Teil VI

Relationale Theorie

Relationale Theorie

Lernziele für heute ...

- Vertiefte Kenntnisse der theoretischen Grundlagen des relationalen Entwurfs
- Korrektheit der Normalisierung
- Details des Syntheseverfahrens

Formalisierung

Wiederholung: Formalisierung Relationenmodell

- **Attribute und Domänen**
  - $\mathcal{U}$ nichtleere, endliche Menge: **Universum**
  - $A \in \mathcal{U}$: **Attribut**
  - $\mathcal{D} = \{D_1, \ldots, D_m\}$ Menge endlicher, nichtleerer Mengen: jedes D_i: **Wertebereich** oder **Domäne**
  - total definierte Funktion $\mathrm{dom} : \mathcal{U} \longrightarrow \mathcal{D}$
  - $\mathrm{dom}(A)$: **Domäne von** A
    $w \in \mathrm{dom}(A)$: **Attributwert** für A

Wiederholung: Formalisierung Relationenmodell /2

- **Relationenschemata und Relationen**
  - $R \subseteq \mathcal{U}$: **Relationenschema**
  - **Relation** r über $R = \{A_1, \ldots, A_n\}$ (kurz: $r(R)$) ist endliche Menge von Abbildungen $t : R \longrightarrow \bigcup_{i=1}^{m} D_i$, **Tupel** genannt
  - Es gilt $t(A) \in \mathrm{dom}(A)$ ($t(A)$ Restriktion von t auf $A \in R$)
  - für $X \subseteq R$ analog $t(X)$ **X-Wert** von t
  - Menge aller Relationen über R: **REL**$(R) := \{r \mid r(R)\}$

Wiederholung: Formalisierung Relationenmodell (3)

- **Datenbankschema und Datenbank**
  - Menge von Relationenschemata $S := \{R_1, \ldots, R_p\}$: **Datenbankschema**
  - **Datenbank** über S: Menge von Relationen $d := \{r_1, \ldots, r_p\}$, wobei $r_i(R_i)$
  - Datenbank d über S: $d(S)$
  - Relation $r \in d$: **Basisrelation**

Rechnen mit FDs

Wiederholung: Ableitung von FDs

r

A	B	C
a_1	b_1	c_1
a_2	b_1	c_1
a_3	b_2	c_1
a_4	b_1	c_1

- genügt $A \rightarrow B$ und $B \rightarrow C$
- dann gilt auch $A \rightarrow C$
- nicht ableitbar $C \rightarrow A$ oder $C \rightarrow B$

Formale Ableitung von FDs

- Gilt für f über R **SAT**$_R(F) \subseteq$ **SAT**$_R(f)$, dann **impliziert** F die FD f (kurz: $F \models f$)
- obiges Beispiel:

$$F = \{A \rightarrow B, B \rightarrow C\} \models A \rightarrow C$$

- Hüllenbildung: Ermittlung **aller** funktionalen Abhängigkeiten, die aus einer gegebenen FD-Menge abgeleitet werden können
- **Hülle** $F_R^+ := \{f \mid (f \text{ FD über } R) \land F \models f\}$
- Beispiel:

$$\begin{aligned}\{A \rightarrow B, B \rightarrow C\}^+ \quad = \quad & \{A \rightarrow B, B \rightarrow C, A \rightarrow C, AB \rightarrow C, A \rightarrow BC, \ldots, \\ & AB \rightarrow AB, \ldots\}\end{aligned}$$

Ableitungsregeln

F1	Reflexivität	$X \supseteq Y$	$\Longrightarrow$	$X \rightarrow Y$
F2	Augmentation	$\{X \rightarrow Y\}$	$\Longrightarrow$	$XZ \rightarrow YZ$ sowie $XZ \rightarrow Y$
F3	Transitivität	$\{X \rightarrow Y, Y \rightarrow Z\}$	$\Longrightarrow$	$X \rightarrow Z$
F4	Dekomposition	$\{X \rightarrow YZ\}$	$\Longrightarrow$	$X \rightarrow Y$
F5	Vereinigung	$\{X \rightarrow Y, X \rightarrow Z\}$	$\Longrightarrow$	$X \rightarrow YZ$
F6	Pseudotransitivität	$\{X \rightarrow Y, WY \rightarrow Z\}$	$\Longrightarrow$	$WX \rightarrow Z$

