Logische Grundlagen von Informationssystemen

Gerd Wagner

February 5, 1996

Contents

Part I

Fakten und Tabellen

Chapter 1

Was ist ein Informationssystem ?

1.1 Grundbegriffe

Die aus den Grundbegriffen der Informatik gebildete Kette

$$\text{Daten} \longrightarrow \text{Information} \longrightarrow \text{Wissen}$$

drückt einen zunehmenden Grad an Abstraktheit aus.

Daten stammen u.a. aus Beobachtungen, Messungen und dokumentarischen Informationsquellen (wie Listen, Formulare, Fragebögen, etc.). Zum Beispiel stammen folgene Daten aus einem Flugticket: *27MAR, LLSX1M, WAGNER/GERDMR, BERLIN TXL.*

Informationen enthalten eine Nachricht, sie sagen etwas aus. Zum Beispiel: *"Das Datum des Abflugs ist der 27. März."*, oder *"Der Zieflughafen ist Berlin-Tegel."*. Die Zuordnung von einem Datum zu einem Attribut, wie z.B. *Tarifklasse = LLSX1M*, ergibt eine Information, d.h. eine informative Aussage. Es gibt aber auch Aussagen, die keine Information enthalten, z.B. *"Dieser Junggeselle ist unverheiratet."*, oder *"Gott ist allmächtig."*. Informative Aussagen beziehen sich auf einen konkreten Grundbereich. Sie sind die Basis für einfache Schlußfolgerungen. Informationen sind nicht immer (positiv oder negativ) kategorisch. Temporale Informationen sind durch einen Zeitbezug relativiert. Darüberhinaus gibt es graduell unsichere und (disjunktiv oder existentiell) unvollständige Informationen.

Wissen läßt sich in mathematisch-theoretisches, wissenschaftlich-empirisches und Alltagswissen unterscheiden. Empirisches und Alltagswissen kommt u.a. durch Akkumulation und Interpretation größerer Informationsmengen zustande. Gesetzmäßigkeiten, Begriffsbeziehungen und Handlungswissen werden oft in der Form von Regeln (oder All-Aussagen) ausgedrückt. Zum Beispiel drückt die folgende strikte Regel begriffliches Wissen aus:

Eine Nichte ist die Tochter einer Schwester oder eines Bruders.

Demgegenüber ist die dem Alltagswissen entstammende Regel

Studenten sind (normalerweise) nicht berufstätig.

anfechtbar, d.h. tolerant gegenüber Ausnahmen. Regelwissen ist die Basis für komplexe Schlußfolgerungen wie z.B. Erklärungen, Diagnosen, und Planung.

1.2 Grundfunktionen eines Informationssystems

1. Erfassung und Aktualisierung von Informationen aus einem bestimmten Bereich
2. Abfrage (Wiedergewinnung) von verfügbaren Informationen
3. Integritätswartung

Darüberhinaus muß ein Informationssystem vielen praktischen Anforderugen genügen: gleichzeitige Zugriffe mehrerer Benutzer müssen möglich sein, unerlaubte Zugriffe müssen durch geeignete Datenschutzmechanismen ausgeschlossen werden und auch bei technischen Störungen und Betriebsausfällen muß ein geordneter Wiederanlauf des Systems gewährleistet sein.

1.3 Was ist ein Wissenssystem ?

Ein Wissenssystem ist ein Informationssystem mit folgenden Erweiterungen:

1. Erfassung und Aktualisierung von Regeln (Deduktionsregeln, Aktionsregeln und Reaktionsregeln)
2. deduktive Anfragenbeantwortung
3. Erzeugung voon Erklärungen und Diagnosen
4. Erzeugung von Plänen

Chapter 2

Relationale Datenbanken

Eine relationale Datenbank ist eine endliche Menge von endlichen Relationen (Tabellen), die einer endlichen Menge von atomaren Aussagen entsprechen. Eine relationale Datenbank repräsentiert eine endliche prädikatenlogische Interpretation im Sinne der Modelltheorie.

2.1 Grundbegriffe

In der Mengentheorie ist eine n-stellige Relation R zwischen den Mengen $D_1, \ldots, D_n$ als eine Teilmenge ihres kartesischen Produkts, d.h. als eine Menge entsprechender Tupel, definiert:

$$R \subseteq D_1 \times \ldots \times D_n = \{\langle d_1, \ldots, d_n \rangle : d_i \in D_i \text{ für } i = 1, \ldots, n\}$$

Eine solche Relation kann als Tabelle dargestellt werden: die i-te Spalte der Tabelle entspricht dann der i-ten Stelle der Relation, jede Zeile der Tabelle entspricht einem Tupel aus der Relation.
Sinnvollerweise werden die Spalten benannt, z.B.:

NimmtTeilAn

StudName	Fach	VorlesgName
Steffi	Informatik	LogGrdlIS
Steffi	Informatik	Pascal
Sascha	Mathematik	Pascal
Dirk	Informatik	DB
Dirk	Informatik	LogGrdlIS

Jede Spalte der Tabelle entspricht einem **Attribut** der dargestellten Relation. Z.B. hat die Relation *NimmtTeilAn* drei Attribute:

$$\mathrm{Attr}(\mathit{NimmtTeilAn}) \quad = \quad \{\mathit{StudName}, \mathit{Fach}, \mathit{VorlesgName}\}$$

Jedes Attribut A hat einen Wertebereich $Dom(A)$.

Zu jeder Relation gibt es eine Funktion, die einem Attribut A und einem Tupel t den entsprechenden Wert $v_A(t) \in Dom(A)$, z.B.

$$v_{Fach}(3) = \text{“Mathematik”} \in \text{char}(30)$$

zuordnet. Das heißt tabellengrafisch: zu einer Spalte A liefert jede Zeile t einen entsprechenden Wert $v_A(t)$. Umgekehrt gilt natürlich auch, daß zu einer Zeile t jede Spalte A einen zugehörigen Wert festlegt, d.h. es gibt auch eine Funktion $t(A) := v_A(t)$. Diese Funktionen können sich auch auf eine Menge X von Attributen beziehen: $v_X(t)$ liefert dann ein entsprechendes Tupel als Wert, z.B. für $X = \{StudName, VorlesgName\}$ gilt

$$v_X(2) = \langle \text{Steffi}, \text{Pascal} \rangle$$

2.1.1 Schlüssel

Ein **Schlüssel** zu einer Relation R ist eine minimal identifizierende Menge von Attributen, d.h. eine minimale Menge $X \subseteq \text{Attr}(R)$, so daß

$$\forall t_1, t_2 \in R:\ v_X(t_1) = v_X(t_2)\ \Rightarrow\ t_1 = t_2$$

2.1.2 Relationsschema und Relationsinstanz

Ein **Basisrelationsschema** r besteht aus einer Menge von Attributen, $\text{Attr}(r) = \{A_1, \ldots, A_n\}$, und einem ausgezeichneten Schlüssel PK(r) (dem sog. *Primärschlüssel*):

$$r\ =\ \langle \text{Attr}(r),\ \text{PK}(r) \rangle$$

Ein Relationsschema wird auch vereinfachend als $r(A_1, \ldots, A_n)$ notiert.

Eine Relation R ist eine *Instanz* über einem Relationsschema $r(A_1, \ldots, A_n)$, symbolisch $R : r$, wenn

$$R \subseteq Dom(A_1) \times \ldots \times Dom(A_n)$$

2.1.3 Datenbankschema und Datenbankinstanz

Ein relationales **Datenbankschema** Σ besteht aus einer Menge von Relationsschemata $\{r_1, \ldots, r_m\}$ und einer Menge von Integritätsbedingungen IC:

$$\Sigma = \langle \{r_1, \ldots, r_m\},\ IC \rangle$$

Eine relationale **Datenbankinstanz** X über einem Schema Σ, symbolisch $X : \Sigma$, ist eine Menge von Relationsinstanzen über den gegebenen Relationsschemata:

$$X = \{R_1 : r_1, \ldots, R_m : r_m\}$$

X ist eine *zulässig* Instanz, wenn sie alle Integritätsbedingungen erfüllt, d.h. wenn

$$\forall\, F \in IC:\ X \vdash F$$

2.2 Eine Datenbank als Aussagenmenge

Jedes Tupel $\langle c_1, \ldots, c_n \rangle$ einer Relation $R : r$ entspricht einer *atomaren Aussage* $r(c_1, \ldots, c_n)$. Eine relationale Datenbank $\Delta = \langle \{R_1 : r_1, \ldots, R_m : r_m\}, IC \rangle$ kann daher durch die Vereinigung aller aus ihren Tabellen R_i gewonnenen Aussagenmengen repräsentiert werden:

$$A_\Delta = \bigcup_{i=1}^{m} \{r_i(\vec{c}) \mid \vec{c} \in R_i\}$$

Eine relationale Datenbank Δ repräsentiert also eine Menge von atomaren Aussagen: $A_\Delta \subseteq \mathrm{At}$.

