Datenbanken

—

Berechnung von Normalformen

Thomas Studer

Institut für Informatik
Universität Bern

Repetition: Logische Folgerung

Seien

- $\mathcal{S} = (A_1, \ldots, A_n)$ ein Schema,
- F eine Menge von funktionalen Abhängigkeiten über $\mathcal{S}$ und
- $X \to Y$ eine weitere funktionale Abhängigkeit über $\mathcal{S}$.

Wir sagen, $X \to Y$ *folgt logisch aus* F, in Zeichen

$$F \models X \to Y \quad ,$$

falls jede Instanz R von $\mathcal{S}$, welche alle Abhängigkeiten in F erfüllt, auch $X \to Y$ erfüllt.

Beispiel

Gegeben sei die Menge

$$F = \{W \to X,\ W \to Y,\ XY \to Z\}$$

von funktionalen Abhängigkeiten. Dann gilt unter anderem:

1. $F \models W \to X$,
2. $F \models W \to XY$,
3. $F \models W \to Z$.

Armstrong-Kalkül

Gegeben sei eine Menge F von funktionalen Abhängigkeiten über U. Dann wird $F \vdash X \to Y$ für $X, Y \subseteq U$ induktiv definiert durch:

1. Elemente von F.

   Ist $X \to Y$ ein Element von F, so gilt $F \vdash X \to Y$.

2. Reflexivität.

   Ist Y eine Teilmenge von X, so gilt $F \vdash X \to Y$.

3. Augmentation.

   Aus $F \vdash X \to Y$ und $Z \subseteq U$ folgt $F \vdash XZ \to YZ$.

4. Transitivität.

   Aus $F \vdash X \to Y$ und $F \vdash Y \to Z$ folgt $F \vdash X \to Z$.

Beispiel

Betrachte:

$$U := \{\texttt{Autor},\ \texttt{Jahrgang},\ \texttt{Titel}\}$$
$$F_1 := \{\texttt{Autor} \to \texttt{Jahrgang}\}\ .$$

Somit gilt mit Regel *Elemente von* F

$$F_1 \vdash \texttt{Autor} \to \texttt{Jahrgang}\ .$$

Dann folgt mit Regel *Augmentation* (für $Z = \{\texttt{Autor}, \texttt{Titel}\}$)

$$F_1 \vdash \{\texttt{Autor}, \texttt{Titel}\} \to \{\texttt{Autor}, \texttt{Jahrgang}, \texttt{Titel}\}\ .$$

Theoreme

Theorem (Korrektheit und Vollständigkeit)

Ist F eine Menge von funktionalen Abhängigkeiten über U, so gilt für alle $X, Y \subseteq U$

$$F \vdash X \to Y \quad \Longleftrightarrow \quad F \models X \to Y \ .$$

Theorem (Charakterisierung der Hülle $F+$)

Ist F eine Menge von funktionalen Abhängigkeiten über U, so gilt

$$F^+ = \{X \to Y \mid F \models X \to Y\} = \{X \to Y \mid F \vdash X \to Y\} \ .$$

Beispiel

Im vorherigen Beispiel haben wir im Armstrong-Kalkül

$$F_1 \vdash \{\texttt{Autor}, \texttt{Titel}\} \to \{\texttt{Autor}, \texttt{Jahrgang}, \texttt{Titel}\}$$

hergeleitet. Mit obigem Korollar folgt nun daraus

$$\{\texttt{Autor}, \texttt{Titel}\} \to \{\texttt{Autor}, \texttt{Jahrgang}, \texttt{Titel}\} \in F_1^+ \ .$$

Folgerungen

Theorem

Ist F eine Menge von funktionalen Abhängigkeiten über U, so gilt für alle $X, Y, Z \subseteq U$:

5. Vereinigung.

$$F \vdash X \to Y \text{ und } F \vdash X \to Z \quad \Longrightarrow \quad F \vdash X \to YZ.$$

6. Zerlegung.

$$F \vdash X \to Y \text{ und } Z \subseteq Y \quad \Longrightarrow \quad F \vdash X \to Z.$$

7. Einfachheit.

$$F \vdash X \to Y_1 Y_2 \ldots Y_n \quad \Longleftrightarrow \quad F \vdash X \to Y_i \text{ für alle } 1 \leq i \leq n.$$

Definition (Einfache funktionale Abhängigkeit)

Funktionale Abhängigkeiten mit einelementigen rechten Seiten werden als *einfache* funktionale Abhängigkeiten bezeichnet.

