Folgerungen

Definition:

Sei F eine Formel und ψ eine Interpretation von F .

Wenn $\psi(F)=1$ ist, heißt ψ *Modell von F* ($\psi \models F$ oder $\models_{\psi} F$).

F heißt *erfüllbar*, falls F mindestens ein Modell hat, sonst *unerfüllbar*.

F heißt *(allgemein-)gültig* oder *Tautologie*, falls jede Interpretation in einer zu F passenden Struktur ein Modell von F ist.

Satz: F ist Tautologie genau dann, wenn $\neg F$ unerfüllbar ist

Definition:

G heißt *Folgerung von F_1, \dots, F_k* ($F_1, \dots, F_k \models G$), wenn für jedes ψ gilt: falls ψ Modell von F_1, \dots, F_k ist, dann ist ψ auch Modell von G .

F und G heißen *äquivalent* ($F \equiv G$), falls für jedes ψ gilt: $\psi(F) = \psi(G)$.

Satz: G ist Folgerung von F_1, \dots, F_k g.d.w. $(F_1 \wedge F_2 \wedge \dots \wedge F_k) \Rightarrow G$ eine Tautologie ist. F und G sind äquivalent g.d.w. F Folgerung von G ist und umgekehrt.

Prädikatenlogik: Beispiele (2)

Formeln F_1, \dots, F_5 bzw. Formel $F = F_1 \wedge \dots \wedge F_5$:

$$F_1 = \forall x, y, z (G(f(f(x, y), z), f(x, f(y, z))))$$

$$F_2 = \forall x (G(f(x, e), x))$$

$$F_3 = \forall x, y (G(x, y) \Leftrightarrow G(y, x))$$

$$F_4 = \forall x \forall y \forall z (G(x, y) \wedge G(y, z) \Rightarrow G(x, z))$$

$$F_5 = \forall x \exists y (G(f(x, y), e))$$

Prädikatenlogik: Deduktionskalküle

alle aussagenlogischen Äquivalenzen

Wenn $F \Leftrightarrow G$ und F als Teilformel in H vorkommt, dann ist $H \Leftrightarrow H[F/G]$

$$\neg \forall (F) \Leftrightarrow \exists x (\neg F)$$

$$\neg \exists (F) \Leftrightarrow \forall x (\neg F)$$

$$(\forall x (F)) \wedge G \Leftrightarrow \forall x (F \wedge G) \quad \text{falls } x \text{ in } G \text{ nicht frei vorkommt}$$

$$(\forall x (F)) \vee G \Leftrightarrow \forall x (F \vee G) \quad \text{falls } x \text{ in } G \text{ nicht frei vorkommt}$$

$$(\exists x (F)) \wedge G \Leftrightarrow \exists x (F \wedge G) \quad \text{falls } x \text{ in } G \text{ nicht frei vorkommt}$$

$$(\exists x (F)) \vee G \Leftrightarrow \exists x (F \vee G) \quad \text{falls } x \text{ in } G \text{ nicht frei vorkommt}$$

$$(\forall x (F)) \wedge (\forall y (G)) \Leftrightarrow \forall x (F \wedge G[y/x]), \text{ falls } x \text{ in } G \text{ nicht frei vorkommt}$$

$$(\exists x (F)) \vee (\exists y (G)) \Leftrightarrow \exists x (F \vee G[y/x]), \text{ falls } x \text{ in } G \text{ nicht frei vorkommt}$$

$$\forall x (\forall y (F)) \Leftrightarrow \forall y (\forall x (F))$$

$$\exists x (\exists y (F)) \Leftrightarrow \exists y (\exists x (F))$$

$$\forall x (F) \Rightarrow F[x/a] \text{ für alle Konstanten } a$$

$$((\forall x (F)) \wedge (F \Rightarrow G)) \Rightarrow (\forall x (G))$$

Vollständige und korrekte Kalküle: Resolutionskalkül, Tableauekalkül

Beispiel für prädikatenlogische Deduktion (1)

1. *i) Informatikstudenten können programmieren.*
2. *ii) Studenten mit guten Mathematiknoten studieren Informatik.*
3. *iii) Studenten, die mit einem Informatiker verwandt sind,*
4. *haben gute Mathematiknoten.*
5. *iv) Alle saarländischen Studenten sind mit Heinz Becker verwandt.*
6. *v) Die Verwandtschaftsbeziehung ist symmetrisch.*
7. *vi) Heinz Becker hat Informatik studiert.*