Example 1 *Die Datenbank*

$$\Delta = \{R{:}r, S{:}s\} = \{ \boxed{\begin{array}{cc} a & c \\ b & c \end{array}}, \quad \begin{array}{c} \hline a \\ b \\ c \\ \hline \end{array} \}$$

entspricht der Aussagenmenge

$$A_\Delta = \{r(a,c),\ r(b,c),\ s(a), s(b), s(c)\}$$

2.2.1 Aktualisierung und Informationswachstum

Eine Datenbank wird mit Hilfe zweier Grundoperationen aktualisiert: zum einen werden neue atomare Informationen $a \in \mathrm{At}$ in die Datenbank eingefügt:,

$$\mathsf{Upd}(A_\Delta, a) := A_\Delta \cup \{a\},$$

zum anderen werden vorhandene Informationen gelöscht:

$$\mathsf{Upd}(A_\Delta, -a) := A_\Delta - \{a\}$$

Um den Wert eines Attributs zu ändern, werden Löschung und Einfügung kombiniert: zuerst wird der alte Wert gelöscht und dann der neue eingefügt. Wenn zum Beispiel Sascha sein Hauptfach wechselt und anstatt Mathematik nun Informatik wählt, ergibt sich folgende Atualisierung der Uni-Datenbank:

$$\Delta'_{\mathrm{Uni}} = \mathsf{Upd}(\Delta_{\mathrm{Uni}},\ -s(\mathit{Sascha}, \mathit{Mathe}) \wedge s(\mathit{Sascha}, \mathit{Informatik}))$$

Im Laufe der Zeit verändert sich der Inhalt, d.h. der Zustand, einer Datenbank. Man spricht auch von der *Evolution* einer Datenbank. In einigen Fällen wächst die Informationsmenge, in anderen nimmt sie ab. Der Prozeß des Informationswachstums wird durch die Definition einer Informationsordnung zwischen Datenbanken erfaßt. Eine Datenbank Δ' enthält mindestens soviel Information wie Δ, wenn die Aussagenmenge von Δ in der von Δ' enthalten ist:

$$\Delta \leq \Delta' \text{ wenn } A_\Delta \subseteq A_{\Delta'}$$

2.3 Anfragenbeantwortung

2.3.1 Datenbank-Abfragen als logische Formeln

Mit Hilfe von *Ob-Anfragen* kann man abfragen, ob etwas der Fall ist oder nicht. Ob-Anfragen entsprechen Aussagen, d.h. Formeln ohne freie Variablen. Die Menge der Ob-Anfragen wird mit L^0_{Query} bezeichnet. Wenn $F \in L^0_{\text{Query}}$ in Δ der Fall ist, schreibt man

$$\Delta \vdash F$$

Die in einer Datenbank gespeicherten Informationen werden mit Hilfe von Abfragen zurückgewonnen. Zum Beispiel entspricht die Abfrage

> *Wer nimmt an der Mathematikvorlesung "Analysis I" und an der Informatikvorlesung "Datenbanken" teil ?*

der Formel

$$nta(x, Math, AnaI) \wedge nta(x, Inf, DB)$$

mit einer freien Variablen x für die Namen von StudentInnen. Abfragen können also durch Formeln einer logischen Abfragesprache L_{Query} ausgedrückt werden. In den Formeln der Abfragsprache L_{Query} können die logischen Funktoren $\wedge$ (Konjunktion), $\vee$ (Disjunktion), $-$ (Negation) und $\exists$ (Existenzquantor) vorkommen. Eine Antwort auf eine Anfrageformel $F(x_1, \ldots, x_m)$ mit freien Variablen $x_1, \ldots, x_m$ ist ein Konstanten-Tupel $\langle c_1, \ldots, c_m \rangle$, dessen Einsetzung in F ableitbar ist:

$$\Delta \vdash F(c_1, \ldots, c_m)\}$$

Die Menge aller Antworten ist eine Relation:

$$\mathsf{Ans}(\Delta,\, F(x_1, \ldots, x_m)) = \{\langle c_1, \ldots, c_m \rangle \mid \Delta \vdash F(c_1, \ldots, c_m)\}$$

Eine Antwort kann auch als eine Substitution aufgefaßt werden:

$$\mathsf{Ans}(\Delta,\, F(x_1, \ldots, x_m)) = \{\{x_1/c_1, \ldots, x_m/c_m\} : \Delta \vdash F(c_1, \ldots, c_m)\}$$

2.3.2 Inferenz von Ob-Anfragen

Um abzufragen, welche Aussagen F auf der Basis einer Datenbank Δ gelten, bedient man sich der *natürlichen Inferenzrelation* $\vdash$, die induktiv über den Aufbau von Formeln definiert ist.

Für eine atomare Ob-Anfrage $r(c_1, \ldots, c_m)$ wird geprüft, ob die Information $r(c_1, \ldots, c_m)$ in der Datenbank enthalten ist:

$$(\vdash a) \quad \Delta \vdash r(c_1, \ldots, c_m) \text{ wenn } r(c_1, \ldots, c_m) \in A_\Delta$$

Wenn $R : r$ eine m-stellige Relation in Δ ist, dann ist

$$\mathsf{Ans}(\Delta, r(x_1, \ldots, x_m)) = R$$

$(\vdash a)$	$\Delta \vdash a$	wenn	$a \in A_\Delta$
$(\vdash -a)$	$\Delta \vdash -a$	wenn	$a \notin A_\Delta$
$(\vdash \wedge)$	$\Delta \vdash F \wedge G$	wenn	$\Delta \vdash F$ und $\Delta \vdash G$
$(\vdash \vee)$	$\Delta \vdash F \vee G$	wenn	$\Delta \vdash F$ oder $\Delta \vdash G$
$(\vdash \exists)$	$\Delta \vdash \exists x F(x)$	wenn	$\mathsf{Ans}(\Delta, F(x)) \neq \emptyset$
$(\vdash \forall)$	$\Delta \vdash \forall x F(x)$	wenn	$\mathsf{Ans}(\Delta, -F(x)) = \emptyset$

Diese induktive Definition wird durch die folgenden Ersetzungsregeln:

$$\begin{array}{lcl@{\qquad}lcl} -(F \vee G) & \longrightarrow & -F \wedge -G & --F & \longrightarrow & F \\ -(F \wedge G) & \longrightarrow & -F \vee -G & & & \end{array}$$

vervollständigt. Für die Datenbank Δ von Beispiel 1 erhalten wir

$$\mathsf{Ans}(\Delta,\ s(x) \wedge -\exists y r(x,y)) = \{x/c\}$$

da $\Delta \vdash s(c) \wedge -\exists y r(c,y)$.
Für Ob-Anfragen F ist die Antwort einer relationalen Datenbank Δ entweder **yes** oder **no** :

$$\mathsf{Ans}(\Delta, F) = \begin{cases} \text{yes} & \text{if } \mathrm{KB} \vdash F \\ \text{no} & \text{if } \mathrm{KB} \vdash -F \end{cases}$$

2.3.3 Nicht alle Formeln sind auswertbar

Eine Anfrage F kann nur dann sinnvoll beantwortet werden, wenn die Antwortmenge $\mathsf{Ans}(\Delta, F)$ endlich ist. Dies ist jedoch nicht bei jeder Anfrageformel der Fall.
Betrachten wir die Datenbank $\Delta_{\mathrm{Uni}} = \{\mathit{StudentIn} : s, \mathit{NimmtTeilAn} : \mathit{nta}\}$, wobei das Relationsschema s aus den Attributen *Name*, *Hauptfach* und *Semester* besteht und die zugehörige Tabelle folgendermaßen gefüllt ist:

StudentIn =

Dirk	Informatik	7
Steffi	Mathematik	6
Sascha	Mathematik	9

Dann erhalten wir für die Anfrage, welche StudentInnen studieren was im wievielten Semester, die (endliche) Basisrelation *StudentIn* selbst zur Antwort:

$$\mathsf{Ans}(\Delta_{\mathrm{Uni}},\ s(x,y,z)) = \mathit{StudentIn}$$

Für die – etwas merkwürdige – Anfrage, in welchem Semester Steffi nicht ist, ergibt sich die folgende unendliche Antwortmenge:

$$\mathsf{Ans}(\Delta_{\mathrm{Uni}},\ -s(\mathit{Steffi},\ \mathit{Mathematik},\ z)) = \{1,2,3,4,5,7,8,9,\ldots\}$$

Für die – vielleicht noch merkwürdigere – Anfrage, welches Fach Steffi studiert oder in welchem Semester Dirk ist, ergibt sich:

$$\mathsf{Ans}(\Delta_{\mathrm{Uni}},\ \exists zs(\mathit{Steffi},\ y,\ z) \vee \exists ys(\mathit{Dirk},\ y,\ z)) = \begin{array}{|cc|} \text{Mathematik} & 1 \\ \text{Mathematik} & 2 \\ \vdots & \vdots \\ \text{Mathematik} & n \\ \vdots & \vdots \\ \text{Informatik} & 7 \\ \text{Sinologie} & 7 \\ \vdots & \vdots \\ \mathit{irgendwas} & 7 \\ \vdots & \vdots \end{array}$$

Offensichtlich sind also gewisse negierte und disjunktive Formeln problematisch. Um solche absurden Anfrageformeln auszuschließen, definiert man die Menge der *auswertbaren* Formeln wie folgt.

(a) Atomare Formeln $r(x_1, \ldots, x_m)$ sind auswertbar.

