Unit 0x08

Normalformen II

Kanonische Überdeckung

Motivation/Erinnerung

Aufgabe

- Entwicklung einer 'guten' Datenbank […] u.a. minimal, redundanzfrei […]

Vorgehen

- Konzeptueller Entwurf […] ER-Modells ✔
- Überführung in eine 'gute' Datenbank […] dazu benötigen wir:
  - Implementationsentwurf […] ➜ Relationales Modell ✔
  - Mathematisches Modell […] ➜ Relationale Algebra ✔
  - Gütemaß ➜ Normalformen
    - Kanonische Überdeckung (funktionale Abhängigkeiten ✔, Attributhülle ✔),
    - Synthesealgorithmus,
    - Dekompositionsalgorithmus

Äquivalente Menge funktionaler Abhängigkeiten

Motivation

- Gegeben sei ein Schema R={A,B,C,D} mit FD={ A→B, B→C, AB→CD }

Kann man die FDs 'vereinfachen'?

- 'vereinfachen' bedeuten i.d.R., so viel wie möglich zu streichen, ohne die Bedeutung zu ändern bzw. hier einen 'äquivalenten' ('so gut wie') Ausdruck mit *weniger* oder *'kleineren'* funktionalen Abhängigkeiten zu erhalten. 'kleiner' bedeutet wiederum, Attribute auf den *rechten und linken Seiten* der funkt. Abhängigkeiten weglassen zu können ➜ Rechts- und Linksreduktion (folgen).
- Hier: Da A→B gilt, ist AB→CD 'so gut wie' A→CD, und daher FD'={ A→B, B→C, A→CD } 'äquivalent' zu obiger Menge FDs.

Q&A

- Wie formal konkretisieren?

Äquivalente Menge funktionaler Abhängigkeiten

Definition

- Zu einem Schema R und einer Menge FD funktionaler Abhängigkeiten bezeichne
  - **FD^+**

  die Menge aller funktionalen Abhängigkeiten, die von FD impliziert werden.
- Anstatt die Armstrong-Axiome zu bemühen, überlege man sich, dass gilt
  - $\alpha \to \beta \in FD^+$ genau dann, wenn $\beta \subseteq$ AttrHülle(FD,α).
- Zwei Mengen funktionaler Abhängigkeiten FD_1 und FD_2 heißen **äquivalent**, falls
  - $FD_1^+ = FD_2^+$, in Zeichen $FD_1 \equiv FD_2$.
- Die kanonische Überdeckung ist in einem gewissen Sinne eine *minimale äquivalente* Menge funktionaler Abhängigkeiten (Definition folgt).
- Technisch: Manchmal schreiben wir auch etwas ungenau $\beta \in$ AttrHülle(FD,α).

Äquivalente Menge funktionaler Abhängigkeiten

Beispiel FD^+

- Geg. sei Schema R={A, B, C, D}, FD={ A→B, B→C, AB→CD }, dann ist
  - FD^+={ A→ABCD, B→BC, C→C, D→D, AB→ABCD, AC→ABCD, AD→ABCD, BC→BC, BD→BD, CD→CD, ABC→ABCD, ABD→ABCD, BCD→BCD, ABCD→ABCD }

  (hier sind A→B und A→C zu A→BC etc. zusammengefasst).
- Spaltet man die nicht-trivialen Abhängigkeiten ab, ergeben sich die Darstellungen
  - FD^+={ A→BCD, B→C } ∪ { triviale Abhängigkeiten }, bzw.
  - FD^+={ A→BD, B→C } ∪ { triviale Abhängigkeiten } ∪ { transitive Abhängigkeiten }.
- FD^+ ergibt sich, perDefinition, durch die Betrachtung aller Attributhüllen bzw. der Vereinigung der dazugehöri-gen Abhängigkeiten.

Q&A

- Wie sieht wohl eine kleinste, äquivalente Menge aus…?

Äquivalente Menge funktionaler Abhängigkeiten

Beispiel Äquivalenz

- Geg. sei Schema R={A, B, C, D}, FD={ A→B, B→C, AB→CD }, dann ist
  - FD'={ A→B, B→C, A→CD } *äquivalent* zu FD, da gilt $FD'^{+}=FD^{+}$ (nachrechnen), oder aber durch Betrachtung der im Vergleich *relevanten* Attributhüllen, also hier
    - CD⊆AttrHülle(FD,AB), und
    - CD⊆AttrHülle(FD',A).
- *Anders ausgedrückt:* Da die Attributhüllen in beiden Fällen gleich sind, sind auch die funktionalen Abhängigkeiten, die sich einmal aus AB→CD und einmal aus A→CD ergeben, dieselben. Die restlichen funktionalen Abhängigkeiten sind unberührt, also gilt wiederum $FD'^{+}=FD^{+}$.
- *Generell:* Wenn wir Attribute weglassen, so untersuchen wir eine Menge FD', primär gegeben durch FD' = FD \ {untersuchte funkt. Abh.} ∪ {modifizierte funkt. Abh.}.