Universum: Menge aller Studenten

Prädikate:

KannProgrammieren $P(x)$

StudiertInformatik $I(x)$

HatGuteMathenoten $M(x)$

SindVerwandt $V(x,y)$

IstSaarländer $S(x)$

Funktionen und Konstanten:

HeinzBecker $hb()$

Beispiel für prädikatenlogische Deduktion (2)

Formeln:

1. i) $\forall x (I(x) \Rightarrow P(x)) \wedge$
2. ii) $\forall x (M(x) \Rightarrow I(x)) \wedge$
3. iii) $\forall x \forall y ((V(x,y) \wedge I(y)) \Rightarrow M(x)) \wedge$
4. iv) $\forall x (S(x) \Rightarrow V(x,hb)) \wedge$
5. v) $\forall x \forall y ((V(x,y) \Leftrightarrow V(y,x)) \wedge$
6. vi) $I(hb)$

Vereinfachung, sprich Deduktion:

(iii) \wedge (iv) \wedge (vi):

$$\begin{array}{l} \forall x \forall y (I(hb) \wedge (S(x) \Rightarrow V(x,hb)) \wedge (V(x,y) \wedge I(y)) \Rightarrow M(x))) \\ \text{—} \forall x (I(hb) \wedge (S(x) \Rightarrow V(x,hb)) \wedge (V(x,hb) \wedge I(hb)) \Rightarrow M(x))) \\ \text{—} \forall x (S(x) \Rightarrow M(x)) (*) \end{array}$$

(*) \wedge (ii) \wedge (i):

$$\begin{array}{l} \forall x ((S(x) \Rightarrow M(x)) \wedge (M(x) \Rightarrow I(x)) \wedge (I(x) \Rightarrow P(x))) \\ \text{—} \forall x (S(x) \Rightarrow P(x)) \end{array}$$

Fazit: alle Saarländer können programmieren!

6.2 Domain-Relationenkalkül (DRK)

Anfragen sind prädikatenlogische Mengenspezifikationen der Form

$$\{\langle x_1, \dots, x_n \rangle \mid F(x_1, \dots, x_n)\}$$

wobei F ein prädikatenlogischer Ausdruck über einer Menge von **Domain-Variablen** ist mit x_1, \dots, x_n als einzigen freien Variablen.

Die Menge der zulässigen prädikatenlogischen Ausdrücke F ist:

- 1) Für Variablen x_1, \dots, x_n und eine Relation R mit n Attributen ist $\langle x_1, \dots, x_n \rangle \in R$ (auch $R(x_1, \dots, x_n)$) ein zulässiger Ausdruck.
- 2) Für Variablen x, y , Konstanten c und Vergleichsoperationen $\theta \in \{=, \neq, <, >, \leq, \geq\}$ sind $x \theta y$ und $x \theta c$ zulässige Ausdrücke.
- 3) Falls F_1 und F_2 zulässige Ausdrücke sind, dann sind auch $F_1 \wedge F_2$, $F_1 \vee F_2$, $\neg F_1$ und (F_1) zulässig.
- 4) Falls F ein zulässiger Ausdruck mit einer freien Variable x ist, dann sind auch $\exists x: F(x)$ und $\forall x: F(x)$ zulässige Ausdrücke.
- 5) Falls F ein zulässiger Ausdruck ist, dann ist auch (F) zulässig.
- 6) Nur die durch 1) bis 5) erzeugten Ausdrücke sind zulässig.

Semantik des DRK

Eine *Datenbank* über einem Schema mit Relationen $R_1(A_{11}, \dots, A_{1n_1}), \dots, R_m(A_{m1}, \dots, A_{mn_m})$ ist eine Formelmengens H , die für jede Relation R_i ein Prädikatsymbol R_i verwendet und für jedes Tupel $\langle a_1, \dots, a_{n_i} \rangle \in \text{val}(R_i)$ eine atomare Formel $R_i(a_1, \dots, a_{n_i})$ enthält, bzw. ein Modell von H .

