# Kapitel 4
# Anfragen unter Abhängigkeiten

## Konjunktive Anfragen

- vorhergehende Kapitel
- Bsp: Datalog
- Auswertung von Joins verschiedener Relationen
- Minimierung von Anfragen durch Rewriting

## Weitere Optimierungsmöglichkeit

- Abhängigkeiten innerhalb einer Relation können ebenfalls ausgenutzt werden

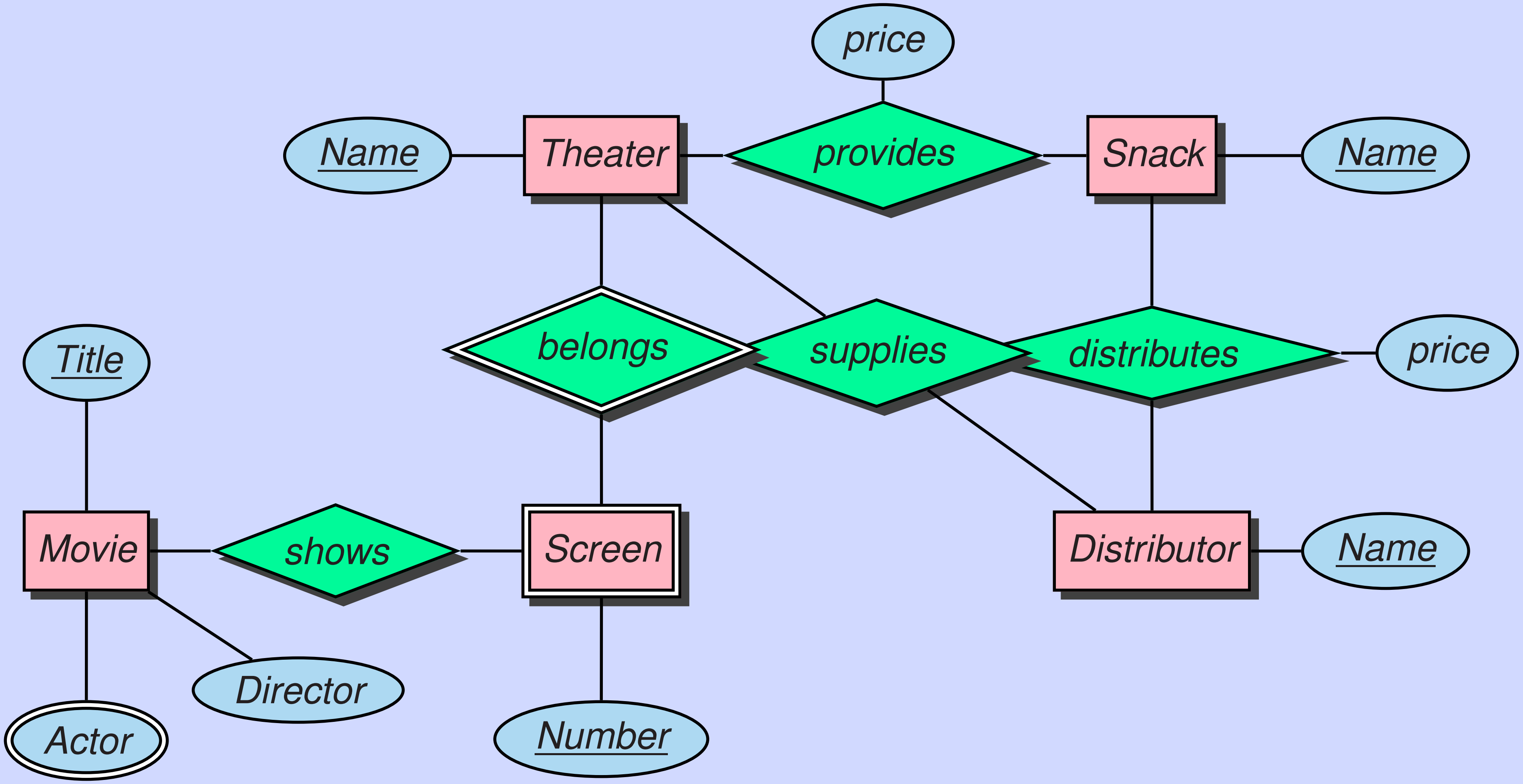
## Literatur

- Abiteboul, Hull, Vianu, *Foundations of Databases*, Chapter 8
- Maier: Theory of Relational Databases, Chapters 4-8