F1-F3 bekannt als **Armstrong-Axiome** (sound, complete)

- **gültig** (sound): Regeln leiten keine FDs ab, die logisch nicht impliziert
- **vollständig** (complete): alle implizierten FDs werden abgeleitet
- **unabhängig** (independent) oder auch bzgl. $\subseteq$ minimal: keine Regel kann weggelassen werden

Beweis: $\mathbf{F1}$

- Annahme: $X \supseteq Y$, $X, Y \subset R$, $t_1, t_2 \in r(R)$ mit $t_1(X) = t_2(X)$
- dann folgt: $t_1(Y) = t_2(Y)$ wegen $X \supseteq Y$
- daraus folgt: $X \rightarrow Y$

Beweis: $\mathbf{F2}$

- Annahme: $X \rightarrow Y$ gilt in $r(R)$, jedoch nicht: $XZ \rightarrow YZ$
- dann müssen zwei Tupel $t_1, t_2 \in r(R)$ existieren, so dass gilt
  1. $t_1(X) = t_2(X)$
  2. $t_1(Y) = t_2(Y)$
  3. $t_1(XZ) = t_2(XZ)$
  4. $t_1(YZ) \neq t_2(YZ)$
- Widerspruch wegen $t_1(Z) = t_2(Z)$ aus (1) und (3), woraus folgt: $t_1(YZ) = t_2(YZ)$ (in Verbindung mit (4))

Beweis: **F3**

- Annahme: in $r(R)$ gelten:

  (1) $X \rightarrow Y$

  (2) $Y \rightarrow Z$

- demzufolge für zwei beliebige Tupel $t_1, t_2 \in r(R)$ mit $t_1(X) = t_2(X)$ muss gelten:

  (3) $t_1(Y) = t_2(Y)$ (wegen (1))

  (4) $t_1(Z) = t_2(Z)$ (wegen (3) und (2))

- daher gilt: $X \rightarrow Z$

Alternative Regelmenge

- B-Axiome oder **RAP-Regeln**

$$
\begin{array}{lll}
\textbf{R} \text{ Reflexivität} & \{\} \implies X \rightarrow X \\
\textbf{A} \text{ Akkumulation} & \{X \rightarrow YZ, Z \rightarrow AW\} \implies X \rightarrow YZA \\
\textbf{P} \text{ Projektivität} & \{X \rightarrow YZ\} \implies X \rightarrow Y
\end{array}
$$

- Regelmenge ist vollständig, da Armstrong-Axiome daraus abgeleitet werden können

Membership-Problem

Membership-Problem

Kann eine bestimmte FD $X \rightarrow Y$ aus der vorgegebenen Menge F abgeleitet werden, d.h. wird sie von F impliziert?

Membership-Problem: „$X \rightarrow Y \in F^+$?“

- **Hülle einer Attributmenge** X bzgl. F ist
  $X_F^+ := \{A \mid X \rightarrow A \in F^+\}$
- Membership-Problem kann durch das modifizierte Problem

  Membership-Problem (2): „$Y \subseteq X_F^+$?“

  in linearer Zeit gelöst werden

Algorithmus **Closure**: Ermittlung der Hülle X_F^+ von X bzgl. F

$\textbf{Closure}(F, X)$:
$\quad X^+ := X$
$\quad$ `repeat`
$\quad\quad \overline{X}^+ := X^+$ /* **R**-*Regel* */
$\quad\quad$ `forall` FDs $Y \rightarrow Z \in F$
$\quad\quad\quad$ `if` $Y \subseteq X^+$ `then` $X^+ := X^+ \cup Z$ /* **A**-*Regel* */
$\quad$ `until` $X^+ = \overline{X}^+$
$\quad$ `return` X^+