Äquivalente Menge funktionaler Abhängigkeiten

Äquivalente Minimierung

- Wir möchten eine 'kleinere' aber äquivalente Menge mit weniger Attributen finden.

  D.h. wir versuchen, Attribute wegzulassen und sehen, ob das, was übrig bleibt, äquivalent ist. Das kann man für die rechten und linken Seiten machen.
- Geg. sei Schema R={A, B, C, D}, FD={ A→B, B→C, AB→CD }. Wir starten *links* und betrachten nur die funktionalen Abhängigkeiten, wo wir links auch Attribute weglassen können. Folglich untersuchen wir, ob wir AB→CD durch A→CD oder B→CD ersetzen können. Es ergibt sich *mit der Argumentation von zuvor*:
  - { A→B, B→C, A→CD } *äquivalent* zu FD, da CD⊆AttrHülle(FD,A).
  - { A→B, B→C, B→CD } *nicht äquivalent* zu FD, da: CD⊈AttrHülle(FD,B)=BC.

Da wir links Attribute weggelassen haben, ist es eine Linksreduktion.

In den Überlegungen haben wir in der Attributhülle das zweite Argument variiert.

Äquivalente Menge funktionaler Abhängigkeiten

Äquivalente Minimierung

- Geg. sei das reduzierte Schema R={A, B, C, D}, FD={ A→B, B→C, A→CD }. Wir testen nun Attribute auf der *rechten* Seite. Folglich untersuchen wir u.a., ob wir A→CD durch A→C oder A→D äquivalent ersetzen können.
  - FD'={ A→B, B→C, A→D } ist *äquivalent* zu FD, da C⊆AttrHülle(FD',A).

    Wir folgern also C aus den *anderen* funktionalen Abhängigkeiten und können es somit hier rechts auch weglassen.
  - FD''={ A→B, B→C, A→C } ist *nicht äquivalent* zu FD, da D⊈AttrHülle(FD'',A).

    Wir können hier D *nicht* aus den *anderen* funktionalen Abhängigkeiten folgern und es somit auch hier rechts nicht weglassen.

Da wir rechts Attribute weggelassen haben, ist es eine Rechtsreduktion.

In den Überlegungen haben wir in der Attributhülle das erste Argument variiert.

Äquivalente Menge funktionaler Abhängigkeiten

Äquivalente Minimierung

- Das ursprüngliche Schema R={A, B, C, D}, FD={ A→B, B→C, AB→CD } haben wir durch *Links- und Rechtsreduktion* in eine äquivalente und kürzere Darstellung
  - { A→B, B→C, A→D }

  überführt.
- Fasst man die funktionalen Abhängigkeiten noch zusammen (Vereinigung), ergibt sich die **kanonische Überdeckung:**
  - { A→BD, B→C }. (vgl. Frage zu Beginn)
- In dem Beispiel haben wir nur einzelne funktionale Abhängigkeiten von FD betrachtet. Allgemein muss man bei der kanonischen Überdeckung *alle* betrachten.

Kanonischen Überdeckung

Definition

- Zu einem Schema R und einer Menge FD funktionaler Abhängigkeiten bezeichne
  - **FD^C**

  die **kanonischen Überdeckung** von FD, falls folgende Kriterien erfüllt sind:
  - $FD^C \equiv FD$, d.h. $FD^{C+} = FD^{+}$.
  - Alle $\alpha \to \beta \in FD^C$ sind minimal in dem Sinne, dass
    - $\forall a \in \alpha$ gilt: $FD^C \setminus (\alpha \to \beta) \cup (\alpha\text{-}a \to \beta) \not\equiv FD^C$
    - $\forall b \in \beta$ gilt: $FD^C \setminus (\alpha \to \beta) \cup (\alpha \to \beta\text{-}b) \not\equiv FD^C$
  - alle linken Seiten kommen nur einmal vor (einzigartig, Vereinigungsregel).