Das Ergebnis einer *Anfrage* $\{\langle x_1, \dots, x_n \rangle \mid F(x_1, \dots, x_n)\}$ über der Datenbank H ist:

$\{\langle a_1, a_2, \dots, a_n \rangle \mid a_1, a_2, \dots, a_n \in U \text{ und } H \models F[x_1/a_1, x_2/a_2, \dots, x_n/a_n]\}$.

Sonderfall:

wenn F keine freien Variablen enthält, ist das Ergebnis

1 wenn $H \models F$ und 0 sonst.

Beispiele für DRK-Anfragen (1)

- 1) Finden Sie (die Namen) alle(r) Kunden mit negativem Saldo.
→ $\{ \langle k, n, st, sa, r \rangle \mid \langle k, n, st, sa, r \rangle \in \text{Kunden} \wedge sa < 0.0 \}$ bzw.
→ $\{ \langle n \rangle \mid \exists k, st, sa, r: \langle k, n, st, sa, r \rangle \in \text{Kunden} \wedge sa < 0.0 \}$
- 2) Finden Sie die Namen aller Kunden, die eine unbezahlte Bestellung haben, die vor Anfang Oktober erfolgte.
→ $\{ \langle n \rangle \mid \exists k, st, sa, r: \langle \underline{k}, n, st, sa, r \rangle \in \text{Kunden} \wedge$
 $\exists b, m, t, p, me, su, stat: \langle b, m, t, \underline{k}, p, me, su, stat \rangle \in \text{Best.} \wedge$
 $m < 10 \wedge stat \neq \text{'bezahlt'} \}$
- 3) Namen der Homburger Kunden, die seit Sept. ein Prod. aus Homburg geliefert bekommen haben, jeweils mit der Bez. des Produkts.
→ $\{ \langle n, bez \rangle \mid \exists k, st, sa, r: \langle \underline{k}, n, st, sa, r \rangle \in \text{Kunden} \wedge$
 $\exists p, g, pr, l, v: \langle \underline{p}, bez, g, pr, l, v \rangle \in \text{Produkte} \wedge$
 $\exists b, m, t, me, su, stat: \langle b, m, t, \underline{k}, \underline{p}, me, su, stat \rangle \in \text{Best.} \wedge$
 $st = \text{'Homburg'} \wedge mw9 \wedge l = \text{'Homburg'} \}$

Beispiele für DRK-Anfragen (2)

- 4) Kunden, von denen mindestens eine Bestellung registriert ist.
→ $\{ \langle k, n, st, sa, r \rangle \mid \langle \underline{k}, n, st, sa, r \rangle \in \text{Kunden} \wedge$
 $\exists b, m, t, p, me, su, stat: \langle b, m, t, \underline{k}, p, me, su, stat \rangle \in \text{Best.} \}$
- 5) Kunden, von denen keine Bestellung registriert ist.
→ $\{ \langle k, n, st, sa, r \rangle \mid \langle k, n, st, sa, r \rangle \in \text{Kunden} \wedge$
 $\forall b, m, t, k', p, me, su, stat:$
 $\langle b, m, t, k', p, me, su, stat \rangle \in \text{Best.} \Rightarrow k \neq k' \}$
- 6) Kunden, die alle überhaupt lieferbaren Produkte bestellt haben.
→ $\{ \langle k, n, st, sa, r \rangle \mid \langle \underline{k}, n, st, sa, r \rangle \in \text{Kunden} \wedge$
 $\forall p: (\exists bez, g, pr, l, v: \langle \underline{p}, bez, g, pr, l, v \rangle \in \text{Produkte})$
 \Rightarrow
 $(\exists b, m, t, me, su, stat: \langle b, m, t, \underline{k}, \underline{p}, me, su, stat \rangle$
 $\in \text{Bestellungen}) \}$

Anfrage-GUI Query-by-Example

Finden Sie die Namen der Homburger Kunden, die seit Anfang Sept. ein Produkt aus Homburg oder Saarbrücken geliefert bekommen haben, dessen Preis über 100 DM liegt, jeweils mit der Bez. des Produkts.