$\textbf{Member}(F, X \rightarrow Y)$: /* *Test auf* $X \rightarrow Y \in F^+$ */
$\quad$ `return` $Y \subseteq \textbf{Closure}(F, X)$ /* **P**-*Regel* */

Algorithmus **Closure**: Beispiel

$$A \rightarrow C \in \{\underbrace{A \rightarrow B}_{f_1}, \underbrace{B \rightarrow C}_{f_2}\}^+?$$

- $\mathbf{Member}(\{f_1, f_2\}, A \rightarrow C)$
- $C \subseteq \mathbf{Closure}(\{f_1, f_2\}, A)$
- X^+ ist initial $\{A\}$, schrittweises Hinzunehmen von B und C

Überdeckungen

- F heißt **äquivalent** zu G
- oder: F **Überdeckung** von G; kurz: $F \equiv G$ falls $F^+ = G^+$
- d.h.:

$$\forall g \in G : g \in F^+ \;\wedge\; \forall f \in F : f \in G^+$$

- wichtige Entwurfsaufgabe: Finden einer Überdeckung, die
  - einerseits so wenig Attribute wie möglich in ihren funktionalen Abhängigkeiten und
  - andererseits möglichst wenig funktionale Abhängigkeiten insgesamt enthält
- verschiedene Formen von Überdeckung: nicht-redundant, reduziert, minimal, ringförmig

Reduktionsoperationen

- Ziel: Entfernen überflüssiger Attribute auf linker bzw. rechter Seite von FDs
- **Linksreduktion**: entfernt unwesentliche Attribute auf der linken Seite einer FD
- **Rechtsreduktion**: entsprechend auf der rechten Seite
- erw. Relationenschema $\mathcal{R} = (R, \mathcal{K})$, FD-Menge F über R, A ist ein Attribut aus R und $X \rightarrow Y$ eine FD aus F

Unwesentliche Attribute

A heißt **unwesentlich** in $X \rightarrow Y$ bzgl. F, wenn

- $X = AZ, Z \neq X \quad \Longrightarrow \quad (F - \{X \rightarrow Y\}) \cup \{Z \rightarrow Y\} \equiv F$ oder
- $Y = AW, W \neq Y \quad \Longrightarrow \quad (F - \{X \rightarrow Y\}) \cup \{X \rightarrow W\} \equiv F$

Reduktionsoperationen /2

- A kann also aus der FD $X \rightarrow Y$ entfernt werden, ohne dass sich die Hülle von F ändert
- FD $X \rightarrow Y$ heißt **linksreduziert**, wenn kein Attribut in X unwesentlich ist.
- FD $X \rightarrow Y$ heißt **rechtsreduziert**, wenn kein Attribut in Y unwesentlich ist.

Minimale Überdeckung

- Eine **minimale Überdeckung** ist eine Überdeckung, die eine minimale Anzahl von FDs enthält
- Auswahl der **kleinsten** aller nicht-redundanten Überdeckungen
- FD-Menge F heißt **minimal** gdw.

$$\forall F' \, [F' \equiv F \Rightarrow |F| \leq |F'|]$$

- Bestimmung etwa durch reduzierte Überdeckung mit anschließender Äquivalenzklassenbildung (später)

Reduzierte Überdeckung

ReducedCover(F):
  `forall` FD $X \rightarrow Y \in F$ */* Linksreduktion */*
    `forall` $A \in X$ */* A unwesentlich ? */*
      `if` $Y \subseteq$ **Closure**$(F, X - \{A\})$
      `then` ersetze $X \rightarrow Y$ durch $(X - A) \rightarrow Y$ in F
  `forall` verbleibende FD $X \rightarrow Y \in F$ */* Rechtsreduktion */*
    `forall` $B \in Y$ */* B unwesentlich ? */*
      `if` $B \subseteq$ **Closure**$(F - \{X \rightarrow Y\} \cup \{X \rightarrow (Y - B)\}, X)$
      `then` ersetze $X \rightarrow Y$ durch $X \rightarrow (Y - B)$

  Eliminiere FDs der Form $X \rightarrow \emptyset$
  Vereinige FDs der Form $X \rightarrow Y_1, X \rightarrow Y_2, \ldots$ zu $X \rightarrow Y_1 Y_2 \ldots$
  `return` resultierende FDs