Wenn F und G auswertbar sind, dann auch

$(\wedge)$	$F \wedge G$
$(\exists)$	$\exists x F$
$(\vee)$	$F \vee G$, wenn $\mathrm{Free}(F) = \mathrm{Free}(G)$
$(\wedge -)$	$F \wedge -G$, wenn $\mathrm{Free}(G) \subseteq \mathrm{Free}(F)$
$(\wedge \mathit{Cond})$	$F \wedge \mathit{Cond}$, wenn $\mathrm{Free}(\mathit{Cond}) \subseteq \mathrm{Free}(F)$

2.3.4 Anfragenbeantwortung mit Relationaler Algebra

Die Relationale Algebra ermöglicht die kompositionale Auswertung komplexer Anfragen, indem sie jedem logischen Hauptfunktor einer auswertbaren Formel eine algebraische Operation zuordnet. Zum Beispiel wird die disjunktive Anfrageformel "Wer ist im 6. oder 7. Semester ?" mit Hilfe der Vereinigungsoperation ausgewertet:

$$\begin{aligned} &\mathsf{Ans}(\Delta_{\mathrm{Uni}},\ \exists ys(x, y, 6) \vee \exists ys(x, y, 7)) \\ =\quad &\mathsf{Ans}(\Delta_{\mathrm{Uni}},\ \exists ys(x, y, 6))\ \cup\ \mathsf{Ans}(\Delta_{\mathrm{Uni}},\ \exists ys(x, y, 7)) \end{aligned}$$

Die fundamentalen Operationen der Relationalen Algebra sind: Vereinigung, Verbund, Differenz, Projektion und Selektion.

Vereinigung

Die Vereinigung zweier Tabellen ist definiert als

$$R \cup S = \{t \mid t \in R \ \text{ or } \ t \in S\}$$

Dabei wird vorausgesetzt, daß beide Tabellen kompatible Spalten haben. Die Vereinigungsoperation wird zur Auswertung disjunktiver Anfragen verwendet:

$$\mathsf{Ans}(\Delta, F \vee G) = \mathsf{Ans}(\Delta, F) \cup \mathsf{Ans}(\Delta, G)$$

Verbund

Die Verbundoperation kombiniert die Tupel der beteiligten Relationen:

$$R \bowtie S = \{t \mid t[R] \in R \ \ \& \ \ t[S] \in S\}$$

Hierbei bezeichnet $t[R]$ die Einschränkung des Tupels t auf die Attribute von R.
Die Verbundoperation dient zur Auswertung konjunktiver Anfragen:

$$\mathsf{Ans}(\Delta, F \wedge G) = \mathsf{Ans}(\Delta, F) \bowtie \mathsf{Ans}(\Delta, G)$$

Dabei werden gemeinsame freie Variablen in den Anfrageformeln F und G als gemeinsame Attribute interpretiert.

Differenz

Die Differenz zwischen R und S ist definiert als

$$R - S = \{t \in R \mid t[S] \notin S\}$$

Die Negation in Anfragen wird mit Hilfe der Differenz ausgewertet:

$$\mathsf{Ans}(\Delta, F \wedge -G) = \mathsf{Ans}(\Delta, F) - \mathsf{Ans}(\Delta, G)$$

Pojektion

Die Projektion schneidet ausgewählte Spalten aus einer Tabelle heraus:

$$\mathrm{Proj}(X, R) = \{t[X] \mid t \in R\}$$

Dabei bezeichnet $X \subseteq \mathrm{Attr}(R)$ die Menge der ausgewählten Attribute.
Die Projektion dient der Auswertung des Existenzquantors in Anfrageformeln:

$$\mathsf{Ans}(\Delta, \exists x_i F) = \mathrm{Proj}(\{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_m\}, \mathsf{Ans}(\Delta, F))$$

Ein Effekt der Projektion kann sein, daß sich die Anzahl der Tupel verringert, weil gewisse Tupel in den verbliebenenn Attributen übereinstimmen.

Selektion

Bedingungen an einzelne Tupel werden mit Hilfe einer Formel $Cond$ ausgedrückt, die aus einer Booleschen Kombination von Vergleichsausdrücken der Form $v \star w$ besteht, wobei v und w entweder Attribute/Variablen oder Konstanten sein können und $\star$ einer der üblchen Vergleichsoperatoren $=, <, >, \leq, \geq, \neq$ ist.

$$\mathrm{Sel}(Cond,\ R) = \{t \in R \mid t \text{ erfüllt } Cond\}$$

Die Selektion wird benötigt, um Anfrageformeln mit Zusatzbedingungen auszuwerten:

$$\mathsf{Ans}(\Delta, F \wedge Cond) = \mathrm{Sel}(Cond,\ \mathsf{Ans}(\Delta, F))$$

2.3.5 SQL-Abfragen

In SQL werden Abfragen folgendermaßen ausgedrückt:

```
SELECT Attributes
FROM Relations
WHERE Condition
```

Die Abfrage der gesamten Relation $R : r$ entspricht z.B. der SQL- Abfrage

$$\texttt{SELECT * FROM } r \ = \ \mathsf{Ans}(\Delta, r(x_1, \ldots, x_m)) \ = \ R$$

Die WHERE-Klausel führt zu einer Selektion:

$$\texttt{SELECT * FROM } r \texttt{ WHERE } Cond \ = \mathrm{Sel}(Cond, R)$$

2.3.6 Reiter's Datenbank-Abgeschlossenheitsannahme

Die Definition der Ob-Anfragen-Beantwortung,

$$\mathsf{Ans}(\Delta, F) = \begin{cases} \text{yes} & \text{if KB} \vdash F \\ \text{no} & \text{if KB} \vdash -F \end{cases}$$

wonach jede Ob-Anfrage mit 'ja' oder 'nein' beantwortet wird, enthält eine starke Annahme: sie setzt voraus, daß die gesamte Extension eines Prädikats r in der Datenbanktabelle R aufgezeichnet ist: $\mathsf{Ans}(\Delta, r(t)) = \text{no}$, wenn $r(t) \notin A_\Delta$, d.h. wenn $t \notin R$.
Diese Annahme heißt nach Reiter [1978] *Closed-World Assumption*, abgekürzt *CWA*. Sie ist nicht immer realistisch. Wenn z.B. die Datenbank eines Einwohnermeldeamts Informationen darüber enthält, welche Einwohner verheiratet sind, dann ist nicht davon auszugehen, daß diese Informationen vollständig sind. Sofern dort kein Eintrag über eine Heirat von Peter Müller vorliegt, heißt das noch lange nicht, daß Peter Müller tatsächlich nicht verheiratet ist. Er könnte auf hoher See von einem Kapitän getraut worden sein, und diese Information müßte nicht unbedingt an die Einwohnermeldedatenbank weitergegeben worden sein.

2.4 Integritätsbedingungen

Im allgemeinen hat eine Integritätsbedingung die Form einer Ob-Anfrage F, und es wird verlangt, daß F in jedem zulässigen Datenbankzustand gilt, d.h. mit 'ja' beantwortet wird.
In relationalen Datenbanken gibt es vor allem zwei wichtige Arten von Integritätsbedingungen: *funktionale Abhängigkeiten* und *Inklusionsabhängigkeiten.*

2.4.1 Funktionale Abhängigkeiten

Eine Relation R erfüllt eine zwischen zwei Attributmengen $X, Y \subseteq \mathrm{Attr}(R)$ bestehende *funktionale Abhängigkeit* $X \to Y$, symbolisch $R \models X \to Y$, wenn gilt:

$$\forall t_1, t_2 \in R: \ v_X(t_1) = v_X(t_2) \ \Rightarrow \ v_Y(t_1) = v_Y(t_2)$$

In prädikatenlogischer Darstellung entspricht dies folgender Ob-Anfrage:

$$\neg\exists\vec{x},\vec{y_1},\vec{y_2} : r(\vec{x},\vec{y_1}) \wedge r(\vec{x},\vec{y_2}) \wedge \vec{y_1} \neq \vec{y_2}$$

wobei das Variablentupel $\vec{x}$ der Attributmenge X entspricht, und die Tupel $\vec{y_1},\vec{y_2}$ der Attributmenge Y entsprechen.
Ein Schlüssel $X \subseteq \mathrm{Attr}(r)$ drückt eine spezielle funktionle Abhängigkeit, nämlich $X \to \mathrm{Attr}(r)$, aus.