# 4.1 Daten-Abhängigkeiten

- spezielle *semantische* Integritätsbedingungen
- schränken die Menge der möglichen Instanzen zu einem Datenbank-Schema ein
  - funktionale Abhängigkeiten (functional dependencies)
  - mehrwertige Abhängigkeiten (multivalued dependencies)
  - Verbundabhängigkeiten (Join-Dependencies)
  - Inklusionsabhängigkeiten (Inclusion Dependencies)

**Beispiel 4.1 (ER-Schema einer Movie-Datenbank)**



## Anmerkungen

- Screen ist ein schwacher Entitätstyp
  Schlüssel: (Screen.Number, Theater.Name)
  Damit ergibt sich das Schema von shows zu (Theater, Screen, Title)
- Movie.Actor ist multivalued

## Geltende Integritätsbedingungen

- In jedem Saal eines Kinos wird nur ein Film gezeigt.
- Das Snack-Angebot ist in allen Sälen eines Kinos dasselbe,
- Ein Distributor bietet seine Snacks überall zum gleichen Preis (oder garnicht) an,
- Die Snacks werden in den Kinos zu dem vom Distributor angegebenen Preis verkauft,

**Beispiel 4.2**

*... und so ist die Movie-Datenbank gespeichert:*

| Movies | Title | Director | Actor |
|---|---|---|---|
| | The Birds | Hitchcock | Hedren |
| | The Birds | Hitchcock | Taylor |
| | Bladerunner | Scott | Hannah |

| Offerings | Snack | Distributor | Price | Theater |
|---|---|---|---|---|
| | coffee | Smart | 2.35 | Rex |
| | coffee | Smart | 2.35 | Le Champo |
| | coffee | Smart | 2.35 | Cinoche |
| | coffee | Leclerc | 2.60 | Cinoche |
| | wine | Smart | 0.80 | Rex |
| | wine | Smart | 0.80 | Cinoche |
| | popcorn | Leclerc | 5.60 | Cinoche |
| | popcorn | Leclerc | 5.60 | Rex |
| | ⋮ | ⋮ | ⋮ | ⋮ |

| Showings | Theater | Screen | Title | Snack | Price |
|---|---|---|---|---|---|
| | *Rex* | *1* | *The Birds* | *coffee* | *2.35* |
| | *Rex* | *1* | *The Birds* | *popcorn* | *5.60* |
| | *Rex* | *2* | *Bladerunner* | *coffee* | *2.35* |
| | *Rex* | *2* | *Bladerunner* | *popcorn* | *5.60* |
| | *Le Champo* | *1* | *The Birds* | *tea* | *1.30* |
| | *Le Champo* | *1* | *The Birds* | *popcorn* | *5.60* |
| | *Cinoche* | *1* | *The Birds* | *Coke* | *1.50* |
| | *Cinoche* | *1* | *The Birds* | *Wine* | *1.80* |
| | *Cinoche* | *2* | *Bladerunner* | *Coke* | *1.50* |
| | *Cinoche* | *2* | *Bladerunner* | *Wine* | *1.80* |
| | ⋮ | ⋮ | ⋮ | ⋮ | |

□

## Begriffe und Notation

- $\mathbf{R} = \{R_1, \ldots, R_n\}$ bezeichnet ein Datenbankschema
- $R_i = (A_{1i}, \ldots, A_{1k_i})$, bezeichnet ein Relationsschema,
  oft auch als $R(A, B, C)$ oder $R(U)$.
  $X \subseteq U$ bezeichnet entsprechend eine Teilmenge der Attribute.
- $\mathcal{I}(\mathbf{R})$ bezeichnet eine Instanz zu einem Datenbankschema,
- $r_i = \mathcal{I}(R_i)$ bezeichnet die zu $R_i$ gehörende Relation innerhalb dieser Instanz,
  oft auch als $r = \mathcal{I}(R)$.
  $s, t \in r$ bezeichnet Tupel in $r$.

# ARTEN VON ABHÄNGIGKEITEN

## Funktionale Abhängigkeiten

Sei $R(U)$ ein Relationsschema, $r = \mathcal{I}(R)$ und $X, Y \subseteq U$.
$r$ erfüllt die **funktionale Abhängigkeit (FA)** $X \rightarrow Y$ , gdw. für alle $t, s \in r$ gilt:

$$t[X] = s[X] \Rightarrow t[Y] = s[Y] \qquad \text{(geschrieben: } r \models X \rightarrow Y\text{)}$$

Im Beispiel (aus ER-Diagramm und zusätzlichen Integritätsbedingungen):

- Movies $\models$ Title $\rightarrow$ Director
- Showings $\models$ {Theater, Screen} $\rightarrow$ Title
- Offerings $\models$ {Snack, Distributor} $\rightarrow$ Price

## Sonderfall: Schlüsselabhängigkeiten

Eine FA von der Form $X \rightarrow R$ ist eine **Schlüsselabhängigkeit (key dependency)**;
$X$ ist ein Schlüsselkandidat von $R$.