Reduzierte Überdeckung: Beispiel

- Geg.: FD-Menge

$$F = \{f_1 : A \rightarrow B, f_2 : AB \rightarrow C, f_3 : A \rightarrow C, f_4 : B \rightarrow A, f_5 : C \rightarrow E\}$$

1. Linksreduktion: bei FD f_2 Attribut A streichen, da $C \subseteq \textbf{Closure}(F, \{A\})$ (wegen f_3)
2. Rechtsreduktion: FD f_3 durch $A \rightarrow \{\}$ ersetzt, da $C \subseteq \textbf{Closure}(\{A \rightarrow B, B \rightarrow C, A \rightarrow \{\}, B \rightarrow A, C \rightarrow E\}, \{A\})$
3. Streichen von $A \rightarrow \{\}$

- Ergebnis:

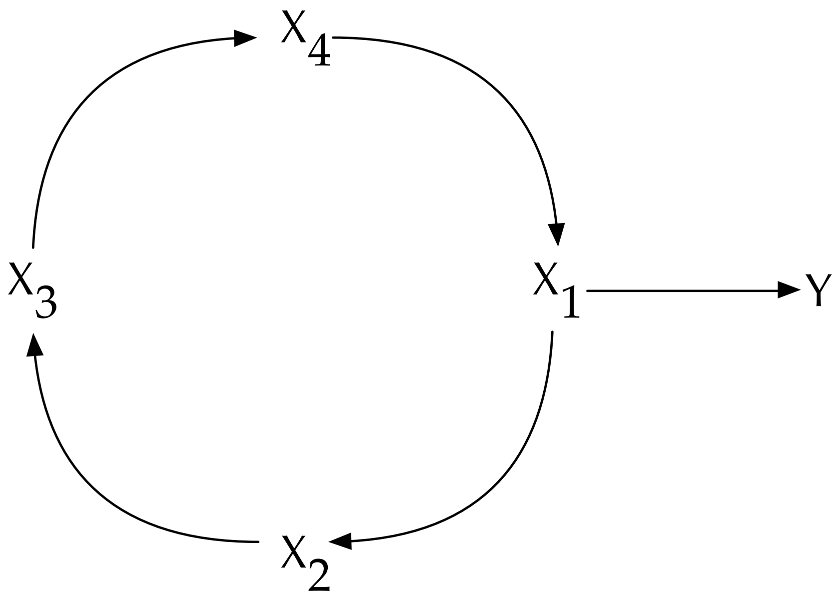
$$\textbf{ReducedCover}(F) = \{A \rightarrow B, B \rightarrow C, B \rightarrow A, C \rightarrow E\}$$

Äquivalenzklassen

- FDs mit äquivalenten linken Seiten werden zu einer Äquivalenzklasse zusammengefasst
- FDs $X_1 \rightarrow Y_1$ und $X_2 \rightarrow Y_2$ liegen in einer **Äquivalenzklasse**, wenn $X_1 \rightarrow X_2$ und $X_2 \rightarrow X_1$ gelten
- In einigen Fällen können nun zwei solche FDs in einer Äquivalenzklasse zu einer FD $X \rightarrow Y_1 Y_2$ zusammengefasst werden
- Da die FDs einer Äquivalenzklasse in die Form $X_1 \rightarrow X_2, X_2 \rightarrow X_3, \ldots, X_n \rightarrow X_1, X_1 \rightarrow Y$ überführt werden können, nennt man eine Überdeckung dieser Form eine **ringförmige Überdeckung**

Äquivalenzklassen /2

- linke Seiten sind äquivalent, wenn sie sich gegenseitig funktional bestimmen
- Relationenschema R mit $X_i, Y \subset R$, FD-Menge $X_i \rightarrow X_j$ und $X_i \rightarrow Y$ mit $1 \leq i, j \leq n$ kann dargestellt werden durch $(X_1, X_2, \ldots, X_n) \rightarrow Y$



