7.1.2 Membership-Test - fortgesetzt

Membership-Test:

$$X \to Y \in \mathcal{F}^+?$$

(Attribut-)Hülle X^+ von X (bzgl. $\mathcal{F}$)

$$X^+ = \{A \mid A \in V \text{ und } X \to A \in \mathcal{F}^+\}.$$

Membership-Test Variante 2:

Berechne zunächst X^+ mittels (A6) - (A8) und teste anschließend, ob $Y \subseteq X^+$.

XPlus-Algorithmus

XPlus($X, Y, \mathcal{F}$) boolean {
  result := X;
  WHILE (changes to result) DO
    FOR each $X' \to Y' \in \mathcal{F}$ DO
      IF ($X' \subseteq$ result) THEN result := result $\cup$ Y';
    end.
  IF ($Y \subseteq$ result) RETURN true ELSE false;
}

Der XPlus-Algorithmus hat eine Laufzeit, die polynomiell in der Darstellung von $\mathcal{F}$ ist.

Beispiel XPlus-Algorithmus

Sei $V = \{A, B, C, D, E, F, G, H, I\}$ und
$\mathcal{F} = \{AB \to E, BE \to I, E \to G, GI \to H\}$.

Es soll getestet werden, ob $AB \to GH \in \mathcal{F}^+$.

Axiom	Anwendung	`result`
(A7)	$AB \to AB$	$\{A, B\}$
(A8)	$AB \to ABE$	$\{A, B, E\}$
(A8)	$AB \to ABEI$	$\{A, B, E, I\}$
(A8)	$AB \to ABEIG$	$\{A, B, E, I, G\}$
(A8)	$AB \to ABEIGH$	$\{A, B, E, I, G, H\}$
(A6)	$AB \to GH$	

Schlüssel - jetzt formal definiert:

Sei $V = \{A_1, \dots, A_n\}$. $X \subseteq V$ heißt *Schlüssel* für V (bzgl. $\mathcal{F}$), wenn

- $X \to A_1 \dots A_n \in \mathcal{F}^+$,
- $Y \subset X \Rightarrow Y \to A_1 \dots A_n \notin \mathcal{F}^+$.

Basierend auf dem XPlus-Algorithmus können wir zu gegebenen $V, \mathcal{F}$ einen Schlüssel berechnen.

(1) Beginne mit $X := V$.

(2) Für jedes $A \in V$: falls $(X \setminus \{A\})^+ = V$, dann $X := X \setminus \{A\}$.

(3) X ist ein Schlüssel.

Bemerkung

- Jedes $A \in V$ wird nur einmal betrachtet; der XPlus-Algorithmus wird n-mal aufgerufen.
- Sofern mehrere Schlüssel existieren (Beispiel: $V = \{\texttt{Stadt, Adresse, PLZ}\}$, $\mathcal{F} = \{\texttt{Stadt Adresse} \to \texttt{PLZ}, \texttt{PLZ} \to \texttt{Stadt}\}$), wird nur einer dieser berechnet.
- Der Test, ob eine Attributemenge Schlüssel ist, ist exponentiell (NP-vollständig).

7.2 Verlustfreie und abhängigkeitsbewahrende Zerlegungen

- Sei ein Relationsschema R gegeben durch eine Attributmenge V und eine Menge funktionaler Abhängigkeiten $\mathcal{F}$.
- Sei $\rho = \{X_1, \ldots, X_k\}$ eine *Zerlegung* von V.
- ρ *verlustfrei, wenn:* Sei $r \in \mathsf{Sat}(V, \mathcal{F})$ und seien $r_i = \pi[X_i]r$, $1 \leq i \leq k$ die Projektionen von r auf die einzelnen Elemente der Zerlegung.

  r ist mittels $\bowtie$ aus den einzelnen Relationen der Zerlegung ρ exakt rekonstruierbar.

  ρ *abhängigkeitsbewahrend, wenn:* Die funktionalen Abhängigkeiten in $\mathcal{F}$ können auch über den Schemata der Zerlegung ρ ausgedrückt werden.

7.2.1 Verlustfreiheit

Sei $\rho = \{X_1, \ldots, X_k\}$ eine *Zerlegung* von V.