2.4.2 Inklusionsabhängigkeiten

Eine Datenbank Δ erfüllt eine *Inklusionsabhängigkeit* zwischen zwei Relationen $R : r$ und $S : s$ bezüglich zweier Attributmengen $X \subseteq \mathrm{Attr}(r)$ und $Y \subseteq \mathrm{Attr}(s)$, symbolisch

$$\Delta \models r[X] \subseteq s[Y]$$

wenn $R[X] \subseteq S[Y]$. Dabei bezeichnet $R[X]$ die Projektion der Relation R auf die Attribut-Teilmenge $X \subseteq \mathrm{Attr}(r)$.
In prädikatenlogischer Darstellung entspricht dies folgender Ob-Anfrage:

$$\Delta \vdash \neg\exists\vec{x},\vec{y},\vec{z} : r(\vec{x},\vec{y}) \wedge \neg s(\vec{x},\vec{z})$$

wobei das Variablentupel $\vec{x}$ sowohl der Attributmenge X als auch Y entspricht.
Die wichtigste Art von Inklusionsabhängigkeit stellen *referentielle Integritätsbedingungen* dar (in SQL heißen sie auch *Fremdschlüssel*). Eine solche 'Fremdschlüsselabhängigkeit' besteht zwischen zwei Tabellen R und S, wenn von jedem R-Tupel verlangt wird, daß die Wertekombination gewisser Attribute $X \subseteq \mathrm{Attr}(r)$ als Primärschlüsselwert eines S-Tupels vorkommen:

$$R[X] \subseteq S[\mathrm{PK}(s)]$$

Dies wird in SQL bei der Definition des Relationsschemas folgendermaßen ausgedrückt:

```
CREATE TABLE r(
...
FOREIGN KEY (X)
REFERENCES s (PK(s))
)
```

2.4.3 Allgemeine Integritätsbedingungen in SQL

Erst im neuen SQL2-Standard wurde eine allgemeinere Form von Integritätsbedingungen definiert. Mit

`CHECK` *Condition* `AFTER UPDATE ON` *Table* `[FOR EACH ROW]`

kann eine beliebige SQL-Bedingung ereigniskontrolliert überprüft werden (dabei können auch dynamische Integritätsbedingungen mit Hilfe der Präfixe '`old.`' und '`new.`' formuliert werden). Ist die Bedingung nicht erfüllt, so wird die dem Aktualisierungsereignis (entweder `UPDATE`, `INSERT` oder `DELETE`) zugrundeliegende Transaktion abgebrochen. Alternativ kann die Bedingung per `BEFORE COMMIT` auch erst am Ende der Transaktion überprüft werden.

2.5 Eine Datenbank als prädikatenlogische Interpretation

Eine relationale Datenbank repräsentiert eine endliche Herbrand- Interpretation im Sinne der Prädikatenlogik. Dabei definiert das Datenbankschema Σ die formale Sprache L_Σ, und der Datenbankzustand Δ definiert eine Interpretation $\mathcal{I}_\Delta$ dieser Sprache, so daß $\mathcal{I}_\Delta$ ein prädikatenlogisches Modell von Δ ist.

2.5.1 Die Sprache L_Σ

Ein Datenbankschema $\Sigma = \{r_1, \ldots, r_n\}$ definiert eine prädikatenlogische Signatur $\langle Const_Sigma\ Rel_\Sigma\rangle$, bestehend aus einer Menge von Konstantensymbolen

$$Const_\Sigma = \bigcup_{i=1}^{n} \{\mathrm{Dom}(A) \mid A \in \mathrm{Attr}(r_i)\}$$

und einer Menge von Relationssymbolen

$$Rel_\Sigma = \{r_1, \ldots, r_n\}.$$

Die prädikatenlogische Formelsprache $L_\Sigma = L_\Sigma(-, \wedge, \vee, \exists, \forall)$ ist dann wie üblich auf dieser Signatur aufgebaut.

2.5.2 Die Interpretation $\mathcal{I}_\Delta$

Eine Datenbank $\Delta : \Sigma$ definiert eine Interpretation $\mathcal{I}_\Delta$ der Sprache L_Σ.

1. Der Individuenbereich $U_{\mathcal{I}_\Delta}$ besteht genau aus der Konstantenmenge $Const_\Sigma$.

2. Konstantensymbole werden durch sich selbst interpretiert: $\mathcal{I}_\Delta(c) = c$ für alle $c \in Const_\Sigma$.

3. Relationssymbole r_i werden durch die in Δ enthaltene Relation R_i interpretiert: $\mathcal{I}_\Delta(r_i) = R_i$ für $i = 1, \ldots, n$.

Damit gilt für die Erfüllung atomarer Aussagen:

$$\mathcal{I}_\Delta \models r(c_1, \ldots, c_m) \;\Leftrightarrow\; \langle \mathcal{I}_\Delta(c_1), \ldots, \mathcal{I}_\Delta(c_m)\rangle \in \mathcal{I}_\Delta(r) \;\Leftrightarrow\; \langle c_1, \ldots, c_m\rangle \in R$$

Die durch die Datenbank Δ induzierte Interpretation $\mathcal{I}_\Delta$ ist ein Modell der Datenbank:

$$\boxed{\mathcal{I}_\Delta \models A_\Delta}$$

d.h. $\mathcal{I}_\Delta$ erfüllt die Aussagenmenge von Δ. Tatsächlich ist $\mathcal{I}_\Delta$ das kleinste Herbrand-Modell von A_Δ.

2.5.3 Folgern auf der Basis minimaler Modelle

2.6 Reiter's Datenbank-Vervollständigungstheorie

Eine interessante Frage ist: um welche zusätzlichen Sätze muß man die Aussagenmenge A_Δ erweitern, so daß sich dieselben Ob-Anfragen, die von $\mathcal{I}_\Delta$ erfüllt werden, auch als normale Folgerungen ergeben, d.h. für welche Menge Γ_Δ von Sätzen gilt

$$\mathcal{I}_\Delta \models F \quad \text{iff} \quad A_\Delta \cup \Gamma_\Delta \models F$$

Reiter [1984] hat gezeigt, daß sich eine solche Vervollständigung von A_Δ in drei Etappen erreichen läßt:

1. Eine Menge $Comp_\Delta$ von **Vollständigkeitsaxiomen** legt für alle Prädikate r fest, daß ihre Extension nicht größer als $\mathcal{I}_\Delta(r) = R$ ist. Das heißt, $Comp_\Delta$ enthält für jedes Relationssymbol $r \in \Sigma$ mit der Interpretation $R = \{t_1, \ldots, t_k\}$ den Satz
$$\forall \vec{x}(r(\vec{x}) \supset \vec{x} = t_1 \vee \vec{x} = t_2 \vee \ldots \vee \vec{x} = t_k)$$

2. Eine Menge UNA_Δ von **Namenseindeutigkeitsaxiomen** legt fest, daß ungleiche Namen ungleiche Individuen bezeichnen. Das heißt, für je zwei in Δ vorkommende Konstantensymbole c, c' enthält UNA_Δ den Satz
$$c \neq c'$$

3. Ein **Wertebereichs-Abgeschlossenheitsaxiom** DCA_Δ legt fest, daß es außer den in Δ genannten Individuen keine weiteren gibt:
$$\forall x(x = c_1 \vee x = c_2 \vee \ldots \vee x = c_l)$$
wobei $\{c_1, \ldots, c_l\}$ eine erschöpfende Aufzählung der in Δ vorkommenden Konstantensymbole ist.

Es ist leicht zu sehen, daß die Vollständigkeitsaxiome zusammen mit den Namenseindeutigkeitsaxiomen zur CWA führen:

$$A_\Delta \cup Comp_\Delta \cup UNA_\Delta \models -r(c_1, \ldots, c_m) \quad \text{iff} \quad \langle c_1, \ldots, c_m \rangle \notin R$$

Die Namenseindeutigkeitsaxiome und das Wertebereichs-Abgeschlossenheitsaxiom führen dazu, daß nur endliche Herbrandinterpretationen, deren Grundbereich genau aus den Konstanten von Δ besteht, als Modelle akzeptiert werden, d.h. Γ_Δ erfüllen können. Durch das Wertebereichs-Abgeschlossenheitsaxiom wird erreicht, daß allquantifizierte Aussagen genau dann gefolgert werden können, wenn sie von $\mathcal{I}_\Delta$ erfüllt werden. Die inhaltliche Bedeutung dieser Annahme, wonach es außer den in der Datenbank explizit genannten keine weiteren Entitäten gibt, ist jedoch eher fraglich. Dies scheint auf ein allgemeines bedeutungstheoretisches Problem des Allquantors hinzuweisen: seine 'natürliche' Semantik ist offenbar nicht identisch mit seiner modelltheoretischen Semantik.

Wir illustrieren Reiter's Vervollständigungstheorie mit einem Beispiel. Für die Datenbank $\Delta : \Sigma = \{P : p, Q : q\}$ mit den Tabellen

$$P = \boxed{\begin{array}{c} 1 \\ 2 \end{array}} \quad Q = \boxed{\begin{array}{cc} c & d \end{array}}$$

ergibt sich $A_\Delta = \{p(1), p(2), q(c,d)\}$ und

$$\begin{array}{rcl} Comp_\Delta & = & \{\forall x(p(x) \supset x = 1 \vee x = 2),\, \forall x, y(q(x,y) \supset \langle x, y\rangle = \langle c, d\rangle)\} \\ UNA_\Delta & = & \{1 \neq 2,\, 1 \neq c,\, 1 \neq d,\, 2 \neq c,\, 2 \neq d,\, c \neq d\} \\ DCA_\Delta & = & \forall x(x = 1 \vee x = 2 \vee x = c \vee x = d) \end{array}$$

Offensichtlich gilt: $\mathcal{I}_\Delta \models -p(c)$ und ebenso $A_\Delta \vdash -p(c)$. Es ist leicht einzusehen, daß man $-p(c)$ hingegen mit der modelltheoretischen Folgerungsrelation nicht aus A_Δ erhält: $A_\Delta \not\models -p(c)$. Dazu muß man nur die Interpretation $\mathcal{I}_1 = \{p(1), p(2), p(c), q(c,d)\}$ betrachten, die zwar ein Modell von A_Δ ist, aber nicht $-p(c)$ erfüllt. Die Interpretation $\mathcal{I}_1$ ist jedoch kein Modell der Vervollständigung Γ_Δ, da sie das Vollständigkeitsaxiom für p verletzt. Es gilt wie gewünscht:

$$A_\Delta \cup Comp_\Delta \cup UNA_\Delta \cup \{DCA_\Delta\} \quad \models \quad -p(c)$$

2.7 Übungsaufgaben

1. Wodurch genau verletzt die Interpretation $\mathcal{I}_1$ im Beispiel zu Reiter's Vervollständigungstheorie das Vollständigkeitsaxiom für p ?