Mehr zu Normalformen

Wiederholung: Integritätsbedingungen

- Identifizierende Attributmenge $K := \{B_1, \ldots, B_k\} \subseteq R$:

$$\forall t_1, t_2 \in r \; [t_1 \neq t_2 \quad \Longrightarrow \quad \exists B \in K : t_1(B) \neq t_2(B)]$$

- **Schlüssel**: ist minimale identifizierende Attributmenge
  - {`Name, Jahrgang, Weingut`} und
  - {`WeinID`} für `WEINE`
- **Primattribut**: Element eines Schlüssels
- **Primärschlüssel**: ausgezeichneter Schlüssel
- **Oberschlüssel** oder **Superkey**: jede Obermenge eines Schlüssels (= identifizierende Attributmenge)

Schüssel und Hüllenbildung

- Identifizierende Attributmenge $K := \{B_1, \ldots, B_k\} \subseteq R$:

$$\forall t_1, t_2 \in r\ [t_1 \neq t_2 \quad \Longrightarrow \quad \exists B \in K : t_1(B) \neq t_2(B)]$$

$$\Longrightarrow \quad K \rightarrow R$$

- **Hülle einer Attributmenge** X bzgl. F ist
$X_F^+ := \{A \mid X \rightarrow A \in F^+\} \quad \Longrightarrow$
  - $K_F^+ = R$
    $\Longrightarrow \quad K =$ Identifizierende Attributmenge
  - $K_F^+ = R \ \wedge \ X\ minimal$ (linksreduziert)
    $\Longrightarrow \quad K =$ Schlüsselkandidat

- Attribute, die nur funktional-abhängig sind, sind **nicht Teil** eines Schlüssels

- Attribute, die nie funktional-abhängig sind (nur bestimmend), sind **immer** Teil eines Schlüssels

Wiederholung: Zweite Normalform

- **partielle Abhängigkeit** liegt vor, wenn ein Attribut funktional schon von einem **Teil** des Schlüssels abhängt

Name	Weingut	Farbe	Anbaugebiet	Region	Preis
La Rose ...	Ch. La Rose	Rot	Saint-Emilion	Bordeaux	39.00
Creek Shiraz	Creek	Rot	Barossa Valley	Südaustralien	7.99
Pinot Noir	Creek	Rot	Barossa Valley	Südaustralien	10.99
Zinfandel	Helena	Rot	Napa Valley	Kalifornien	5.99
Pinot Noir	Helena	Rot	Napa Valley	Kalifornien	19.99
Riesling Reserve	Müller	Weiß	Rheingau	Hessen	14.99
Chardonnay	Bighorn	Weiß	Napa Valley	Kalifornien	9.90

$$
\begin{aligned}
f_1&: \texttt{Name, Weingut} \rightarrow \texttt{Preis}\\
f_2&: \texttt{Name} \rightarrow \texttt{Farbe}\\
f_3&: \texttt{Weingut} \rightarrow \texttt{Anbaugebiet, Region}\\
f_4&: \texttt{Anbaugebiet} \rightarrow \texttt{Region}
\end{aligned}
$$

Formale Definition der zweiten Normalform

- Hinweis: partiell abhängiges Attribut stören nur, wenn es **kein** Primattribut ist
- 2NF formal: erweitertes Relationenschema $\mathcal{R} = (R, \mathcal{K})$, FD-Menge F über R

Zweite Normalform

- Y **hängt partiell** von X bzgl. F ab, wenn die FD $X \rightarrow Y$ nicht linksreduziert ist
- Y **hängt voll** von X ab, wenn die FD $X \rightarrow Y$ linksreduziert ist
- $\mathcal{R}$ ist in **2NF**, wenn $\mathcal{R}$ in 1NF ist und jedes Nicht-Primattribut von R voll von jedem Schlüssel von $\mathcal{R}$ abhängt