## Verbundabhängigkeiten

(Darstellung einer Relation als Verbund einer Zerlegung)

Sei $R(U)$ ein Relationsschema, $r = \mathcal{I}(R)$ und $X_1, \ldots, X_n$, mit $X_i \subseteq U$ eine Zerlegung von $R$. $r$ erfüllt die **Verbundabhängigkeit (VA)** $\bowtie [X_1, \ldots, X_n]$ gdw. gilt:

$$r = \bowtie_{i=1}^{n} \pi[X_i](r).$$

- d.h., definiert man die einzelnen $\pi[X_i](r)$ als Relationen $r_i$, so ist $r$ deren Join.

Im Beispiel:

- Movies $\models \bowtie$ [{Title, Director}, {Title, Actor}]
- Showings $\models \bowtie$ [{Theater, Screen, Title}, {Theater, Snack, Price}]
  also
  $$\text{Showings} = \text{shows} \bowtie \text{provides}$$
- Offerings $\models \bowtie$ [{Snack, Distr., Price}, {Theater, Snack, Price}, {Distr., Theater}],
  also
  $$\text{Offerings} = \text{distributes} \bowtie \text{provides} \bowtie \text{supplies}$$

## Anmerkung

Betrachten Sie

$$\text{Offerings} = \text{distributes} \bowtie \text{provided} \bowtie \text{supplies}$$

als View.

- Hypergraph von Offerings ist zyklisch: (SDP, TSP, DT)
- Join-Dependency gilt aufgrund der angenommenen Integritätsbedingungen

## Sonderfall: Mehrwertige Abhängigkeiten

Ist $n = 2$, und gilt

$$\mathcal{I} \models \bowtie [XY, X(U - Y)]$$

so spricht man von einer **mehrwertigen Abhängigkeit**, $X \twoheadrightarrow Y$

- Es gilt: Wenn $X \to Y$, dann auch $X \twoheadrightarrow (U - Y)$

Im Beispiel:

- Showings $\models$ Theater $\twoheadrightarrow$ $\{$Screen, Title$\}$ und
- Showings $\models$ Theater $\twoheadrightarrow$ $\{$Snack, Price$\}$

## Inklusionsabhängigkeiten

Seien $R_1(U_1)$ und $R_2(U_2)$ Relationsschemata, $r_1 = \mathcal{I}(R_1)$, $r_2 = \mathcal{I}(R_2)$ und $Y \subseteq U_1 \cap U_2$. $r_1, r_2$ erfüllen die **Inklusionsabhängigkeit (IA)** $R_1[Y] \subseteq R_2[Y]$ gdw. gilt:

$$\pi[Y](r_1) \subseteq \pi[Y](r_2).$$

Im Beispiel:

- Showings[Title] $\subseteq$ Movies[Title]
- Showings[Snack] $\subseteq$ Offerings[Snack]
- Offerings[Theater] $\subseteq$ Showings[Theater]

- Das Wissen über Inklusionsabhängigkeiten kann im übrigen auch die Berechnung von Joins vereinfachen, da dadurch dangling Tuples ausgeschlossen werden können:
  - Die Anwendung des Semijoins in vollständigen Reduzierern kann entfallen
  - In Anfragen mit zyklischen Hypergraphen kann der Zyklus aufgebrochen werden
- Wichtiges Beispiel: Referentielle Integrität/Fremdschlüssel

# WEITERES ZU FUNKTIONALEN ABHÄNGIGKEITEN

Sei $U$ eine Menge von Attributen, $\Sigma$ eine Menge funktionaler Abhängigkeiten über $U$.