ρ heißt *verlustfrei*, wenn für jede Relation $r \in \mathrm{Sat}(V, \mathcal{F})$ gilt:

$$r = \pi[X_1]r \bowtie \ldots \bowtie \pi[X_k]r.$$

Beispiel

- Sei $V = \{A, B, C\}$ und $\mathcal{F} = \{A \to B, A \to C\})$.
- Sei beispielsweise $r \in Sat(V, \mathcal{F})$ wie folgt:

$$r = \begin{array}{ccc} A & B & C \\ \hline a_1 & b_1 & c_1 \\ a_2 & b_1 & c_2 \end{array}$$

- Seien $\rho_1 = \{AB, BC\}$ und $\rho_2 = \{AB, AC\}$.
- $r \subset \pi[AB]r \bowtie \pi[BC]r$,
  ρ_1 ist nicht verlustfrei.
- $r = \pi[AB]r \bowtie \pi[AC]r$,
  ρ_2 ist verlustfrei (für r).

Satz

Sei V eine Attributmenge mit einer Menge $\mathcal{F}$ funktionaler Abhängigkeiten. Sei $\rho = (X_1, X_2)$ eine Zerlegung von V.

ρ ist verlustfrei genau dann, wenn

$$(X_1 \cap X_2) \to (X_1 \setminus X_2) \in \mathcal{F}^+, \text{oder } (X_1 \cap X_2) \to (X_2 \setminus X_1) \in \mathcal{F}^+.$$

Korollar

Sei $R = (V, \mathcal{F})$ ein Relationsschema und sei $X \to Y \in \mathcal{F}$, wobei $X \cap Y = \emptyset$.

Dann ist die Zerlegung $\rho = (V \setminus Y, XY)$ verlustfrei.

Beweis: $(V \setminus Y) \cap XY = X$; $XY \setminus (V \setminus Y) = Y$.

7.2.2 Abhängigkeitsbewahrung

Beispiel

Sei $V = \{A, B, C, D\}$ und $\rho = \{AB, BC\}$.

- Betrachte $\mathcal{F} = \{A \to B, B \to C, C \to A\}$.

  Ist ρ abhängigkeitsbewahrend bzgl. $\mathcal{F}$?
- Betrachte $\mathcal{F}' = \{A \to B, B \to C, C \to B, B \to A\}$

  Ist ρ abhängigkeitsbewahrend bzgl. $\mathcal{F}'$?
- Mengen funktionaler Abhängigkeiten heißen *äquivalent* genau dann, wenn ihre Hüllen gleich sind.
- Offensichtlich $\mathcal{F} \equiv \mathcal{F}'$ und somit ρ abhängigkeitsbewahrend sowohl bzgl. $\mathcal{F}$ als auch bzgl. $\mathcal{F}'$.

Definition

- Sei $R = (V, \mathcal{F})$ gegeben. Sei weiter $Z \subseteq V$.
- Sei die *Projektion* von $\mathcal{F}$ auf Z definiert zu

$$\pi[Z]\mathcal{F} = \{X \to Y \in \mathcal{F}^+ \mid XY \subseteq Z\}.$$

- Eine Zerlegung $\rho = \{X_1, \ldots, X_k\}$ von V heißt *abhängigkeitsbewahrend* bzgl. $\mathcal{F}$, wenn

$$\bigcup_{i=1}^{k} \pi[X_i]\mathcal{F} \equiv \mathcal{F}.$$

zum vorangehenden Beispiel

Sei $V = \{A, B, C, D\}$, $\rho = \{AB, BC\}$ und $\mathcal{F} = \{A \to B, B \to C, C \to A\}$.
Ist ρ abhängigkeitsbewahrend bzgl. $\mathcal{F}$?
Ja, denn

- $\{A \to B, B \to A\} \subseteq \pi[AB]\mathcal{F}$,
- $\{B \to C, C \to B\} \subseteq \pi[BC]\mathcal{F}$ und
- $\{A \to B, B \to A\} \cup \{B \to C, C \to B\} \equiv \mathcal{F}$.

Beobachtung: Nicht jede verlustfreie Zerlegung ist abhängigkeitsbewahrend!

- $R = (V, \mathcal{F})$, wobei $V = \{\texttt{Stadt, Adresse, PLZ}\}$,
- $\mathcal{F} = \{\texttt{Stadt Adresse} \rightarrow \texttt{PLZ}, \texttt{PLZ} \rightarrow \texttt{Stadt}\}$.
- $\rho = \{X_1, X_2\}$: $X_1 = \{\texttt{Adresse, PLZ}\}$ und $X_2 = \{\texttt{Stadt, PLZ}\}$.
- ρ ist verlustfrei, da $(X_1 \cap X_2) \rightarrow (X_2 \setminus X_1) \in \mathcal{F}$.
- ρ ist nicht abhängigkeitsbewahrend.