Entwurfsverfahren

Entwurfsverfahren: Ziele

- Universum $\mathcal{U}$ und FD-Menge F gegeben
- lokal erweitertes Datenbankschema $S = \{(R_1, \mathcal{K}_1), \ldots, (R_p, \mathcal{K}_p)\}$ berechnen mit
  - $\mathbf{T1}$: S charakterisiert vollständig F
  - $\mathbf{S1}$: S ist in 3NF bezüglich F
  - $\mathbf{T2}$: Dekomposition von $\mathcal{U}$ in $R_1, \ldots, R_p$ ist verbundtreu bezüglich F
  - $\mathbf{S2}$: Minimalität, d.h. $\not\exists S' : S'$ erfüllt $\mathbf{T1}$, $\mathbf{S1}$, $\mathbf{T2}$ und $|S'| < |S|$

Entwurfsverfahren: Beispiel

- Datenbankschemata schlecht entworfen, wenn nur eins dieser vier Kriterien nicht erfüllt
- Beispiel: $S = \{(AB, \{A\}), (BC, \{B\}), (AC, \{A\})\}$ erfüllt $\mathbf{T1}$, $\mathbf{S1}$ und $\mathbf{T2}$ bezüglich $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$

  in dritter Relation AC-Tupel redundant oder inkonsistent
- korrekt: $S' = \{(AB, \{A\}), (BC, \{B\})\}$

Dekomposition

- Geg.: initiales Universalrelationenschema $\mathcal{R} = (\mathcal{U}, \mathcal{K}(F))$ mit allen Attributen und einer von erfassten FDs F über R implizierten Schlüsselmenge
  - Attributmenge $\mathcal{U}$ und eine FD-Menge F
  - suche alle $K \rightarrow \mathcal{U}$ mit K minimal, für die $K \rightarrow \mathcal{U} \in F^+$ gilt ($\mathcal{K}(F)$)
- Ges.: Zerlegung in $D = \{\mathcal{R}_1, \mathcal{R}_2, \dots\}$ von 3NF-Relationenschemata

Dekomposition: Algorithmus

Decompose$(\mathcal{R})$:

Setze $D := \{\mathcal{R}\}$

while $\mathcal{R}' \in D$, das 3NF nicht erfüllt

/ Finde Attribut A, das transitiv von K abhängig ist */*

if Schlüssel K mit $K \rightarrow Y, Y \not\rightarrow K, Y \rightarrow A, A \notin KY$ **then**

/ Zerlege Relationenschema R bzgl. A */*

$R_1 := R - A$, $R_2 := YA$

$\mathcal{R}_1 := (R_1, \mathcal{K})$, $\mathcal{R}_2 := (R_2, \mathcal{K}_2 = \{Y\})$

$D := (D - \mathcal{R}') \cup \{\mathcal{R}_1\} \cup \{\mathcal{R}_2\}$

end if

end while

return D

Dekomposition: Beispiel

- initiales Relationenschema $R = ABC$
- funktionale Abhängigkeiten $F = \{A \rightarrow B, B \rightarrow C\}$
- Schlüssel $K = A$

Dekomposition: Beispiel /2

- initiales Relationenschema R mit Name, Weingut, Preis, Farbe, Anbaugebiet, Region
- funktionale Abhängigkeiten

$$
\begin{array}{lll}
f_1: & \texttt{Name, Weingut} & \rightarrow \texttt{Preis} \\
f_2: & \texttt{Name, Weingut} & \rightarrow \texttt{Weingut} \\
f_3: & \texttt{Name, Weingut} & \rightarrow \texttt{Name} \\
f_4: & \texttt{Name} & \rightarrow \texttt{Farbe} \\
f_5: & \texttt{Weingut} & \rightarrow \texttt{Anbaugebiet, Region} \\
f_6: & \texttt{Anbaugebiet} & \rightarrow \texttt{Region}
\end{array}
$$