Kunden

KNr	Name	Stadt	Saldo	Rabatt
_55	.print	Homburg		

Bestellungen

BestNr	Monat	Tag	KNr	PNr	Menge	Summe	Status
	≥ 9		_55	_33			

Produkte

PNr	Bez	Gewicht	Preis	Lagerort	Vorrat
_33	.print		> 100	Homburg Saarbrücken	

6.3 Tupel-Relationenkalkül (TRK)

Anfragen sind prädikatenlogische Mengenspezifikation der Form

$$\{t \mid F(t)\} \text{ bzw. } \langle t1.A1, \dots, tn.An \rangle \mid F(t1, \dots, tn)\},$$

wobei F eine prädikatenlogische Formel über einer Menge von **Tupelvariablen** ist mit t bzw. $t1, \dots, tn$ als einzigen freien Variablen und $A1, \dots, An$ Attributnamen sind.

Die Menge der zulässigen prädikatenlogischen Ausdrücke F ist:

- 1) Für eine Variable r und eine Relation R ist $r \in R$ (auch $R(r)$) ein zulässiger Ausdruck.
- 2) Für Var. r, s , Attr. A, B mit $\text{dom}(A)=\text{dom}(B)$, Konst. $c \in \text{dom}(A)$ und Vergleichsoperationen $\theta \in \{=, \neq, <, >, \leq, \geq\}$ sind $r.A \theta s.B$ und $r.A \theta c$ zulässige Ausdrücke.
- 3) Falls $F1$ und $F2$ zulässige Ausdrücke sind, dann sind auch $F1 \wedge F2$, $F1 \vee F2$, $\neg F1$ und $(F1)$ zulässig.
- 4) Falls F ein zulässiger Ausdruck mit einer freien Tupelvariable r ist, dann sind auch $\exists r: F(r)$ und $\forall r: F(r)$ zulässige Ausdrücke.
- 5) Falls F ein zulässiger Ausdruck ist, dann ist auch (F) zulässig.
- 6) Nur die aufgrund von 1) bis 5) erzeugten Ausdrücke sind zulässig.

Semantik des TRK

Übersetzung in den DRK:

In einer Anfrage der Form $\{t / F(t)\}$ oder $\{<t.B1, \dots, t.Bn> / F(t)\}$ wird

- jeder Term der Form $r.Ai$ mit einer Tupelvariablen r durch eine Domain-Variable ra_i ersetzt und
- jeder Term der Form $r \in R$
über einer Relation mit Schema $R(A1, \dots, Am)$
durch einen Term $R(ra_1, \dots, ra_m)$ mit Domain-Variablen ra_1, \dots, ra_m .

Beispiele für TRK-Anfragen (1)

- 1) Finden Sie (die Namen) alle(r) Kunden mit negativem Saldo.
→ $\{t \mid t \in \text{Kunden} \wedge t.\text{Saldo} < 0.0\}$ bzw.
 $\{t.\text{Name} \mid t \in \text{Kunden} \wedge t.\text{Saldo} < 0.0\}$
- 2) Finden Sie die Namen aller Kunden, die eine unbezahlte Bestellung haben, die vor Anfang Oktober erfolgte.
→ $\{t.\text{Name} \mid t \in \text{Kunden} \wedge \exists b: b \in \text{Bestellungen} \wedge$
 $t.\text{KNr}=b.\text{KNr} \wedge b.\text{Monat} < 10 \wedge b.\text{Status} \neq \text{'bezahlt'}\}$
- 3) Namen der Homburger Kunden, die seit Sept. ein Prod. aus Homburg geliefert bekommen haben, jeweils mit der Bez. des Produkts.
→ $\{k.\text{Name}, p.\text{Bez} \mid k \in \text{Kunden} \wedge p \in \text{Produkte} \wedge$
 $\exists b: b \in \text{Bestellungen} \wedge$
 $k.\text{Stadt}=\text{'Homburg'} \wedge b.\text{Monat} \geq 9 \wedge p.\text{Lagerort}=\text{'Homburg'} \wedge$
 $k.\text{KNr}=b.\text{KNr} \wedge b.\text{PNr}=p.\text{PNr}\}$

Beispiele für TRK-Anfragen (2)