`Stadt Adresse` und `Adresse PLZ` sind Schlüssel zu R.

7.3 Normalformen

Ist dies ein guter Schemaentwurf?

Stadt

SNr	SName	LCode	LFläche
7	Freiburg	D	357
9	Berlin	D	357
40	Moscow	RU	17075
43	St.Petersburg	RU	17075

Kontinent

KName	LCode	KFläche	Prozent
Europe	D	3234	100
Europe	RU	3234	20
Asia	RU	44400	80

- Anomalie beim Einfügen: Es können nur Länder aufgenommen werden, zu denen auch Städte existieren.
- Anomalie beim Löschen: Werden Städte gelöscht, können u.U. alle Informationen über gewisse Länder verloren gehen.
- Anomalie beim Ändern: Änderungen der Fläche eines Landes müssen bei mehreren Zeilen vorgenommen werden.

Sei $R = (V, \mathcal{F})$ ein Schema. Wir wollen eine Zerlegung $\rho = (X_1, \ldots, X_k)$ von R finden, die die folgenden Eigenschaften erfüllt:

- jedes $R_i = (X_i, \pi[X_i]\mathcal{F})$, $1 \leq i \leq k$ ist in einer gewünschten Normalform,
- ρ ist verlustfrei und (möglichst) auch abhängigkeitsbewahrend,
- k minimal.

Begriffe

- Sei X Schlüssel zu R und $X \subseteq Y \subseteq V$, dann nennen wir Y *Superschlüssel* von R.
- Gilt $A \in X$ für irgendeinen Schlüssel X von R, so heißt A *Schlüsselattribut (SA)* in R;
- gilt $A \notin X$ für jeden Schlüssel X, so heißt A *Nicht-Schlüsselattribut (NSA)*.

3. Normalform

Ein Relationsschema $R = (V, \mathcal{F})$ ist in *3. Normalform* (3NF) genau dann, wenn jedes NSA $A \in V$ die folgende Bedingung erfüllt.

Wenn $X \to A \in \mathcal{F}$, $A \notin X$, dann ist X ein Superschlüssel.

Die Bedingung der 3NF verbietet nichttriviale funktionale Abhängigkeiten $X \to A$, in denen ein NSA A in der Weise von einem Schlüssel K transitiv funktional abhängt, dass $K \to X$, $K \not\subseteq X$ gilt und des Weiteren $X \to A$.

Welche Art von Redundanz wird so vermieden?

Hinweis: Es existieren weitere, schärfere Definitionen von Normalformen in der Literatur.

Welche funktionale Abhängigkeiten verletzen die 3NF?

Stadt

SNr	SName	LCode	LFläche
7	Freiburg	D	357
9	Berlin	D	357
40	Moscow	RU	17075
43	St.Petersburg	RU	17075

Kontinent

KName	LCode	KFläche	Prozent
Europe	D	3234	100
Europe	RU	3234	20
Asia	RU	44400	80

Was ist hier zu sagen?

Stadt'

SNr	SName	LCode
7	Freiburg	D
9	Berlin	D
40	Moscow	RU
43	St.Petersburg	RU

Land'

LCode	LFläche
D	357
RU	17075

Lage'

LCode	KName	Prozent
D	Europe	100
RU	Europe	20
RU	Asia	80

Kontinent'

KName	KFläche
Europe	3234
Asia	44400

7.4 Minimale Überdeckung: Basis für Normalisierungsalgorithmen

- Seien $\mathcal{F}$ Mengen von funktionalen Abhängigkeiten.
- Wir suchen eine *"minimale" Überdeckung* von $\mathcal{F}$.

  $\mathcal{G}$ überdeckt $\mathcal{F}$, wenn $\mathcal{F} \subseteq \mathcal{G}^+$.
- Strategie: Bilde $\mathcal{G}$ durch Streichen von Attributen in den FAs von $\mathcal{F}$ oder Entfernen von FAs in $\mathcal{F}$ in einer Weise, die die Äquivalenz nicht zerstört.