Attributhülle

Definition (Attributhülle X^+)

Gegeben seien eine Menge F von funktionalen Abhängigkeiten über der universellen Menge U sowie eine Menge $X \subseteq U$. Die *Attributhülle* X^+ *von* X *unter* F ist dann definiert durch

$$X^+ := \{A \in U \mid F \vdash X \to A\} \ .$$

Lemma

Lemma

Für alle Mengen F von funktionalen Abhängigkeiten über der universellen Menge U sowie für alle Mengen $X, Y \subseteq U$ gilt

$$F \vdash X \to Y \quad \Longleftrightarrow \quad Y \subseteq X^+ \ .$$

Beweis: Es sei Y die Attributmenge $\{A_1, A_2, \ldots, A_n\}$.

Von links nach rechts. Es gelte also $F \vdash X \to Y$. Aufgrund der Zerlegungseigenschaft folgt $F \vdash X \to A_i$ für alle $1 \leq i \leq n$. Dies ergibt $Y \subseteq X^+$.

Vno rechts nach links. Sei $Y \subseteq X^+$. Mit der Definition von X^+ folgt daraus $F \vdash X \to A_i$ für alle $1 \leq i \leq n$. Mit der Vereinigungseigenschaft erhalten wir also unmittelbar $F \vdash X \to Y$.

Algorithmus zur Berechnung von X^+

Gegeben seien eine universelle Menge U von Attributen, eine Menge F von funktionalen Abhängigkeiten über U sowie eine Menge $X \subseteq U$.

Eingabe: F, X

$R \ := \ X; \quad alt\_R \ := \ \emptyset;$

$\texttt{WHILE } R \neq alt\_R \texttt{ DO}$

$\quad alt\_R \ := \ R;$

$\quad \texttt{FOR EACH } V \to W \texttt{ IN } F \texttt{ DO}$

$\quad\quad \texttt{IF } V \subseteq R \texttt{ THEN } R \ := \ R \cup W$

Ausgabe: $X^+ \ := \ R$

Beispiel zum Algorithmus

Betrachte:

$$
\begin{aligned}
U &= \{A, B, C, D, E, F, G\} \\
F &= \{AC \to G,\ BD \to CE,\ E \to A,\ G \to BF\} \\
X &= \{C, E\}
\end{aligned}
$$

Iterative Berechnung von R:

$$
\begin{aligned}
&\{C, E\} \\
&\{C, E, A\} \\
&\{C, E, A, G\} \\
&\{C, E, A, G, B, F\} \quad = \quad X^+
\end{aligned}
$$

Überdeckungen

Definition

Gegeben seien Mengen F und G von funktionalen Abhängigkeiten über U.

1. G *überdeckt* F, in Zeichen $F \leq G$, falls $F^+ \subseteq G^+$.
2. F und G sind *äquivalent*, falls F und G sich gegenseitig überdecken, das heisst $\quad F \simeq G \quad :\Longleftrightarrow \quad F \leq G \wedge G \leq F \;$.

Überdeckungen

Lemma

Gegeben seien Mengen F und G von funktionalen Abhängigkeiten über U. Dann gilt $F \subseteq G^+ \iff F \leq G$.

Beweis: Rechts nach links: offensichtlich.

Links nach rechts: Sei $F \subseteq G^+$. Damit gilt

$$\begin{aligned} F^+ &= \{Y \to Z \mid F \vdash Y \to Z\} \\ &\subseteq \{Y \to Z \mid G^+ \vdash Y \to Z\} = (G^+)^+ = G^+ . \end{aligned}$$

Daraus folgt mit der vorhergehenden Definition sofort die Behauptung, dass $F \leq G$ gilt.

Test auf $F \leq G$

Eingabe: F, G.

Überprüfe für jedes $Y \to Z \in F$, ob $Y \to Z \in G^+$ durch:

  Berechne Y^+ unter G und teste $Z \subseteq Y^+$.

Falls dies immer der Fall ist,

Ausgabe: $F \leq G$; anderenfalls Ausgabe: $F \not\leq G$.

Lemma

Lemma

Jede Menge F von funktionalen Abhängigkeiten über U ist zu einer Menge G äquivalent, die nur einfache funktionale Abhängigkeiten enthält, d.h. funktionale Abhängigkeiten mit einelementigen rechten Seiten.