- 4) Kunden, von denen mindestens eine Bestellung registriert ist.
→ $\{t \mid t \in \text{Kunden} \wedge \exists b: b \in \text{Bestellungen} \wedge b.\text{KNr}=t.\text{KNr}\}$
- 5) Kunden, von denen keine Bestellung registriert ist.
→ $\{t \mid t \in \text{Kunden} \wedge \neg(\exists b: b \in \text{Bestellungen} \wedge b.\text{KNr}=t.\text{KNr})\}$
oder
 $\{t \mid t \in \text{Kunden} \wedge \forall b: b \in \text{Bestellungen} \Rightarrow b.\text{KNr} \neq t.\text{KNr}\}$
- 6) Kunden, die alle überhaupt lieferbaren Produkte bestellt haben.
→ $\{t \mid t \in \text{Kunden} \wedge \forall p: p \in \text{Produkte} \Rightarrow$
 $(\exists b: b \in \text{Bestellungen} \wedge b.\text{PNr}=p.\text{PNr} \wedge$
 $b.\text{KNr}=t.\text{KNr})\}$

Domain-unabhängige Anfragen

Problem: Anfragen liefern u.U. unendliche Resultatmengen.

Beispiel: $\{t \mid \neg(t \in R)\}$ für eine endliche Relation R.

Lösungsidee:

Eine Anfrage des TRK heißt *domain-unabhängig* g.d.w. für jede mögliche Ausprägung der Datenbank die Resultatmenge von der Datenbank, nicht aber von den zugrundeliegenden Domains abhängt.

Wahl des Universums U bei der Interpretation einer TRK-Formel F:

- *unbeschränkte Interpretation:*

U besteht aus der Vereinigung aller *Domains* der Datenbank

- *beschränkte Interpretation:*

U besteht aus dem *aktiven Domains* der Datenbank und der Formel F

Definition:

F heißt *domain-unabhängig* g.d.w. ihre unbeschränkte Interpretation für jeden Domain Dom, der den aktiven Domain ADom umfasst, mit ihrer beschränkten Interpretation übereinstimmt.

Domain-unabhängige Anfragen: Beispiele

Sei R eine Relation mit $\text{sch}(R)=\{A\}$, $\text{dom}(A)=\{1,2\}$, $\text{val}(R)=\{\langle 1 \rangle\}$.

Der Ausdruck $\{t \mid \neg (t \in R)\}$ hat

als unbeschränkte Interpretation den Wert $\{\langle 2 \rangle\}$ und
als beschränkte Interpretation den Wert \emptyset .

Der Ausdruck $\{t \mid \exists x : \neg (x=t) \wedge x.A=1\}$ hat

als unbeschränkte Interpretation den Wert $\{\langle 2 \rangle\}$ und
als beschränkte Interpretation den Wert \emptyset .

Der Ausdruck $\{t \mid t \in R \wedge \forall x : x=t\}$ hat

als unbeschränkte Interpretation den Wert \emptyset und
als beschränkte Interpretation den Wert $\{\langle 1 \rangle\}$.

Sichere Anfragen: Idee

$F(x)$ mit Var. x heißt *beschränkt*, wenn für jede Interpretation von F gilt:

- wenn $F(x)$ wahr ist,
dann liegt die Interpretation von x in $ADom$ von F .

$F(x)$ mit Var. x heißt *unbeschränkt*, wenn für jede Interpret. von F gilt:

- wenn die Interpretation von x nicht in $ADom$ von F liegt,
muß $F(x)$ wahr sein.

Durch syntaktische Einschränkungen von Formeln können

- **freie Variablen beschränkt** werden
(so daß in $\{t \mid F(t)\}$ $F(t)$ nur für t aus $ADom$ erfüllbar ist),
- **durch Existenzquantoren gebundene Variablen beschränkt** werden
(so daß in $\exists x : G(x)$ $G(x)$ nur für x aus $ADom$ erfüllbar ist) und
- **durch Allquantoren gebundene Variablen unbeschränkt** werden
(so daß in $\forall x : G(x)$ $G(x)$ für Interpret. von x , die nicht im $ADom$ liegen, immer wahr ist).

Sichere Anfragen: Syntaktische Einschränkung

Definition:

Sei $F(t)$ eine Formel des TRK mit freier Variable t . Zu jeder Var. in F läßt sich ein Schema aus F und dem Datenbankschema ableiten.