2. Wie sind $Comp_\Delta$, UNA_Δ und DCA_Δ für $A_\Delta = \{q(*), r(c,1), r(*,2)\}$ definiert ?

Chapter 3

Abstrakte Wissenssysteme

Im allgemeinen kann eine Wissensbasis aus einer mehr oder weniger komplexen Datenstruktur bestehen, die in der Lage ist, Wissen zu repräsentieren. Dies kann z.B. eine aus logischen Ausdrücken bestehende (evtl. verschachtelte) Menge, eine Sequenz oder ein Graph sein. Wir gehen daher abstrakt von einer Menge L_{KB} aus, deren Elemente genau die zulässigen Wissensbasis-Instanzen sind. Im Falle relationaler Datenbanken sind das Grundatom-Mengen, d.h. $L_{\mathrm{KB}} = 2^{\mathrm{At}^0}$.
Genauso muß für jedes Wissenssystem die Menge der zulässigen Anfrageformeln festgelegt sein. Wir bezeichnen sie mit L_{Query}. Bei relationalen Datenbanken ist L_{Query} die Menge aller auswertbaren Formeln aus $L(-, \wedge, \vee, \exists, \forall)$.
Ein Wissensystem besteht aus zwei grundlegenden Operationen: einer Inferenzoperation $\vdash$ (bzw. der auf ihr basierenden Antwortoperation Ans) und einer Aktualisierungsoperation Upd. Die Antwortoperation verarbeitet Anfragen, und die Aktualisierungsoperation verarbeitet Eingaben an die Wissensbasis. Dabei können Anfragen und Eingaben sowohl von menschlichen Benutzern als auch von anderen Quellen ausgehen. Zum Beispiel könnte ein automatischer Prozeß eine Anfrage an die Wissensbasis richten, oder ein Sensor könnte neue Eingabedaten liefern. Wieder muß festgelegt sein, was zulässige Eingaben sind. Wir bezeichnen die Menge der zulässigen Eingaben mit L_{Input}. Da jede eingegebene Information auch wieder abfragbar seiin soll, verlangen wir, daß $L_{\mathrm{Input}} \subseteq L_{\mathrm{Query}}$.
Damit ergibt sich folgende Definition für ein abstraktes Wissenssystem.

Definition 1 (Wissenssystem) *Eiin abstraktes Wissenssystem ist ein Quintupel*

$$\boldsymbol{K} = \langle\, L_{\mathrm{KB}},\, \vdash,\, L_{\mathrm{Query}},\, \mathsf{Upd},\, L_{\mathrm{Input}} \,\rangle$$

bestehend aus einer Inferenzrelation $\vdash$ zwischen Wissensbasen aus L_{KB} und Ob-Anfragen aus L^0_{Query}, und einer Aktualisierungsoperation $\mathsf{Upd} : L_{\mathrm{KB}} \times L^0_{\mathrm{Input}} \to L_{\mathrm{KB}}$, *so daß für jede Wissensbasis $X \in L_{\mathrm{KB}}$ gilt:*

(KS1) $X \vdash 1$, *and* $\mathsf{Upd}(X, 1) = X$.

(KS2) $L_{\mathrm{Input}} \subseteq L_{\mathrm{Query}}$.

(KS3) $\mathsf{Upd}(X, F) \vdash F$, *für alle Eingaben $F \in L^0_{\mathrm{Input}}$, die mit X konsistent sind.*

Die Definition der (definiten) Antwortoperation basiert auf der Inferenzrelation:

$$\mathsf{Ans}(X,\ F(x_1,\ldots,x_m)) := \{\langle c_1,\ldots,c_m\rangle \mid X \vdash F(c_1,\ldots,c_m)\}$$

Eine Antwort kann auch als eine Substitution aufgefaßt werden:

$$\mathsf{Ans}(X,\ F(x_1,\ldots,x_m)) = \{\{x_1/c_1,\ldots,x_m/c_m\} : X \vdash F(c_1,\ldots,c_m)\}$$

Da es oft nützlich ist, eine Wissensbasis mit einer Menge von Eingaben aktualisieren zu können, 'überladen' wir das Operatorsymbol Upd und bezeichnen damit auch die allgemeinere Aktualisierungsoperation

$$\mathsf{Upd} : L_{\mathrm{KB}} \times 2^{L^0_{\mathrm{Input}}} \to L_{\mathrm{KB}}$$

die so definiert werden muß, daß für jede endliche Eingabenmenge $A \subseteq L^0_{\mathrm{Input}}$ gilt: $\mathsf{Upd}(\mathrm{KB}, A) = \mathsf{Upd}(\mathrm{KB}, \bigwedge A)$.
Wir schreiben auch $\mathrm{KB}+F$ als Abkürzung von $\mathsf{Upd}(\mathrm{KB}, F)$, und $\mathrm{KB}-F$ als Abkürzung für $\mathsf{Upd}(\mathrm{KB}, -F)$.

3.0.1 Reguläre Wissenssysteme

Um Wissensbasen hinsichtlich ihres Informationsgehalts vergleichen zu können, benötigen wir eine Informations- oder *Wissensordnung* $\leq$ zwischen Wissensbasen, so daß

$$\mathrm{KB}_1 \leq \mathrm{KB}_2 \quad \textit{falls } \mathrm{KB}_2 \textit{ mindestens soviel Information wie } \mathrm{KB}_1 \textit{ enthält.}$$

Die informationsleere Wissensbasis bezeichnen wir mit 0. Per definitionem gilt, daß $0 \leq X$ für alle $X \in L_{\mathrm{KB}}$, d.h. 0 ist das kleinste Element von $\langle L_{\mathrm{KB}}, \leq\rangle$.
Im allgemeinen bleiben Antworten bei Informationswachstum nicht unbedingt erhalten. Anfrageformeln, für die alle Antworten erhalten bleiben, heißen *persistent*. Formal: seien $X_1, X_2 \in L_{\mathrm{KB}}$ Wissensbasen mit $X_1 \leq X_2$, dann ist eine Ob-Anfrage $F \in L^0_{\mathrm{Query}}$ *persistent* wenn $X_1 \vdash F$ impliziert, daß $X_2 \vdash F$; eine allgemeine Anfrage $F \in L_{\mathrm{Query}}$ ist *persistent* wenn $\mathsf{Ans}(X_1, F) \subseteq \mathsf{Ans}(X_2, F)$. Die Menge der persistenten Anfrageformeln bezeichnen wir mit L_{PersQ}. Wenn alle Anfrageformeln $F \in L_{\mathrm{Query}}$ persistent sind, heißt auch das betreffende Wissenssystem und seine Inferenzrelation persistent. Ein Operator der Abfragesprache heißt persistent, wenn jede aus persistenten Subformeln mit ihm gebildete Formel wieder persistet ist.

Eine Eingabe $F \in L^0_{\mathrm{Input}}$ heißt *ampliativ*, wenn für jede Wissensbasis $X \in L_{\mathrm{KB}}$ gilt: $X \leq \mathsf{Upd}(X, F)$; F heißt *reduktiv*, wenn $X \geq \mathsf{Upd}(X, F)$. Die Menge der ampliativen Eingaben bezeichnen wir mit L_{AmpI}. Ein Wissenssystem und seine Aktualisierungsoperation heißen ampliativ, wenn alle Eingaben $F \in L^0_{\mathrm{Input}}$ ampliativ sind.

Wir nehmen an, daß jedes Wissenssystem auf einer bestimmten Art von *Informationseinheiten* $L_{\mathrm{Unit}} \subseteq L_{\mathrm{Input}}$ beruht. Zum Beispiel bestehen die Informationseinheiten des Systems Relationaler Datenbanken aus Atomen. Eine Informationseinheit repräsentiert eine elementare Information mit positivem Informationsgehalt.

Eine Wissensbasis kann widersprüchliche Informationen beinhalten. Um das Ausmaß an Widersprüchlichkeit zu erfassen, gehen wir von einer Funktion $\mathrm{Inc} : Ł_{\mathrm{KB}} \to 2^{L_{\mathrm{Unit}}}$

aus, die alle widersprüchlichen Informationseinheiten einer Wissensbasis Xin der Menge $\mathrm{Inc}(X) \subseteq L_{\mathrm{Unit}}$ zusammenfaßt.