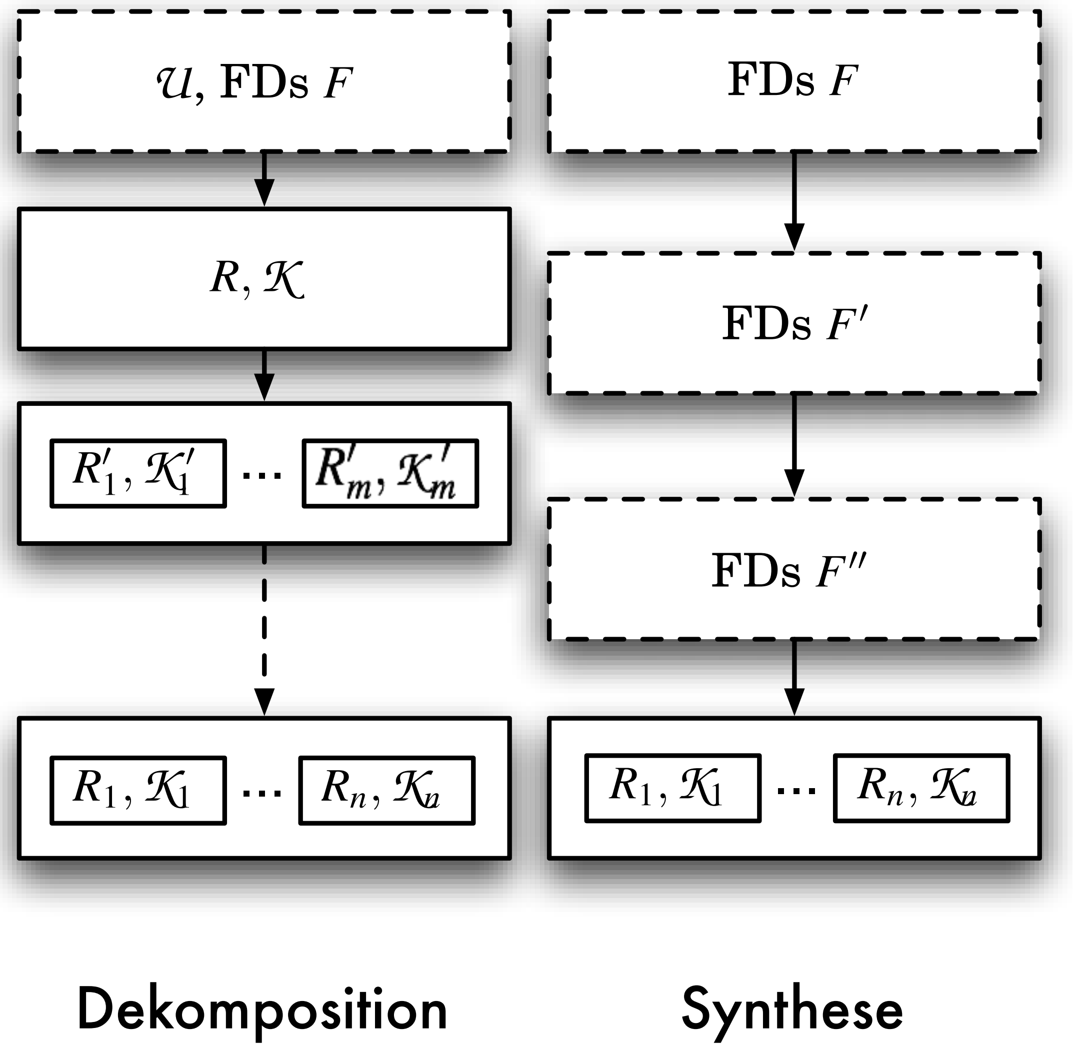
Dekomposition: Bewertung

- Vorteile: 3NF, Verbundtreue
- Nachteile: restliche Kriterien nicht, reihenfolgeabhängig, NP-vollständig (Schlüsselsuche)

Details zum Syntheseverfahren

- Prinzip: Synthese formt Original-FD-Menge F in resultierende Menge von Schlüsselabhängigkeiten G so um, dass $F \equiv G$ gilt
- „Abhängigkeitstreue“ im Verfahren verankert
- 3NF und Minimalität wird auch erreicht, reihenfolgeunabhängig
- Zeitkomplexität: quadratisch

Vergleich Dekomposition — Synthese



Syntheseverfahren für Relationenschema R mit FDs F

Ges.: verlustfreie und abhängigkeitstreue Zerlegung in $R_1, \ldots R_n$, wobei alle R_i in 3NF sind