Die **Beschränktheit** und die **Unbeschränktheit** der Attr. von Var. in F sind:

- 1) In $t \in R$ sind alle Attribute von t beschränkt.
- 2) In $t.A=c$ mit einer Konstanten c ist $t.A$ beschränkt.
- 3) In $r.A=s.B \wedge F$ ist $r.A$ beschränkt, wenn $s.B$ in F beschränkt ist, und $s.B$ beschränkt, wenn $r.A$ in F beschränkt ist..
- 4a) In $F \wedge G$ ist $t.A$ beschränkt g.d.w. $t.A$ in F oder in G beschränkt ist.
- 4b) In $F \wedge G$ ist $t.A$ unbeschränkt g.d.w. $t.A$ in F und in G unbeschr. ist.
- 5a) In $F \vee G$ ist $t.A$ beschränkt g.d.w. $t.A$ in F und in G beschränkt ist.
- 5b) In $F \vee G$ ist $t.A$ unbeschränkt g.d.w. $t.A$ in F oder in G unbeschr. ist.
- 6a) In $\neg F$ ist $t.A$ beschränkt g.d.w. $t.A$ in F unbeschränkt ist.
- 6b) In $\neg F$ ist $t.A$ unbeschränkt g.d.w. $t.A$ in F beschränkt ist.
- 7a) In $\exists x: F(x)$ ist $t.A$ beschränkt g.d.w. $t.A$ in F beschränkt ist.
- 7b) In $\exists x: F(x)$ ist $t.A$ unbeschränkt g.d.w. $t.A$ in F unbeschränkt ist.
- 8a) In $\forall x: F(x)$ ist $t.A$ beschränkt g.d.w. $t.A$ in F beschränkt ist.
- 8b) In $\forall x: F(x)$ ist $t.A$ unbeschränkt g.d.w. $t.A$ in F unbeschränkt ist.

Sichere und unsichere Anfragen: Beispiele

Sei R eine Relation mit $\text{sch}(R)=\{A,B\}$, $\text{val}(R)=\langle 2,2 \rangle$.

$$\{t \mid \neg(t \in R) \vee t.A=2 \vee t.B=2\}$$

$$\{t \mid \neg(t \in R) \wedge t.A=2\}$$

$$\{t \mid \neg(t \in R) \wedge t.A=2 \wedge t.B=2\}$$

$$\{t \mid \exists s: (s \in R \wedge s.A=t.A) \}$$

$$\{t \mid \exists s: (s \in R \vee s=t) \}$$

$$\{t \mid \exists s: (s \in R \wedge s=t) \}$$

$$\{t \mid \neg(\exists s: (s \in R \wedge s=t)) \}$$

$$\{t \mid \forall s: (s \in R \wedge s=t) \}$$

$$\{t \mid \forall s: (s \in R \Rightarrow s=t) \}$$

$$\{t \mid \forall s: (s \in R \Rightarrow (s=t \wedge t.A=2 \wedge t.B=2)) \}$$

$$\{t \mid t \in R \wedge \forall s: (s \in R \Rightarrow (s=t \wedge t.A=2 \wedge t.B=2)) \}$$

$$\{t \mid \forall s: (s \in R \wedge (s=t \wedge t.A=2 \wedge t.B=2)) \}$$

$$\{t \mid \forall s: (s \in R \Rightarrow \neg(s=t)) \}$$

$$\{t \mid t \in K \wedge \forall p: p \in P \Rightarrow (\exists b: b \in B \wedge b.PNr=p.PNr \wedge b.KNr=t.KNr) \}$$

Rezept für sichere Anfragen

Satz:

Eine Anfrage des TRK ist sicher

und damit garantiert domain-unabhängig, wenn

- die Anfrage die Form $\{t \mid t \in R \wedge F(t)\}$ hat und.
- jede existenzquantifizierte Teilformel die Form $\exists x: x \in R \wedge F(t)$ hat und
- jede allquantifizierte Teilformel die Form $\forall x: x \in R \Rightarrow F(x)$ hat.

6.4 Mächtigkeit relationaler Anfragesprachen

Satz:

- Alle Anfragen, die sich mit der **RA** ausdrücken lassen, lassen sich auch mit dem **sicheren TRK** oder dem **sicheren DRK** ausdrücken.
- Alle Anfragen, die sich mit dem **sicheren TRK** ausdrücken lassen, lassen sich auch mit der **RA** oder dem **sicheren DRK** ausdrücken.
- Alle Anfragen, die sich mit dem **sicheren DRK** ausdrücken lassen, lassen sich auch mit der **RA** oder dem **sicheren TRK** ausdrücken.