Beweis: Wir gehen aus von einer Menge F von funktionalen Abhängigkeiten über U und definieren

$$G_F \; := \; \{X \to A \mid \exists Y (X \to Y \in F \text{ und } A \in Y)\} \; .$$

Aufgrund der Zerlegungseigenschaft folgt $X \to A \in F^+$ für alle A und Y mit $A \in Y$ und $X \to Y \in F$. Folglich ist $G_F \subseteq F^+$.
Ist andererseits Y die Menge $\{A_1, A_2, \dots, A_n\}$ und gilt

$$X \to A_i \in G_F \quad \text{für alle } 1 \leq i \leq n,$$

so folgt mit der Vereinigungseigenschaft auch $X \to Y \in G_F^+$. Daraus folgt $F \subseteq G_F^+$.
$G_F \subseteq F^+$ und $F \subseteq G_F^+$ ergeben $F \simeq G_F$.
G_F ist eine Menge von einfachen funktionalen Abhängigkeiten.

Minimalität

Definition

Eine Menge F von funktionalen Abhängigkeiten über U heisst *minimal*, falls folgende Bedingungen erfüllt sind:

1. F enthält nur einfache funktionale Abhängigkeiten.
2. Keine funktionale Abhängigkeit in F ist redundant, d.h.

$$(\forall X \to A \in F)(F \setminus \{X \to A\} \not\equiv F) \ . \tag{M2}$$

3. Kein Attribut auf der linken Seite einer funktionalen Abhängigkeit aus F ist redundant, d.h.

$$(\forall X \to A \in F)(\forall Y \subsetneq X)((F \setminus \{X \to A\}) \cup \{Y \to A\} \not\equiv F) \tag{M3}$$

Definition

Gegeben sei eine Menge F von funktionalen Abhängigkeiten über U. Eine *minimale Überdeckung von F* ist eine minimale Menge von funktionalen Abhängigkeiten, welche zu F äquivalent ist.

Algorithmus für minimale Überdeckungen

Eingabe: Eine Menge F von funktionalen Abhängigkeiten über U.

1. Spalte alle nicht-einfachen funktionalen Abhängigkeiten in F auf. Nenne die neue Menge F'.
2. Entferne sukzessive alle redundanten Attribute im Sinne von (M3) der vorhergehenden Definition aus F'. Nenne die neue Menge F''.
3. Entferne sukzessive alle redundanten funktionalen Abhängigkeiten im Sinne von (M2) der vorhergehenden Definition aus F''. Nenne die neue Menge F'''.

Ausgabe: $\texttt{MU}(F) := F'''$.

Beispiel

Attributmenge $\{A, B, C\}$ mit $F = \{ AB \to C, A \to AB, B \to A \}$.

1. Schritt: Die einzige nicht-einfache funktionale Abhängigkeit in F ist $A \to AB$. Wir spalten diese auf und erhalten so die Menge

$$F' = \{ AB \to C, A \to A, A \to B, B \to A \} \ .$$

2. Schritt: Wir testen, ob A redundant in $AB \to C$ ist. Dazu überprüfen wir, ob $C \in \{B\}^+$ unter F' gilt.
Wegen $\{B\}^+ = \{A, B, C\}$, ist dies der Fall. Daher ist A redundant in $AB \to C$ und wird gestrichen. Weitere redundante Attribute auf linken Seiten gibt es nicht. Also erhalten wir

$$F'' = \{ B \to C, A \to A, A \to B, B \to A \} \ .$$

3. Schritt: Wir entfernen alle redundanten funktionalen Abhängigkeiten aus F''. Offensichtlich gilt $A \in \{A\}^+$ unter $F'' \setminus \{A \to A\}$.
Folglich ist $A \to A$ in F'' redundant und wird entfernt. Weitere Redundanzen gibt es dann nicht mehr. Es folgt also

$$\mathtt{MU}(F) = \{ B \to C, A \to B, B \to A \} \ .$$

Bemerkung

Sei F eine Menge von funktionalen Abhängigkeiten. Dann kann es mehrere minimale Überdeckungen von F gegeben.
Betrachte die beiden Mengen von funktionalen Abhängigkeiten über der Attributmenge $\{A, B, C\}$:

$$F_1 := \{A \to B,\, B \to A,\, A \to C,\, C \to A\} \ ,$$
$$F_2 := \{A \to B,\, B \to C,\, C \to A\} \ .$$

Beide Mengen sind offensichtlich minimal und es gilt $F_1^+ = F_2^+$.