Beispiel Rechtsreduktion

- $\mathcal{F}_1 = \{A \to B, B \to A, B \to C, A \to C, C \to A\}$.

  Kann die FA $B \to C$ zu $B \to \emptyset$ reduziert werden, d.h. gestrichen werden?

  Sei $\mathcal{F}_1' = \{A \to B, B \to A, A \to C, C \to A\}$.

  Gilt $\mathcal{F}_1^+ = \mathcal{F}_1'^+$? Ja, denn

  (a) $\mathcal{F}_1^+ \subseteq \mathcal{F}_1'^+$ wegen $XPlus(B, C, \mathcal{F}_1')$.
  (b) $\mathcal{F}_1^+ \supseteq \mathcal{F}_1'^+$ wegen $\mathcal{F}_1 \supseteq \mathcal{F}_1'$.

Beispiel Linksreduktion

- $\mathcal{F}_2 = \{AB \to C, A \to B, B \to A\}$.

  Kann die FA $AB \to C$ zu $B \to C$ reduziert werden, d.h. $AB \to C$ durch $B \to C$ ersetzt werden?

  Sei $\mathcal{F}_2' = \{B \to C, A \to B, B \to A\}$.

  Gilt $\mathcal{F}_2^+ = \mathcal{F}_2'^+$? Ja, denn

  (a) $\mathcal{F}_2^+ \subseteq \mathcal{F}_2'^+$ wegen (A2) und (A6) angewendet auf $B \to C$.
  (b) $\mathcal{F}_2^+ \supseteq \mathcal{F}_2'^+$ wegen $XPlus(B, C, \mathcal{F}_2)$.

Links- und Rechtsreduktion

- Eine Menge $\mathcal{F}$ funktionaler Abhängigkeiten heißt *linksreduziert*, wenn sie die folgende Eigenschaft erfüllt.

  Wenn $X \to Y \in \mathcal{F}, Z \subseteq X$, dann $\mathcal{F}' = (\mathcal{F} \setminus \{X \to Y\}) \cup \{Z \to Y\}$ nicht äquivalent zu $\mathcal{F}$.
- *Linksreduktion:* ersetze $X \to Y$ in $\mathcal{F}$ durch $Z \to Y$.
- Sie heißt *rechtsreduziert*, wenn $X \to Y \in \mathcal{F}, Z \subseteq Y$, dann $\mathcal{F}' = (\mathcal{F} \setminus \{X \to Y\}) \cup \{X \to Z\}$ nicht äquivalent zu $\mathcal{F}$.
- *Rechtsreduktion:* ersetze $X \to Y$ in $\mathcal{F}$ durch $X \to Z$.

Entscheidung mittels XPlus-Algorithmus

- Sei $X \to Y$ eine Abhängigkeit in $\mathcal{F}$ und sei $Z \to Y$, wobei $Z \subseteq X$.

  Wir führen die entsprechende Linksreduktion durch, wenn $\mathit{XPlus}(Z, Y, \mathcal{F})$ das Ergebnis `true` liefert.
- Sei $X \to Y$ eine Abhängigkeit in $\mathcal{F}$ und sei $X \to Z$, wobei $Z \subseteq Y$.

  Wir führen die entsprechende Rechtsreduktion durch, wenn $\mathit{XPlus}(X, Y, \mathcal{F}')$ das Ergebnis `true` liefert.

Satz

Sei eine Menge funktionaler Abhängigkeiten $\mathcal{F}$ gegeben und sei $\mathcal{F}'$ aus $\mathcal{F}$ durch eine Links- oder Rechtsreduktion hervorgegangen.

Dann $\mathcal{F} \equiv \mathcal{F}'$.

minimale Überdeckung

Eine Menge funktionaler Abhängigkeiten $\mathcal{F}^{min}$ ist eine *minimale Überdeckung* zu $\mathcal{F}$, wenn wir sie durch Anwendung der folgenden Schritte erzeugen können:

- Führe alle möglichen Linksreduktionen durch.
- Führe alle möglichen Rechtsreduktionen durch.
- Streiche alle trivialen funktionalen Abhängigkeiten der Form $X \to \emptyset$.
- Vereinige alle funktionalen Abhängigkeiten mit gleicher linker Seite $X \to Y_1, \ldots, X \to Y_n$ zu einer einzigen FA der Form $X \to Y_1 \ldots Y_n$.