## Abschlusseigenschaften

**Definition 4.1**

- $\Sigma$ **impliziert** *die funktionale Abhängigkeit* $X \to Y$ *gdw. für jede Relation* $r$ *aus* $r \models \Sigma$ *auch* $r \models X \to Y$ *folgt.*
  *(Notation:* $\Sigma \models X \to Y$*)*
- $\Sigma^+ = \{X \to Y \mid \Sigma \models X \to Y\}$ *ist die* **Hülle** *von* $\Sigma$. □

Berechnung von $\Sigma^+$: **Armstrong-Axiome**:

- Reflexivität: Aus $Y \subseteq X \subseteq U$ folgt (trivial) $X \to Y$,
- Augmentation: Aus $X \to Y$ und $Z \subseteq U$ folgt $XZ \to YZ$,
- Transitivität: Aus $X \to Y$ und $Y \to Z$ folgt $X \to Z$

Die Armstrong-Axiome sind **korrekt** und **vollständig**.

## Äquivalenz und Minimalität

Seien $\Sigma_1, \Sigma_2$ Mengen von funktionalen Abhängigkeiten.

**Definition 4.2**

- $\Sigma_1$ *und* $\Sigma_2$ *sind* **äquivalent** *gdw.* ${\Sigma_1}^+ = {\Sigma_2}^+$.
- $\Sigma$ *heißt* **minimal**, *gdw.*

  *(1)* $X \to Y \in \Sigma \Rightarrow Y$ *besteht aus genau einem Attribut.*

  *(2)* $X \to A \in \Sigma \Rightarrow \Sigma \setminus \{X \to A\}$ *nicht äquivalent zu* $\Sigma$.

  *(3)* $X \to A \in \Sigma, Z \subset X \Rightarrow \Sigma \setminus \{X \to A\} \cup \{Z \to A\}$ *nicht äquivalent zu* $\Sigma$. □

**Satz 4.1**

*Zu jeder Menge* $\Sigma$ *von funktionalen Abhängigkeiten existiert eine äquivalente minimale Menge von funktionalen Abhängigkeiten* $\Sigma^{min}$. □

**Bemerkung:**

- $\Sigma^{min}$ kann in polynomieller Zeit erzeugt werden.
- $\Sigma^{min}$ ist nicht notwendigerweise eindeutig.

## Erweiterter DB-Schema-Begriff

Häufig werden Integritätsbedingungen als Teil des Datenbankschemas gesehen:

- Sei $\mathbf{R}$ ein Schema, $\Sigma$ eine Menge von Abhängigkeiten in $\mathbf{R}$. Dann ist

$$Sat(\mathbf{R}, \Sigma)$$

  die Menge aller Instanzen $\mathcal{I}$, von $\mathbf{R}$, die jede Abhängigkeit $\sigma \in \Sigma$ erfüllen.

# Normalisierungstheorie und Normalformen

- Normalformen sind auf Basis von Abhängigkeiten definiert
- Zerlegung von Relationen in Teilrelationen
- Erstellung eines normalisierten Schemas: Verlustfreiheit

**Definition 4.3**

*Sei $R(U)$ eine Relation, $\Sigma$ eine Menge von Abhängigkeiten über $U$ und $\rho = \{X_1, \ldots, X_n\}$, $X_i \subseteq U$ eine* **Zerlegung** *von $R$.*

*$\rho$ heißt* **verlustfrei**, *gdw. für jede Relation $r \in \mathcal{I}(R)$, die $\Sigma$ erfüllt, gilt:*

$$r = \pi[X_1](r) \bowtie \ldots \bowtie \pi[X_n](r).$$

□

- Test auf Verlustfreiheit
  – bezüglich einer gegebenen Menge $\Sigma$ von FAs und VAs

## Normalformen

- 3NF und BCNF: basieren auf funktionalen Abhängigkeiten

**Satz 4.2**

*Sei $R(U)$ ein Relationsschema, $r = \mathcal{I}(R)$ und $r \models X \rightarrow Y$.*

*Dann gilt auch $r \models \bowtie [XY, X(U - XY)]$.* $\square$

**Satz 4.3 (Datenbanken I)**

*Sei $R(A, B, C)$ ein Relationsschema und $\Sigma$ eine Menge von funktionalen Abhängigkeiten über $A, B, C$. $R$ ist verlustfrei zerlegbar in $R_1(A, B)$ und $R_2(A, C)$ (d.h., $r \models \bowtie [AB, AC]$) gdw. $\Sigma \models A \rightarrow B$ oder $\Sigma \models A \rightarrow C$.* $\square$

- Ein Datenbankschema $\mathbf{R} = (U, \Sigma)$ ist in BCNF/3NF genau dann, wenn $\mathbf{R}' = (U, \Sigma^+)$ in BCNF/3NF.
- 4NF basiert auf FAs und MVDs
  $\Rightarrow$ Bestimmung der Hülle von FAs und MVDs
- 5NF basiert analog auf Verbundabhängigkeiten