Ein Wissenssystem $\boldsymbol{K} = \langle\, L_{\mathrm{KB}},\, \vdash,\, L_{\mathrm{Query}},$ Upd, $L_{\mathrm{Input}}\,\rangle$ ist dann *regulär*, wenn es eine Präordnung $\langle L_{\mathrm{KB}}, \leq, 0\rangle$ mit kleinstem Element 0, eine Menge $L_{\mathrm{Unit}} \subseteq L_{\mathrm{Input}}$, und eine Funktion $\mathrm{Inc} : L_{\mathrm{KB}} \rightarrow 2^{L_{\mathrm{Unit}}}$ gibt, so daß

(KS4) Konsistente Elementareingaben verstärken den Informaionsgehalt: $X \leq X + u$, für alle $X \in L_{\mathrm{KB}}$, und für alle $u \in L_{\mathrm{Unit}}$, so daß $u \notin \mathrm{Inc}(X)$, und $\mathrm{Inc}(X+u) \subseteq \mathrm{Inc}(X)$.

(KS5) Die Informationsordnung ist mit ampliativer Aktualisierug und persistenter Inferenz kompatibel: für alle $X_1, X_2 \in L_{\mathrm{KB}}$,

$$X_1 \leq X_2 \quad \text{iff} \quad \forall F \in L_{\mathrm{AmpI}} \forall G \in L_{\mathrm{PersQ}} : X_1 + F \vdash G \;\Rightarrow\; X_2 + F \vdash G$$

(KS6) Ampliative Eingaben sind persistente Anfragen: $L_{\mathrm{AmpI}} = L_{\mathrm{PersQ}} \cap L_{\mathrm{Input}}$.

(KS7) Die Inferenz ist konsistent: für alle $X \in L_{\mathrm{KB}}$, und $F \in L_{\mathrm{Input}}$, $X \vdash F$ impliziert $\mathrm{Inc}(X+F) \subseteq \mathrm{Inc}(X)$.

Ein reguläres Wissenssystem wird als 9-Tupel repräsentiert:

$$\langle\, 0,\, \leq,\, L_{\mathrm{KB}},\, \vdash,\, L_{\mathrm{Query}},\, \mathsf{Upd},\, L_{\mathrm{Input}},\, \mathrm{Inc},\, L_{\mathrm{Unit}}\,\rangle$$

Chapter 4

Faktenbasen

Eine Faktenbasis ist eine endliche Menge von positiven und negativen Tabellen. Eine positive Tabelle repräsentiert eine endliche Relation, also die Menge aller Tupel, auf die die Relation zutrifft, oder mit anderen Worten: ihre Wahrheitsextension. Eine negative Tabelle enthält all diejenigen Tupel, für die die Relation definitiv nicht gilt, d.i. ihre Falschheitsextension. Relationen, für die die Abgeschlossenheitsannahme (CWA) gilt, werden nur durch eine positive Tabelle repräsentiert. Ihre Falschheitsextension ist implizit durch ihr Komplement gegeben. In einer Faktenbasis werden also zwei Arten von Relationen unterschieden, je nachdem, ob die CWA gilt oder nicht. Diese Festlegung wird im Faktenbasis-Schema getroffen.

Ein *Faktenbasis-Schema* $\Sigma = \langle Rel, CWRel, IC \rangle$ besteht aus einer Menge von Relationsschemata Rel, einer Teilmenge $CWRel \subseteq Rel$, die festlegt, für welche Relationen die CWA gilt, und einer Menge von Integritätsbedingungen IC.

Zum Beispiel ergeben die folgenden Tabellen für *Beschäftigte*, *RaucherIn*, und *NichtraucherIn*:

$B =$

Name	Zimmer
A	02
T	05
E	03
U	05
H	02

$R =$

Name
A

$\tilde{R} =$

Name
U
E

die Faktenbasis $\Phi_1 = \{B, R, \tilde{R}\}$ über dem Schema

$$\Sigma_1 = \langle \{b, r\},\ \{b\},\ \{\neg \exists x : b(x, 03) \wedge r(x)\} \rangle,$$

d.h. die CWA gilt nur für das Relationssymbol b und es gibt eine Integritätsbedingung, die verlangt, daß in Zimmer 03 nicht geraucht wird.

Eine Faktenbasis kann – wie eine relationale Datenbank – als Aussagenmenge dargestellt werden. Eine positive Tabelle entspricht einer Menge von atomaren Aussagen, während eine negative Tabelle einer Menge von negierten atomaren Aussagen entspricht. So entspricht zum Beispiel die Faktenbasis Φ_1 der literalen Aussagenmenge

$$A_{\Phi_1} = \{\, b(A, 02), b(T, 05), \ldots, r(A), \sim r(U), \sim r(E)\,\}$$

Wir gehen davon aus, daß eine Faktenbasis keine Widersprüche enthält. Das heißt, daß korrespondierende positive und negative Tabellen keine gemeinsamen Einträge enthalten, also $R \cap \tilde{R} = \emptyset$.

Eine Faktenbasis $\Phi : \Sigma$ kann als eine widerspruchsfreie Menge von *Literalen* $A_\Phi \subseteq \mathrm{Lit}(\Sigma)$ repräsentiert werden.

Im folgenden werden Faktenbasen je nach Zweckmäßigkeit als Tabellen-Mengen oder als Aussagenmengen dargestellt.

Mögliche Aktualisierungen sind die Hinzunahme einer neu eingehenden Information in Form eines Literals $l \in \mathrm{Lit}(\Sigma)$, wobei nur konsistente Informationen akzeptiert werden:

$$\mathsf{Upd}(A_\Phi, l) \;=\; \begin{cases} A_\Phi \cup \{l\} \text{ wenn } l \notin A_\Phi \text{ und } A_\Phi \cup \{l\} \vdash IC \\ A_\Phi \text{ sonst} \end{cases}$$

und die Löschung eines Literals:

$$\mathsf{Upd}(A_\Phi, -l) = A_\Phi - \{l\}$$

4.1 Natürliche Inferenz in Faktenbasen

Bei der Inferenz von Ob-Anfragen in Faktenbasen unterscheidet man zwischen der (schon von relationalen Datenbanken bekannten) *schwachen* Negation '$-$' und der durch die Möglichkeit negativer Tabellen neu hinzugekommenen *starken* Negation '$\sim$'. Konjunktion, Disjunktion und die Quantoren werden auf die übliche Art ausgewertet. Bei der folgenden induktiven Definition der Inferenzrelation nehmen wir an, daß $\Phi = \{R : r, \tilde{R} : r, \ldots\}$ eine Faktenbasis über dem Schema $\Sigma = \langle Rel, CWRel, IC\rangle$ ist und daß $r \in Rel$.

(a)	$\Phi \vdash r(c_1, \ldots, c_n)$	wenn	$\langle c_1, \ldots, c_n\rangle \in R$
$(\sim a)$	$\Phi \vdash \sim r(c_1, \ldots, c_n)$	wenn	$\langle c_1, \ldots, c_n\rangle \in \tilde{R}$
$(-a)$	$\Phi \vdash -r(c_1, \ldots, c_n)$	wenn	$\langle c_1, \ldots, c_n\rangle \notin R$
$(-\sim a)$	$\Phi \vdash -\sim r(c_1, \ldots, c_n)$	wenn	$\langle c_1, \ldots, c_n\rangle \notin \tilde{R}$
$(\wedge)$	$\Phi \vdash F \wedge G$	wenn	$\Phi \vdash F$ und $\Phi \vdash G$
$(\vee)$	$\Phi \vdash F \vee G$	wenn	$\Phi \vdash F$ oder $\Phi \vdash G$
$(\forall)$	$\Phi \vdash \forall \vec{x} F(\vec{x})$	wenn	$\Phi \vdash F(\vec{c})$ für alle Konstanten-Tupel $\vec{c}$
$(\exists)$	$\Phi \vdash \exists \vec{x} F(\vec{x})$	wenn	$\Phi \vdash F(\vec{c})$ für ein gewisses $\vec{c}$

Diese Definition wird durch die folgenden DeMorgan-Umformungsregeln:

$$\begin{array}{rcl@{\qquad}rcl}
\sim (F \wedge G) & \longrightarrow & \sim F \vee \sim G & \sim (F \vee G) & \longrightarrow & \sim F \wedge \sim G \\
\sim \exists x F(x) & \longrightarrow & \forall x \sim F(x) & \sim \forall x F(x) & \longrightarrow & \exists x \sim F(x) \\
\sim \sim F & \longrightarrow & F & \sim -F & \longrightarrow & F \\
-(F \vee G) & \longrightarrow & -F \wedge -G & -(F \wedge G) & \longrightarrow & -F \vee -G \\
-\!\sim (F \vee G) & \longrightarrow & -\!\sim F \vee -\!\sim G & -\!\sim (F \wedge G) & \longrightarrow & -\!\sim F \wedge -\!\sim G \\
- \sim \sim F & \longrightarrow & -F & - - F & \longrightarrow & F \\
-\!\sim - F & \longrightarrow & -F & & &
\end{array}$$

und die Definitionen der *exklusiven Disjunktion*

$$F|G \longrightarrow (F \wedge -G) \vee (G \wedge -F)$$

und der *materialen Implikation*

$$F \supset G \longrightarrow -F \vee G$$

vervollständigt. Für jede derartige Umformungsregel $LHS \longrightarrow RHS$ gilt:

$$\Phi \vdash LHS :\Longleftrightarrow \Phi \vdash RHS$$

4.2 Die CWA in Faktenbasen

Die CWA gilt in Faktenbasen nur noch eingeschränkt für die in $CWRel$ aufgeführten Relationssymbole. Entsprechend wird die Definition der Inferenzrelation um eine Klausel für die starke Negation erweitert:

$$\Phi \vdash \sim r(c_1, \ldots, c_n) :\Longleftrightarrow r \in CWRel \;\; \& \;\; \langle c_1, \ldots, c_n \rangle \notin R$$