Algorithmus zur Berechnung der Kanonischen Überdeckung

Berechnung der kanonischen Überdeckung FD^C

- Gegeben sind ein Schema R und einer Menge FD funktionaler Abhängigkeiten.
- **Linksreduktion:** Für alle $\alpha \rightarrow \beta \in$ FD überprüfe, ob ein $a \in \alpha$ überflüssig ist, d.h. ob
  - $\beta \subseteq$ AttrHülle(FD,α-a).

  gilt. In diesem Fall wird $\alpha \rightarrow \beta$ durch α-a$\rightarrow \beta$ in FD ersetzt.
- **Rechtsreduktion:** Für alle $\alpha \rightarrow \beta \in$ FD überprüfe, ob ein $b \in \beta$ überflüssig ist, d.h. ob
  - $\beta \subseteq$ AttrHülle(FD \ $\{\alpha \rightarrow \beta\} \cup \{\alpha \rightarrow \beta$-b$\}$, α)

  gilt. In diesem Fall wird ebenfalls $\alpha \rightarrow \beta$ durch $\alpha \rightarrow \beta$-b in FD ersetzt.
- **Entfernung** mglw. entstandener $\alpha \rightarrow \emptyset$.
- **Vereinigung** mglw. entstandener $\alpha \rightarrow \beta_1$ und $\alpha \rightarrow \beta_2$ zu $\alpha \rightarrow \beta_1\beta_2$ (Vereinigung).

Algorithmus zur Berechnung der Kanonischen Überdeckung

Ein erstes Beispiel…

- R={ A, B, C, D, E, F },
  FD={ ABC→DEF,
  A→BC,
  D→EF }
- *Vorgehen*:
  - erst Links- dann Rechts-reduktion
  - jeweils Attribute weg-lassen und 'stur' die Bedingungen testen

B **Beispiel: Kanonische Überdeckung**

F **= {ABC→DEF, A→BC,D→EF}**

Linksreduktion

- **ABC→DEF**
  - **A** überflüssig? **Nein**, denn **AttrHülle**(F, BC): {BC}.
  - **B** überflüssig? **Ja**, denn **AttrHülle**(F, AC): {AC} ➜ {ABC} ➜ {ABCDEF}
    - Erhalte F={AC→DEF, A→BC, D→EF} A→BC ABC→DEF
  - **C** überflüssig? **Ja**, denn **AttrHülle**(F,A): {A} ➜ {ABC} ➜ {ABCDEF}
    - Erhalte F={A→DEF, A→BC, D→EF} A→BC ABC→DEF

nicht verwechseln: Attribut F und Menge FD

gleiche Menge FD, veränderte Attribute

neu, aber äquivalent

Unterlagen Prof. Striegnitz

Algorithmus zur Berechnung der Kanonischen Überdeckung

Ein erstes Beispiel…

- R={ A, B, C, D, E, F },
  FD={ ABC→DEF,
  A→BC,
  D→EF }
- FD^C={ A→BCD,
  D→EF }

veränderte Menge FD, gleiche Attribute

B Beispiel: Kanonische Überdeckung

F **= {A→DEF, A→BC, D→EF}**

Rechtsreduktion

- **A→DEF**
  - **D** überflüssig? **Nein**, denn
    AttrHülle({A→EF,A→BC,D→EF}, A): {A}➔{AEF}➔{AEFBC}
  - **E** überflüssig? **Ja**, denn
    AttrHülle({A→DF,A→BC,D→EF}, A): {A}➔{ADF}➔{ADFBC}➔{ADFBCE}
    - Erhalte F={A→DF, A→BC, D→EF}
  - **F** überflüssig? **Ja**, denn
    AttrHülle({A→D, A→BC, D→EF}, A): {A}➔{AD}➔{ADBC}➔{ADBCEF}
    - Erhalte F={A→D, A→BC, D→EF}

Die anderen Regeln lassen sich nicht reduzieren ....
Es ergibt sich: *F*^{C}={A→BCD, D→EF}

Unterlagen Prof. Striegnitz

Algorithmus zur Berechnung der Kanonischen Überdeckung

Anmerkungen Teil I

- **Linksreduktion:** Hier sind nur die funktionalen Abhängigkeiten zu überprüfen, die rechts mind. *zwei Attribute* besitzen, denn eine Abhängigkeit der Form $\varnothing \rightarrow \beta$ (ein weggelassenes Attribut bei nur einem Attribut) macht keinen Sinn.
- **Rechtsreduktion:** Hier sind *alle* funktionalen Abhängigkeiten, auch $\alpha \rightarrow$b mit nur einem Attribut b rechts, zu überprüfen. Die veränderte Menge für die Attributhülle sieht dann formal so aus: FD \ {$\alpha \rightarrow$b} ∪ {$\alpha \rightarrow \varnothing$}…

  Wenn aber in der letzten funktionalen Abhängigkeit nichts aus α folgt, kann man diese Abhängigkeit auch direkt weglassen. Also insgesamt heisst das, man überprüft direkt ohne diese Abhängigkeit die Menge FD \ {$\alpha \rightarrow$b}.