7.5 Algorithmus zur Normalisierung

3NF-Synthese: verlustfrei und abhängigkeitsbewahrend

Sei $R = (V, \mathcal{F})$ ein Relationsschema.

1. Sei $\mathcal{F}^{min}$ eine minimale Überdeckung zu $\mathcal{F}$.
2. Betrachte jeweils maximale Klassen von funktionalen Abhängigkeiten aus $\mathcal{F}^{min}$ mit derselben linken Seite. Seien $\mathcal{C}_i = \{X_i \to A_{i1}, X_i \to A_{i2}, \ldots\}$ die so gebildeten Klassen, $i \geq 0$.[1]
3. Bilde zu jeder Klasse $\mathcal{C}_i$, $i \geq 0$, ein Schema mit Format $V_{\mathcal{C}_i} = X_i \cup \{A_{i1}, A_{i2}, \ldots\}$.
4. Sofern keines der gebildeten Formate $V_{\mathcal{C}_i}$ einen Schlüssel für R enthält, berechne einen Schlüssel für R. Sei Y ein solcher Schlüssel. Bilde zu Y ein Schema mit Format $V_K = Y$.
5. $\rho = \{V_K, V_{\mathcal{C}_1}, V_{\mathcal{C}_2}, \ldots\}$ ist eine verlustfreie und abhängigkeitsbewahrende Zerlegung von R in 3NF.

[1] Der von uns betrachtete Algorithmus zur Berechnung von $\mathcal{F}^{min}$ hat diese Klassenbildung bereits vorgenommen.

7.6 empfohlene Lektüre

SYNTHESIZING INDEPENDENT DATABASE SCHEMAS [1)]

JOACHIM BISKUP[2)]
Lehrstuhl für Angewandte Mathematik
insbesondere Informatik
RWTH Aachen
D-5100 Aachen, Germany

UMESHWAR DAYAL[3)] PHILIP A. BERNSTEIN[3)]
Aiken Computational Laboratory
Harvard University
Cambridge, MA 02138

Abstract

We study the following database design problem. Given a universal relation scheme $<U, \mathcal{F}>$ where $\mathcal{F}$ is a set of functional dependencies, find an in some way normalized database schema $\mathcal{D} = \{<X_1, \mathcal{F}_1>, \ldots, <X_n, \mathcal{F}_n>\}$ where $X_i \subset U$ and $\mathcal{F}_i$ is inherited from $\mathcal{F}$, such that $\mathcal{D}$ is an independent representation of the universal scheme $<U, \mathcal{F}>$. This means that $\mathcal{D}$ has both the lossless join property and the faithful closure property, $(\bigcup_{i=1}^{n} \mathcal{F}_i)^+ = \mathcal{F}^+$, where $^+$ denotes the closure of a set of functional dependencies. We show that this goal can easily be achieved by an extension of the well-known synthetic approach of Bernstein and others to database design. We merely have to check whether the usual synthesis procedure has produced a key component $<X_i, \mathcal{F}_i>$ such that $X_i \to U \in \mathcal{F}^+$; in case this is true the output of the synthesis procedure is actually an independent (and not only faithful) representation, otherwise we only have to add one further component, namely just a key. These claims are proved by a careful inspection of the Aho/Beeri/Ullman algorithm to test for losslessness. Finally, we show how to use our method to synthesize minimal independent third normal form schemas.

1. Introduction

In this section we informally discuss the problems addressed in this paper. Precise definitions and explanation of notation will be given in section 2.

The papers of Codd [6], [7] on the relational model of databases and database normalization were a starting point for an extensive study of formal methods to relational database design. This work is surveyed in [3], and the reader is referred to this survey for further background.

Essentially, there are two different approaches to formal database design. Both methods use a so-called universal database schema $<U, \mathcal{C}>$ as input, where U is the set of all attributes that we are interested in and $\mathcal{C}$ is a set of semantic constraints

1) This paper was originally submitted to SIGMOD under the sole authorship of Biskup and was refereed in the form submitted. It was discovered by the program chairman (Bernstein) that the identical result had been published in November 1978 in a technical report by himself and Dayal [8]. This information was not made available to the referees or to the program committee until after the paper had been accepted for publica-

[2]

[2] In: Proceeding SIGMOD '79 Proceedings of the 1979 ACM SIGMOD international conference on Management of data. Kann gegoogelt werden.