Repetition

1. jedes Schema kann verlustfrei in BCNF zerlegt werden;
2. jedes Schema kann verlustfrei und abhängigkeitserhaltend in 3NF zerlegt werden.

Algorithmus für die BCNF-Zerlegung

Gegeben seien ein Relationenschema $\mathcal{S}$ mit einer Menge von funktionalen Abhängigkeiten F.

Eingabe: $\mathcal{S},\ F$

$\mathcal{Z} := \{\mathcal{S}\}$

`WHILE` es gibt $\mathcal{S}_i \in \mathcal{Z}$ mit $\mathcal{S}_i$ nicht in BCNF bez. $\Pi_{\mathcal{S}_i}(F)$ `DO`

wähle ein solches $\mathcal{S}_i$

wähle $X \to A \in F^+$ mit

$$X \subseteq \mathcal{S}_i,\ A \in \mathcal{S}_i,\ A \notin X,\ X \to \mathcal{S}_i \notin F^+$$

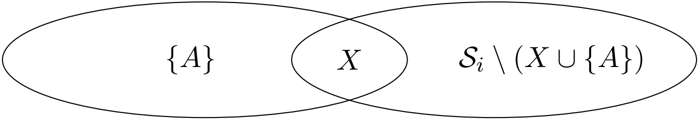
$$\mathcal{Z} := \big(\mathcal{Z} \setminus \{\mathcal{S}_i\}\big) \cup \big\{X \cup \{A\}, \mathcal{S}_i \setminus \{A\}\big\}$$

Ausgabe: $\mathcal{Z}$

Die so erhaltene Ausgabemenge $\mathcal{Z}$ ist dann die gewünschte Zerlegung von $\mathcal{S}$ in BCNF bezüglich F.

Bild

Wähle ein Schema $\mathcal{S}_i$ welches die BCNF Bedingungen verletzt bezüglich einer funktionalen Abhängigkeit $X \to A$. Das Schema $\mathcal{S}_i$ wird dann wie folgt in zwei Schemata zerlegt:



Aus $X \to A \in F^+$ folgt, dass diese Zerlegung von $\mathcal{S}_i$ in $\{X \cup \{A\}, \mathcal{S}_i \setminus \{A\}\}$ verlustfrei ist. Dieser Schritt wird solange wiederholt, bis alle Schemata der Zerlegung in BCNF sind.

Beispiel

Wir betrachten

$$\mathcal{S}_3 = (\mathtt{Stadt}, \mathtt{Str}, \mathtt{PLZ})$$
$$F_3 = \big\{ \{\mathtt{Stadt},\ \mathtt{Str}\} \to \{\mathtt{PLZ}\}, \quad \{\mathtt{PLZ}\} \to \{\mathtt{Stadt}\} \big\} \ .$$

Wir beginnen nun mit $\mathcal{Z} := \{\mathcal{S}_3\}$. Es gilt $\mathcal{S}_3 \in \mathcal{Z}$ ist nicht in BCNF bezüglich F_3. Wir wählen nun eine funktionale Abhängigkeit $X \to A \in F_3^+$, welche die BCNF Bedingungen verletzt. Wir setzen

$$X := \{\mathtt{PLZ}\} \subseteq \mathcal{S}_3 \quad \text{und} \quad A := \mathtt{Stadt} \in \mathcal{S}_3 \ .$$

Es gilt somit $A \notin X$ und $X \to \mathcal{S}_3 \notin F_3^+$. Wir setzen $\mathcal{Z}$ neu auf

$$\big\{(\mathtt{Stadt}, \mathtt{PLZ}),\ (\mathtt{Str}, \mathtt{PLZ})\big\} \ .$$

Jetzt sind alle Elemente von $\mathcal{Z}$ in BCNF und der Algorithmus gibt die Zerlegung $\mathcal{Z}$ als Resultat zurück.

Algorithmus für die 3NF-Zerlegung

Gegeben Schema $\mathcal{S}$ und *minimale* Menge F von funktionalen Abhängigkeiten. Für jede Abhängigkeit $X \to A \in F$ definieren wir ein Schema $\mathcal{S}_{X \to A}$, so dass $\mathcal{S}_{X \to A} = X \cup \{A\}$.