  Wenn dann β=b aus α folgt und man die funktionale Abhängigkeit $\alpha \rightarrow$b direkt weglässt, dann entstehen erst gar keine keine Abhängigkeiten der Form $\alpha \rightarrow \varnothing$.

Algorithmus zur Berechnung der Kanonischen Überdeckung

Anmerkungen Teil II

- **Rechtsreduktion:** Man beachte, dass beispielsweise die Mengen
  - FD={ $\alpha \rightarrow \beta$, A→BC } und FD'={ $\alpha \rightarrow \beta$, A→B, A→C }

  offensichtlich äquivalent sind und daher die Rechtsreduktion generell etwas abgeändert werden kann (persönlicher Stil):
  - Anstatt FD zu modifizieren und A→BC erst durch A→C (Test, ob B notwendig) und dann durch A→B (Test C) zu ersetzen, kann man auch zunächst FD zu einer äquivalenten Menge FD' ändern, in der alle funktionalen Abhängigkeiten mit mehr als einem Attribut auf der rechten Seite aufgespalten werden (Dekomposition).
  - Das führt zu einer größeren Menge FD', die es zu untersuchen gilt, aber insgesamt bleibt die Menge der Tests mit der Attributhülle gleich und jeder einzelne Test ist etwas weniger komplex, weil man die untersuchte funktionale Abhängigkeit nicht modifiziert, sondern wegläßt. Im letzten Schritt werden alle wieder vereinigt.

Algorithmus zur Berechnung der Kanonischen Überdeckung

Anmerkungen Teil III

- **Beispiel Rechtsreduktion:** Die Alternativen sind
  - FD={ $\alpha\rightarrow\beta$, A→BC } und FD'={ $\alpha\rightarrow\beta$, A→B, A→C }

  Klassisch würde man u.a. untersuchen, ob $\beta \subseteq$AttrHülle(FD\\{$\alpha\rightarrow\beta$}∪{$\alpha\rightarrow\beta$-b},α) gilt, d.h.

  - B ⊆ AttrHülle(FD \ {A→BC} ∪ {A→C}, A) ?
    C ⊆ AttrHülle(FD \ {A→BC} ∪ {A→B}, A) ?

  In der anderen Variante untersucht man u.a.

  - C ⊆ AttrHülle(FD' \ {A→B}, A) ?
  - B ⊆ AttrHülle(FD' \ {A→C}, A) ?

Zunächst (FD') größer, aber Modifikation ist einfacher.

Q&A

- Ist die kanonische Überdeckung eindeutig?

Algorithmus zur Berechnung der Kanonischen Überdeckung

Beispiel

- R={ A, B, C, D },
  FD={ A→B,
  BD→C,
  ABC→AB }

Linksred.

$\alpha \to \beta$

$BD \to C$

Att. Hülle(FD, D) = D , C ∈ {D} Nein ↯

Att. Hülle(FD, B) = B   C ∈ {B} Nein ↯

$\alpha \to \beta$

$ABC \to AB$

Att. Hülle (FD, BC) = BC   AB ∈ {BC} ↯

Att. Hülle (FD, AC) = ACB   AB ∈ {ACB} ✓

also B überfl.

FD = { A→B, BD→C, AC→AB }

Att. Hülle (FD, A) = AB ✓

also C überfl.

FD = { A→B, BD→C, A→AB }

Algorithmus zur Berechnung der Kanonischen Überdeckung

Beispiel

- R={ A, B, C, D },
  $\boxed{FD^C=\{ A\rightarrow B,\ BD\rightarrow C \}}$

Rechtsred.

$\alpha \to \beta$
$A \to B$

Attr. Hülle $(\{BD \to C,\ A \to AB\},\ A) = AB$
$B \in \{AB\}$
also B überfl.