Synthesize(F):
  $\hat{F} :=$ **MinimalCover**(F) */* Bestimme minimale Überdeckung */*
  Bilde Äquivalenzklassen C_i von FDs aus $\hat{F}$ mit gleichen oder
    äquivalenten linken Seiten, d.h. $C_i = \{X_i \rightarrow A_{i1}, X_i \rightarrow A_{i2}, \ldots\}$
  Bilde zu jeder Äquivalenzklasse C_i ein Schema der Form
    $R_{C_i} = \{X_i \cup \{A_{i1}\} \cup \{A_{i2}\} \cup \ldots\}$
  if keines der Schemata R_{C_i} enthält einen Schlüssel von R
  then erzeuge weiteres Relationenschema R_K mit Attributen
    aus R, die Schlüssel bilden
  return $\{R_K, R_{C_1}, R_{C_2}, \ldots\}$

Synthese Beispiel

- FD-Menge

$$F = \{A \rightarrow B, AB \rightarrow C, A \rightarrow C, B \rightarrow A, C \rightarrow E\}$$

- minimale Überdeckung

$$\hat{F} = \{A \rightarrow B, B \rightarrow C, B \rightarrow A, C \rightarrow E\}$$

- Zusammenfassung zu Äquivalenzklassen

$$\begin{aligned} C_1 &= \{A \rightarrow B, B \rightarrow C, B \rightarrow A\} \\ C_2 &= \{C \rightarrow E\} \end{aligned}$$

- Syntheseergebnis

$$(ABC, \{\{A\}, \{B\}\}), (CE, \{C\})$$

Erreichung der Verbundtreue

- Erreichen der Verbundtreue durch einfachen „Trick“:
  - Erweitern der Original-FD-Menge F um $\mathcal{U} \rightarrow \delta$ um Dummy-Attribut δ
  - δ wird nach Synthese entfernt
- Beispiel: $\{A \rightarrow B, C \rightarrow E\}$
  - Syntheseergebnis $(AB, \{A\}), (CE, \{C\})$ ist nicht verbundtreu, da Universalschlüssel in keinem Schema enthalten ist
  - Dummy-FD $ABCE \rightarrow \delta$; reduziert auf $AC \rightarrow \delta$
  - liefert drittes Relationenschema

$$(AC, \{AC\})$$

Synthese: Beispiel

- FD-Menge

$$
\begin{aligned}
f_1 &= \{\texttt{Name}, \texttt{Weingut} \rightarrow \texttt{Preis}\} \\
f_2 &= \{\texttt{Name}, \texttt{Weingut} \rightarrow \texttt{Weingut}\} \\
f_3 &= \{\texttt{Name}, \texttt{Weingut} \rightarrow \texttt{Name}\} \\
f_4 &= \{\texttt{Name} \rightarrow \texttt{Farbe}\} \\
f_5 &= \{\texttt{Weingut} \rightarrow \texttt{Anbaugebiet}, \texttt{Region}\} \\
f_6 &= \{\texttt{Anbaugebiet} \rightarrow \texttt{Region}\}
\end{aligned}
$$

Synthese: Beispiel /2

- Ablauf

1. minimale Überdeckung: Entfernen von f_2, f_3 sowie `Region` in f_5
2. Äquivalenzklassen:

$$\begin{aligned}
C_1 &= \{\texttt{Name}, \texttt{Weingut} \rightarrow \texttt{Preis}\} \\
C_2 &= \{\texttt{Name} \rightarrow \texttt{Farbe}\} \\
C_3 &= \{\texttt{Weingut} \rightarrow \texttt{Anbaugebiet}\} \\
C_4 &= \{\texttt{Anbaugebiet} \rightarrow \texttt{Region}\}
\end{aligned}$$

3. Ableitung der Relationenschemata

Zusammenfassung

- Formalisierung des Relationenmodells und der funktionalen Abhängigkeiten
- Algorithmen zur Normalisierung

Kontrollfragen

- Was muß beim Syntheseverfahren beachtet werden, um Spezialfälle wie zyklische Abhängigkeiten oder fehlende Schlüssel zu berücksichtigen?