6.5 Datalog: Deduktionsregeln als Anfragesprache

Extensionale Repräsentation von Information: Fakten, Tabellen

Intensionale Repräsentation von Information: Regeln, Ableitungen

Beispiel:

intensionale Repräsentation von Flugverbindungen (V)

auf der Grundlage einer extensionalen Repräsentation von Flügen (F)

als Formeln des DRK:

$$\forall a, z: F(a, z) \Rightarrow V(a, z)$$

$$\forall a, z, o: V(a, o) \wedge F(o, z) \Rightarrow V(a, z)$$

als Fixpunktgleichung in der RA:

$$V = F \cup (V \mid x \mid [V.Zielort=F.Abflugort] F)$$

Datalog: Definitionen (1)

Eine *Hornklausel* über einer endlichen Menge von Prädikatsymbolen ist eine logische Formel der Form

$$\forall X_1, \dots, X_n: P_1(Z_{11}, Z_{12}, \dots, Z_{1k_1}) \wedge \dots \wedge P_m(Z_{m1}, Z_{m2}, \dots, Z_{mk_m}) \Rightarrow P_0(Z_{01}, Z_{02}, \dots, Z_{0k_0})$$

mit k_i -stelligen Prädikaten P_i , Variablen X_1, \dots, X_n und Konstanten oder Variablen Z_{ij} .

Die Formel

$$P_0(Z_{01}, Z_{02}, \dots, Z_{0k_0})$$

wird als "**Kopf**" (engl.: head) der Hornklausel bezeichnet, die Formel

$$P_1(Z_{11}, Z_{12}, \dots, Z_{1k_1}) \wedge \dots \wedge P_m(Z_{m1}, Z_{m2}, \dots, Z_{mk_m})$$

als "**Rumpf**" (engl.: body).

Datalog: Definitionen (2)

Ein *Datalog-Programm* besteht aus

- einer Menge von Fakten der Form $P_i(V_1, V_2, \dots, V_k)$ mit einem k -stelligen Prädikat P_i und Konstanten V_1, \dots, V_k und
- einer Menge von Regeln in Form von Hornklauseln.

Ein *Datalog-Programm mit Negation* besteht aus

- einer Menge von Fakten und einer Menge von
- Regeln in Form von Klauseln, bei denen Rumpfprädikate negiert sein können.

Eine *Anfrage* (Ziel; engl: goal) ist ein Ausdruck der Form $P(Z_1, \dots, Z_k)$ mit einem k -stelligen Prädikat P und Konstanten oder Variablen Z_1, \dots, Z_k , so daß mindestens eines der Z_j eine (freie) Variable ist.

Eine Regel r heißt *rekursiv*,

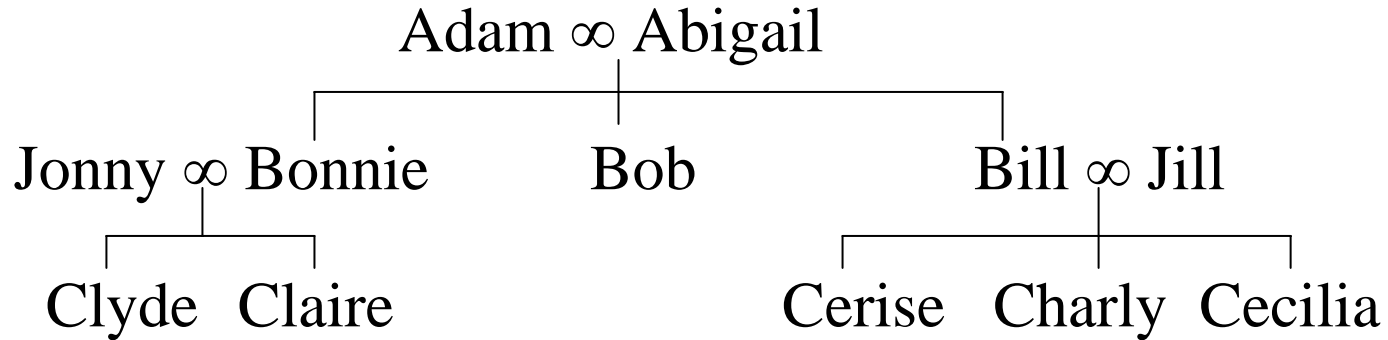
wenn es Regeln r_1, r_2, \dots, r_n gibt mit $r_1 = r_n = r$, so daß für alle i ein Rumpfprädikat von r_i im Kopf von r_{i+1} steht.