4.3 Eine Faktenbasis als partielle Interpretation

Eine Faktenbasis repräsentiert eine endliche partielle Herbrand-Interpretation. Dabei definiert das Schema Σ die formale Sprache L_Σ, und der Faktenbasis-Zustand Φ definiert eine partielle Interpretation $\mathcal{I}_\Phi$ dieser Sprache, so daß $\mathcal{I}_\Phi$ ein partielles Modell der Aussagenmenge A_Φ ist.
Wie schon bei relationalen Datenbanken definiert ein Schema Σ eine prädikatenlogische Signatur $\langle Const_\Sigma \; Rel_\Sigma \rangle$, bestehend aus einer Menge von Konstantensymbolen und einer Menge von Relationssymbolen. Die prädikatenlogische Formelsprache $L_\Sigma = L_\Sigma(-, \sim, \wedge, \vee, \exists, \forall)$ ist dann wie üblich auf dieser Signatur aufgebaut.

4.3.1 Die Interpretation $\mathcal{I}_\Phi$

Eine Faktenbasis $\Phi : \Sigma$ definiert eine partielle Interpretation $\mathcal{I}_\Phi$ der Sprache L_Σ.

1. Der Individuenbereich U von $\mathcal{I}_\Phi$ besteht genau aus der Konstantenmenge $Const_\Sigma$.

2. Konstantensymbole werden durch sich selbst interpretiert: $\mathcal{I}_\Phi(c) = c$ für alle $c \in Const_\Sigma$.

3. Relationssymbole r_i werden durch die in Φ enthaltenen Wahrheits- und Falschheitstabellen R_i und $\tilde{R}_i$ interpretiert: $\mathcal{I}_\Phi(r_i) = \langle R_i, \tilde{R}_i \rangle$, für $i = 1, \ldots, m$. Für $r_i \in CWRel$ ergibt sich die Falschheitstabelle durch das Komplement: $\tilde{R}_i = U^n - R_i$.

Damit gilt für die Erfüllung literaler Aussagen:

$$\begin{array}{l} \mathcal{I}_\Phi \models r(c_1, \ldots, c_m) \;\Leftrightarrow\; \langle \mathcal{I}_\Phi(c_1), \ldots, \mathcal{I}_\Phi(c_m) \rangle \in \mathcal{I}^+_\Phi(r) \;\Leftrightarrow\; \langle c_1, \ldots, c_m \rangle \in R \\ \mathcal{I}_\Phi \models {\sim} r(c_1, \ldots, c_m) \;\Leftrightarrow\; \langle \mathcal{I}_\Phi(c_1), \ldots, \mathcal{I}_\Phi(c_m) \rangle \in \mathcal{I}^-_\Phi(r) \;\Leftrightarrow\; \langle c_1, \ldots, c_m \rangle \in \tilde{R} \end{array}$$

Die durch die Faktenbasis Φ induzierte Interpretation $\mathcal{I}_\Phi$ ist ein partielles Modell der Faktenbasis:

$$\boxed{\mathcal{I}_\Phi \models A_\Phi}$$

d.h. $\mathcal{I}_\Phi$ erfüllt die Aussagenmenge von Φ. Tatsächlich ist $\mathcal{I}_\Phi$ das kleinste partielle Modell von A_Φ.

4.3.2 Folgern auf der Basis minimaler Modelle

Jede partielle Herbrand-Interpretation $\mathcal{I}$ entspricht einer Menge von Grundliteralen $D_\mathcal{I} = \{l \in \mathrm{Lit}^0 : \mathcal{I} \models l\}$. Eine Interpretation $\mathcal{I}$ ist weniger informativ als eine andere interpretation $\mathcal{J}$, symbolisch $\mathcal{I} \leq \mathcal{J}$, wenn $D_\mathcal{I} \subseteq D_\mathcal{J}$. Eine Aussage F kann aus aus einer Aussagenmenge X gefolgert werden, $X \models_4 F$, wenn jedes partielle Modell von X auch ein partielles Modell von F ist, d.h. $\mathsf{Mod}_4(X) \subseteq \mathsf{Mod}_4(F)$. Das Folgern auf der Basis minimaler Modelle ist enstprechend definiert:

$$X \models^m_4 F :\Longleftrightarrow \mathrm{Min}(\mathsf{Mod}_4(X)) \subseteq \mathsf{Mod}_4(F)$$

Im Hinblick auf die Ableitbarkeit einer Ob-Anfrage aus einer Faktenbasis besteht ein dreifacher Zusammenhang zwischen natürlicher Inferenz, minimalem Folgern und der Modell-Relation:

$$\Phi \vdash F \quad \text{iff} \quad A_\Phi \models^m_4 F \quad \text{iff} \quad \mathcal{I}_\Phi \models F$$

4.4 Unvollständige Information

In der mathematischen Logik heißt eine Theorie X bezüglich der modelltheoretischen Folgerungrelation $\models$ *vollständig*, wenn für jede Aussage F gilt:

$$X \models F \quad \text{oder} \quad X \models {\sim} F$$

Zum Beispiel sind relationale Datenbanken wegen der generellen Abgeschlossenheitsannahme (CWA) in diesem Sinn vollständig, Faktenbasen jedoch nicht.

Chapter 5

Ungenaue und unsichere Information

Ungenaue Informationen können als disjunktive Aussagen verstanden werden. Beispiele dafür sind

1. *Der Täter benutzte einen schwarzen BMW oder Mercedes mit Kennzeichen B-ED 007.* Diese Assage könnte folgendermaßen formalisiert werden:

   $$tf(BMW, schwarz, \textit{B-ED 007}) \vee tf(Mercedes, schwarz, \textit{B-ED 007})$$

2. *Das Flötenkonzert Nr.9 von Teleman wurde zwischen 1743 und 1745 uraufgeführt.* Diese Aussage könnte durch komplexe Attributwerte wie z.Bsp. $u(9, [1743, 1745])$ oder $u(9, 1744 \pm 1$, oder durch folgende Disjunktion ausgedrückt werden:

   $$u(9, 1743) \vee u(9, 1744) \vee u(9, 1745)$$

Der Extremfall an Ungenauigkeit liegt dann vor, wenn zu einem Attribut gar keine Information bekannt ist wie zum Beispiel in

Das Kennzeichen des Tatfahrzeugs ist unbekannt.

Solche Aussagen können in relationalen Datenbanken mit Hilfe von *Nullwerten* ausgedrückt werden: $tf(BMW, schwarz, NULL)$. Rein logisch gesehen entsprechen sie einer Existenzaussage

$$\exists x : tf(BMW, schwarz, x)$$

Nullwerte können in relationalen Datenbanken auch als Verlegenheitslösung zur Repräsentation von ungenauer Information benutzt werden. Anstatt der Aussage $u(9, [1743, 1745])$, die in einer relationalen Datenbank nicht repräsentierbar ist, wird einfach $u(9, NULL)$ eingetragen und der damit verbundene Informationsverlust in Kauf genommen.

5.1 Unsichere Information

In vielen Fällen gibt es einen Trade-off zwischen Genauigkeit und Sicherheit: je genauer eine Angabe gemacht werden soll, desto unsicherer ist sie. Ungenauigkeit und Unsicherheit können auch kombiniert auftreten. Die Unsicherheit eines Attributwerts oder einer Aussage kann auf unterschiedliche Weise modelliert werden. Die klassische Modellierung von Unsicherheit beruht auf der Wahrscheinlichkeitstheorie, die aber starke Voraussetzungen an die verfügbare Information stellt. Demgegenüber scheint die *Possibilitäts-Theorie* von Zadeh/Dubois/Prade, bzw. die entsprechende *possibilistische Logik* von Dubois und Prade den praktischen Vehältnissen gegenüber angemessener, da weniger idealisierend, zu sein.
Bei unserem Beispiel der Uraufführung vo Telemans 9. Flötenkonzert bleibend, könnten wir folgende Darstellungen mit einem unsicheren Attributwert erhalten:

Probabilistisch: $u(9, \begin{array}{|cc|} \hline 1743 & 0.25 \\ 1744 & 0.5 \\ 1745 & 0.25 \\ \hline \end{array})$

Possibilistisch: $u(9, \begin{array}{|cc|} \hline 1743 & 0.25 \\ 1744 & 0.5 \\ 1745 & 0.5 \\ \hline \end{array})$

Beachte, daß die Summe der Wahrscheinlichkeitswerte 1 ergeben muß, während dies bei den possibilistischen Unsicherheitswerten nicht der Fall sein muß. Unsicherheitswerte müssen auch nicht unbedingt Zahlen zwischen 0 und 1 sein, es können auch Ausdrücke wie *völlig unsicher*, *ziemlich sicher*, u.ä. sein.