$FD = \{A \to \emptyset,\ BD \to C,\ A \to AB\}$
↑
weglassen

$\alpha \to \beta$
$BD \to C$

Attr. Hülle $(\{A \to AB\},\ BD) = BD$, $C \in \{BD\}$ ↯

$\alpha \to \beta$
$A \to AB$

Attr. Hülle $(\{BD \to C,\ A \to B\},\ A) = AB$, $AB \in \{AB\}$
also A überfl.

$FD = \{BD \to C,\ A \to B\}$

$\alpha \to \beta$
$A \to B$

Attr. Hülle $(\{BD \to C\},\ A) = A$, $B \in \{A\}$ ↯

Kanonische Überdeckung

Roter Faden – Wozu das alles?

- Die Kanonische Überdeckung ist der Hauptbestandteil des *Synthesealgorithmus*, der eine Relation in die sogenannte *dritte Normalform* (Gütekriterium) überführt.
- Relationen, die einer gewissen Normalform entsprechen, vermeiden bestimmte Anomalien.
- Anomalien bezeichnen Zustände in der Datenbank, bei denen Daten falsch bzw. inkonsistent sind, z.B. ermöglicht durch Redundanzen (Beispiel 'Heise' vs. 'Heihse').

Was folgt:

- *Teaser:* Artikeldatenbank inkl. Synthesealgorithmus und Normalformen
- *Unit 0x09:* Normalformen, Anomalien, Synthese- und Dekompositionsalgorithmus

Beispiel Kanonische Überdeckung / Synthesealgorithmus

Nr (N)	Verlag (V)	Magazin (M)	Ausgabe (A)	Gründung (G)	Seite (S)	Thema (T)
aa11	Heise	c't	11/18	1949	15	C++
aa11	Heise	c't	11/18	1949	24	C#
bb22	Heise	c't	12/18	1949	12	Python

Relation

R:={AGMNSTV}

FDs

N->AGMV
V->G
M->V
NS->T

Kandidatenschlüssel

{NS}

Normalformen

1NF ✔ 2NF ⚡ 3NF ⚡ BCNF ⚡ 4NF ⚡

1 Linksreduktion

N->AGMV
V->G
M->V
NS->T

2 Rechtsreduktion

N->AM
V->G
M->V
NS->T

3 $\alpha \rightarrow \emptyset$ entfernen

N->AM
V->G
M->V
NS->T

4 FDs zusammenfassen

N->AM
V->G
M->V
NS->T

Verlag (V)	Gründung (G)
Heise	1949
Computec	1989
Webedia	2007

…

{[N,A,M]},{[M,V]},{[N,S,T]}

Kanonische Überdeckung / Hauptteil Synthesealgorithmus

Vorschau

Synthesealgorithmus

- Gegeben sind ein Schema R und einer Menge FD funktionaler Abhängigkeiten.

1. Bestimme Kanonische Überdeckung FD^C zu FD, d.h.
   - Links- und Rechtsreduktion, Entfernung, Zusammenfassung
2. Für alle $\alpha \rightarrow \beta \in FD^C$ wird Schema $R_K = \alpha \cup \beta$, bzw. Relation $\{[\alpha, \beta]\}$, erzeugt. Die funktionalen Abhängigkeiten, die Attribute aus R_K enthalten, werden R_K zugeordnet.
3. *Ein* Schlüsselkandidat muss in den R_K aus Schritt 2 enthalten sein. Ansonsten wird eine Relation, die die Attribute eines Schlüsselkandidat enthält, zusätzlich erzeugt.
4. Einzelne R_K aus Schritt 2 können andere R_m enthalten. Diese Schemata R_m braucht man dann nicht mehr, denn alle Informationen sind in R_K.

Die funktionalen Abhängigkeiten, geeignet reduziert, bestimmen die Relationen im DBMS.

Vorschau

Normalformen

- Die erste Normalform ist gegeben, wenn es keine zusammengesetzten oder mehrwertigen Attribute gibt.
- Die zweite Normalform ist erfüllt, wenn die erste erfüllt ist und alle nicht-prim Attribute voll funktional abhängig von jedem Schlüsselkandidaten sind.
- Die dritte Normalform liegt vor, wenn die zweite erfüllt ist und jedes nicht-prim Attribut direkt und nicht transitiv von einem Schlüsselkandidaten abhängt.

Anmerkungen

- Beispiele und Anomalien, die verhindert werden, folgen!
- Für die zweite und dritte Normalform benötigen wir die Schlüsselkandidaten.
- Bedingungen beziehen sich auf nicht-prim Attribute. Redundanzen bei prim Attributen sind weiter möglich…