Mächtigkeit von Datalog

Satz:

- 1) Die Menge der Anfragen, die sich mit Datalog *ohne Rekursion und ohne Negation* ausdrücken lassen, ist eine echte Untermenge der Anfragen, die sich mit dem DRK ausdrücken lassen.
- 2) Die Menge der Anfragen, die sich mit Datalog *mit Negation, aber ohne Rekursion* ausdrücken lassen, ist identisch mit der Menge der Anfragen, die sich mit dem DRK ausdrücken lassen.
- 3) Die Menge der Anfragen, die sich mit Datalog *mit Rekursion und mit Negation* ausdrücken lassen, ist eine echte Obermenge der Menge von Anfragen, die sich mit dem DRK ausdrücken lassen.

Datalog-Beispiel: Fakten



Mann (Adam), Mann (Jonny), Mann (Bob), Mann (Bill), Mann (Clyde),
Mann (Charly), Frau (Abigail), Frau (Bonnie), Frau (Jill), Frau (Claire),
Frau (Cerise), Frau (Cecilia)

Ehepaar (Adam, Abigail), Ehepaar (Jonny, Bonnie), Ehepaar (Bill, Jill)

Elternteil (Adam, Bonnie), Elternteil (Adam, Bob), Elternteil (Adam, Bill),
Elternteil (Abigail, Bonnie), Elternteil (Abigail, Bob), Elternteil (Abigail, Bill),
Elternteil (Jonny, Clyde), Elternteil (Jonny, Claire),

Elternteil (Bonnie, Clyde), Elternteil (Bonnie, Claire),

Elternteil (Bill, Cerise), Elternteil (Bill, Charly), Elternteil (Bill, Cecilia),

Elternteil (Jill, Cerise), Elternteil (Jill, Charly), Elternteil (Jill, Cecilia)

SelbeGeneration (Adam, Abigail),

SelbeGeneration (Adam, Adam), SelbeGeneration (Abigail, Abigail)

SpieltMit (Clyde, Charly)

Datalog-Beispiel: Regeln

Elternteil (X, Y)	\Rightarrow Vorfahren (X, Y)
Elternteil (X, Y) \wedge Vorfahren (Y, Z)	\Rightarrow Vorfahren (X, Z)
Elternteil (X, Y) \wedge Mann (X)	\Rightarrow Vater (X, Y)
Elternteil (X, Y) \wedge Frau (X)	\Rightarrow Mutter (X, Y)
Elternteil (X, Y) \wedge Elternteil (X, Z) \wedge Y \neq Z	\Rightarrow Geschwister (Y, Z)
Elternteil (X, Y) \wedge Elternteil (U, W) \wedge Geschwister (X, U) \wedge Frau (W)	\Rightarrow Cousine (Y, W)
Elternteil (X, Y) \wedge Elternteil (U, W) \wedge SelbeGeneration (X, U)	\Rightarrow SelbeGeneration (Y, W)
Geschwister (X, Y)	\Rightarrow SpieltMit (X, Y)
SpieltMit (Clyde, Y)	\Rightarrow SpieltMit (Claire, Y)
TRUE	\Rightarrow SpieltMit (Cecilia, Y)
Elternteil (X, Y)	\Rightarrow Verwandt (X, Y)
Geschwister (X, Y)	\Rightarrow Verwandt (X, Y)
Verwandt (X, Y)	\Rightarrow Verwandt (Y, X)
Verwandt (X, Y) \wedge Verwandt (Y, Z)	\Rightarrow Verwandt (X, Z)
Mann (X) \wedge \neg Ehepaar (X, Y)	\Rightarrow Ledig (X)
Frau (Y) \wedge \neg Ehepaar (X, Y)	\Rightarrow Ledig (Y)