Eine *Unsicherheits-Skala* $\langle \mathsf{C}, 0, 1\rangle$ ist definiert als eine linear geordnete Menge C mit kleinstem und größtem Element 0 und 1.

Dabei können 0 und 1 auch für sprachliche Ausdrücke wie *völlig unsicher* und *ganz sicher* stehen.

5.2 Unsichere Datenbanken

Zur Vereinfachung betrachten wir keine unsicheren Attribute, sondern nur unsichere Aussagen, die Tupeln in einer *unsicheren Relation* entsprechen. Eine unsichere Relation über den Attributen $A_1, \ldots, A_n$ ist eine Funktion die jedem Tupel aus $\mathrm{Dom}(A_1) \times \ldots \times \mathrm{Dom}(A_n)$ einen Unsicherheitswert aus C zuordnet. Eine endliche unsichere Relation kann auch als Tabelle mit einer speziellen Spalte für den Unsicherheitswert dargestellt werden.

Eine unsichere Datenbank ist eine endliche Menge endlicher unsicherer Relationen. Sie kann auch als eine endliche Menge unsicherheitsqualifizierter atomarer Aussagen der Form $a{:}\nu$ repräsentiert werden, wobei a ein relationales Atom und $\nu \in \mathsf{C}$ der ihm zugeordnete Unsicherheitswert ist.

Example 2 *Die unsichere Datenbank bestehend aus den beiden Tabellen*

$$P = \begin{array}{|c|c|} \hline d & 1 \\ b & 0.7 \\ c & 0.1 \\ \hline \end{array} \qquad Q = \begin{array}{|cc|c|} \hline d & c & 0.8 \\ d & b & 0.3 \\ \hline \end{array}$$

korrespondiert mit $X_2 = \{p(d){:}1,\ p(b){:}0.7,\ p(c){:}0.2,\ q(d,c){:}0.8,\ q(d,b){:}0.3\}$.

Definition 2 (Induzierte Unsicherheitsbewertung) *Eine unsichere Datenbank* X *induziert auf natürliche Weise eine Unsicherheitsbewertung* C_X*:*

$$\begin{array}{cl}
(a) & C_X(a) = \begin{cases} \nu \text{ if } a{:}\nu \in X \\ 0 \text{ sonst} \end{cases} \\
(-a) & C_X(-a) = \begin{cases} 0 \text{ if } C_X(a) > 0 \\ 1 \text{ sonst} \end{cases} \\
(\wedge) & C_X(F \wedge G) = \min(C_X(F), C_X(G)) \\
(\vee) & C_X(F \vee G) = \max(C_X(F), C_X(G)) \\
(\exists) & C_X(\exists x F(x)) = \max\{C_X(F(t)) \mid t \in U_\sigma\} \\
(\forall) & C_X(\forall x F(x)) = \min\{C_X(F(t)) \mid t \in U_\sigma\}
\end{array}$$

Alle anderen Fälle von komplexen Aussagen werden mit Hilfe der DeMorgan-Umformungsregeln behandelt.

Definition 3 (Natürliche Inferenz) $X \vdash F{:}\nu$ *iff* $C_X(F) \geq \nu$.

Example 3 *Wir erhalten z.Bsp. folgende Inferenz:*

$$X_2 \vdash [p(b) \wedge \neg\exists x q(b,x)] : 0.7$$

Chapter 6

Temporale Datenbanken

Chapter 7

Nullwerte

7.1 Codd-Tabellen und SQL-Nullwerte

7.2 Skolem-Tabellen

Chapter 8

Disjunktive Information

8.1 ODER-Tabellen

8.2 Disjunktive Datenbanken

Part II

Regelwissen

Chapter 9

Deduktive Datenbanken

Chapter 10

Aktive Datenbanken

Chapter 11

Kooperative Datenbanken

Chapter 12

Objekt-Datenbanken

Dr. Gerd Wagner

Logische Grundlagen von Informationssystemen
Vorlesung WS 1994/95

Übungsblatt 1

Aufgabe 1

Gegeben sind die beiden Tabellen

$R =$

A	B	C
2	d	$*$
1	d	$+$
3	b	$+$

$S =$

A	C	D	E
2	$*$	d	3
2	$*$	b	7
1	$+$	b	2
2	$+$	c	3

und die Datenbank $\Delta_1 = \{R : r, S : s\}$.

1. Definiere ein zu Δ_1 passendes Datenbankschema Σ_1. Welche funktionalen und welche Inklusionsabhängigkeiten (insbesondere Primär- und Fremdschlüssel) gibt es ?

2. Wie sieht das zugehörige Universum U_{Δ_1} aus ?

3. Wie lautet explizit die durch Δ_1 repräsentierte Aussagenmenge A_{Δ_1} ?

4. Welches Resultat liefert die folgende SQL-Abfrage:

   SELECT B,E
   FROM r,s
   WHERE $R.C = S.C$ AND $E > 2$

   Wie kann man dieselbe Abfrage als logische Formel ausdrücken ? Wie wird diese mit Hife der relationalen Algebra verarbeitet ?

5. Sei $F(u, w, x) = \exists y : s(u, w, x, y) \wedge -r(u, x, w)$.

   (a) Gilt $\Delta_1 \vdash F(1, b, +)$?
   (b) Ist $F(u, w, x)$ auswertbar ? Warum ?
   (c) Berechne Ans$(\Delta_1, F(u, w, x))$.
   (d) Zeige durch ein Evolutionsbeispiel (eine geeignete Informationszunahme in Δ_1), daß F nicht persistent ist.

Aufgabe 2

Wer hat das relationale Datenbankmodell vorgeschlagen ? (Jahr und Titel des Aufsatzes)

Aufgabe 3

Welche Argumente sprechen dagegen, eine relationale Datenbank als eine prädikatenlogische Interpretation (im Sinne der Modelltheorie) aufzufassen ?

Aufgabe 4

Gegeben sind die folgenden Tabellen für *Beschäftigte*, *RaucherIn*, und *NichtraucherIn*:

$B =$

Name	Zimmer
A	02
T	05
E	03
U	05
H	02

$R =$

Name
A

$\tilde{R} =$

Name
U
E

und die Faktenbasis $FB_1 = \{b, r, \tilde{r}\}$ über dem Schema $\Phi_1 = \langle\{b, r\},\ \{b\},\ \emptyset\rangle$, d.h. die CWA gilt nur für die Beschäftigtentabelle und es gibt keine Integritätsbedingungen.

1. Sei $F(x) = {\sim}b(x, 05) \wedge {\sim}r(x)$.
   (a) Gilt $A_{FB_1} \vdash F(H)$?
   (b) Berechne Ans$(A_{FB_1}, F(x))$.
2. Formuliere die Formeln für die folgenden Anfragen und berechne die zugehörigen Antworten:
   (a) Von wem ist bekannt, daß sie (bzw. er) raucht oder nicht raucht ?
   (b) Wer in Zimmer 05 ist weder Raucher noch Nichtraucher ?
3. Die Welt verändert sich: U verläßt die Firma und H wird zum militanten Nichtraucher, d.h. aus FB_1 wird FB_2:

   $$A_{FB_2} = \mathsf{Upd}(A_{FB_1}, -b(U, 05) \wedge {\sim}r(H))$$

   Berechne die aktuellen Antworten zu allen obigen Anfragen.

Diplomarbeiten

Esperanca Maneira

Thema: "Die Closed-World Assumption in der modularen Expertensystem-Architektur DESIRE".

Gliederung

1. Einleitung: Was sind Expertensysteme, welche Probleme gibt es ... (v.a. aus dem Buch von Ginsberg)
2. Die Prinzpien von DESIRE: Faktenmengen als (partielle) 'Input-Modelle', Regel-Sprache, Bottom-Up-Auswertung (Vorwärtsverkettung), Modularisierung, 'Meta-Module' für Default-Inferenz, etc.
3. Die Closed-World Assumption in Faktenbasen.
4. Die Closed-World Assumption in Erweiterten Logikprogrammen.
5. Negation-as-Failure und Closed-World Assumption in DESIRE.
6. DESIRE-Default-Regeln (in 'Meta-Modulen') als erweiterte Logikprogrammier-Regeln (in 'Objekt-Modulen').
7. Diskussion von Beispielen (Büro-Konfiguration ?).
8. Evtl. Modifikation der DESIRE-Implementation.

Samira Misraoui

Thema: "Automatische Planung mit negativer Information" oder "Die Closed-World Assumption in automatischen Planungssystemen".

Gliederung

1. Einleitung: Was ist ein Planungsystem, welche Probleme gibt es ...
2. Handlungsrepräsentation und Planung im Situatioskalkül.
3. Handlungsrepräsentation und Planung in STRIPS.
4. Vergleich zwischen Situatioskalkül und STRIPS.
5. Die Closed-World Assumption in Faktenbasen.
6. Aktionsregeln als Verallgemeinerung von STRIPS-Operatoren.
7. Diskussion der Implementation.