Eingabe: $\mathcal{S}, F$

$\mathcal{Z} := \{\mathcal{S}_{X \to A} \mid X \to A \in F\}$

`IF` kein $\mathcal{S}_{X \to A} \in \mathcal{Z}$ enthält Schlüssel für $\mathcal{S}$ bez. F `THEN`

wähle Schema $\mathcal{K}$, welches Schlüssel für $\mathcal{S}$ ist

$\mathcal{Z} := \mathcal{Z} \cup \{\mathcal{K}\}$

Ausgabe: $\mathcal{Z}$

Die so erhaltene Ausgabemenge $\mathcal{Z}$ ist dann die gewünschte Zerlegung von $\mathcal{S}$ in 3NF bezüglich F.

Beispiel

Betrachte: $$\mathcal{S}_2 = (\,\texttt{BuchId}, \texttt{Autor}, \texttt{Jahrgang}, \texttt{Titel}\,)$$
$$F_2 = \big\{\{\texttt{BuchId}\} \to \{\texttt{Autor}, \texttt{Jahrgang}, \texttt{Titel}\}, \{\texttt{Autor}\} \to \{\texttt{Jahrgang}\}\big\}\ .$$

Zuerst berechnen wir die minimale Überdeckung $\texttt{MU}(F_2)$ von F_2.

1. Durch Aufspalten erhalten wir
$$F_2' := \{\,\texttt{BuchId} \to \texttt{Autor}, \texttt{BuchId} \to \texttt{Jahrgang} \\ \texttt{BuchId} \to \texttt{Titel}, \texttt{Autor} \to \texttt{Jahrgang}\,\}\ .$$
2. Redundante Attribute entfernen. Die linken Seiten sind alle ein-elementig. Somit haben wir $F_2'' := F_2'$.
3. Redundante Abhängigkeiten entfernen. Wir finden
$$\texttt{Jahrgang} \in \texttt{BuchId}^+ \text{ unter } F_2'' \setminus \{\texttt{BuchId} \to \texttt{Jahrgang}\}\ .$$
Somit gilt $F_2'' \setminus \{\texttt{BuchId} \to \texttt{Jahrgang}\} \simeq F_2''$ und wir setzen
$$\texttt{MU}(F_2) := F_2''' := F_2'' \setminus \{\texttt{BuchId} \to \texttt{Jahrgang}\}\ .$$

Beispiel

Als Input für den 3NF-Algorithmus verwenden wir nun $\mathcal{S}_2$ und $\mathtt{MU}(F_2)$.

$$\mathcal{S}_2 = (\,\mathtt{BuchId},\, \mathtt{Autor},\, \mathtt{Jahrgang},\, \mathtt{Titel}\,)$$
$$\mathtt{MU}(F_2) := \{\,\mathtt{BuchId} \to \mathtt{Autor},\, \mathtt{BuchId} \to \mathtt{Titel},\, \mathtt{Autor} \to \mathtt{Jahrgang}\,\}$$

Im ersten Schritt erhalten wir so die Zerlegung

$$\mathcal{Z} = \left\{\, (\mathtt{BuchId}, \mathtt{Autor}),\, (\mathtt{BuchId}, \mathtt{Titel}),\, (\mathtt{Autor}, \mathtt{Jahrgang})\,\right\} \; .$$

Das Schema $(\mathtt{BuchId}, \mathtt{Autor}) \in \mathcal{Z}$ enthält einen Schlüssel für $\mathcal{S}_2$ bezüglich $\mathtt{MU}(F_2)$. Somit müssen wir im zweiten Schritt kein Schema hinzufügen.

Damit ist $\mathcal{Z}$ eine verlustfreie und abhängigkeitserhaltende Zerlegung des Schemas $\mathcal{S}_2$ in die dritte Normalform.

Beispiel : Schlüssel hinzufügen

Betrachte

$$\mathcal{S} := \{\underline{A}, \underline{B}, C\}$$
$$F := \{B \to C\}$$

Somit ist $\{A, B\}$ ein Schlüssel für das Schema $\mathcal{S}$

Nach Schritt 1 des Algortithmus haben wir

$$\mathcal{Z} = \{\ (B, C)\ \}\ \ .$$

Wir sehen:

- $\mathcal{Z}$ ist keine Zerlegung.
- $\mathcal{Z}$ enthält kein Schema, das einen Schlüssel für $\mathcal{S}$ enthält.

Nach Schritt 2 des Algortithmus haben wir

$$\mathcal{Z} = \{\ (B, C)\,(A, B)\ \}\ \ .$$