## Pro & Contra Normalisierung

Normalisiertes Schema:

+ Integritätsbedingungen über Joins definiert (“in allen Sälen dieselben Snacks” etc)
+ wenig Speicherbedarf
– Anfragen benötigen Joins

Nicht normalisiertes Schema:

– i.a. mehr Speicherbedarf (größere Tabellen)
  – Redundanz
+ Joins “vorberechnet”, evtl. auch Indexe
+ manchmal übersichtlicher (“wo kann ich einen Film von Hitchcock sehen und Kaffee trinken, und was kostet das dann”) in einer Tabelle.
– Integritätsbedingungen nicht mehr im Schema (im Beispiel könnten die Verbund-Abhängigkeiten verletzt werden)
  – “Update Anomalies”

## Frage

- Anfragen über nicht-normalisiertes Schema:
  kann man FAs und VAs irgendwie ausnutzen?

## Beispiel

**Beispiel 4.3**

*Mit welchen Schauspielern $a$ hat der Regisseur $d$ Filme gedreht?*

$$ans(d, a) \leftarrow Movies(t, d, a'), Movies(t, d', a)$$

*Wegen Movies $\models$ Title $\rightarrow$ Director kann man einfacher fragen:*

$$ans(d, a) \leftarrow Movies(t, d, a)$$

*Anmerkung: Anfrage an eine normalisierte Datenbasis:*

$$ans(d, a) \leftarrow directs(t, d), acts\_in(t, a)$$

$\square$

## 4.2 Tableaux

Ein **Tableau** über einem Datenbankschema $\mathbf{R}$ ist eine Instanz $\mathcal{T}$ von $\mathbf{R}$, wobei die Tupel in $\mathcal{T}$ aus Konstanten und Variablen gebildet werden.

- ausgezeichnete Variablen (distinguished Variables) $V, W, X, Y, Z$,
- nicht ausgezeichnete Variablen $V', V'', \ldots, W', W'', \ldots$

Beispiel:

| $R:$ | $A$ | $B$ | $C$ |
|---|---|---|---|
| | $X$ | $Y$ | $Z'$ |
| | $X$ | $Y'$ | $Z$ |
| | $c$ | $Y''$ | $Z$ |

Eine **Auswertung** eines Tableaus $\mathcal{T}$ ist eine Variablenzuweisung $\nu$ der Variablen in $\mathcal{T}$.

Eine **Einbettung** eines Tableaus $\mathcal{T}$ in eine Instanz $\mathcal{I}$ über $\mathbf{R}$ ist eine Auswertung $\nu$, so daß $\nu(\mathcal{T}) \subseteq \mathcal{I}$ gilt.

- Man bekommt Belegungen der ausgezeichneten Variablen:

$$\mathcal{T}(\mathcal{I}) := \{\nu(V, W, X, Y, Z) \mid \nu(\mathcal{T}) \subseteq \mathcal{I}\}$$

## TABLEAU-ANFRAGEN

Eine **Tableau-Anfrage** $T$ hat die Form $T = (\mathcal{T}, u)$, wobei

- $\mathcal{T}$ ein Tableau gebildet aus Konstanten und Variablen, und
- und $u$ ein Tupel über den Variablen aus $\mathcal{T}$

ist.

Die Antwortmenge zu $T = (\mathcal{T}, u)$ bezüglich einer Instanz $\mathcal{I}$ ist

$$\{\nu(u) \mid \nu(\mathcal{T}) \subseteq \mathcal{I}\}$$

**Beispiel 4.4**

*Das folgende Tableau $\mathcal{T}$ repräsentiert die Anfrage*
*"wo kann ich einen Film von Hitchcock sehen und Kaffee trinken, und was kostet das dann":*

| *Movies* | | | *Shows* | | | *Provides* | | |
|---|---|---|---|---|---|---|---|---|
| $Title$ | $Director$ | $Actor$ | $Theater$ | $Screen$ | $Title$ | $Theater$ | $Snack$ | $Price$ |
| $X_{ti}$ | "Hitchcock" | $X'_{ac}$ | $X_{th}$ | $X'_{sc}$ | $X_{ti}$ | $X_{th}$ | "Coffee" | $X_{pr}$ |

*Die Tableau-Anfrage* $(\mathcal{T}, (X_{th}, X_{pr}))$ *kann auch als konjunktive Anfrage gelesen werden:*

$$ans(X_{th}, X_{pr}) \leftarrow$$
$$Movies(X_{ti}, \text{"Hitchcock"}, X'_{ac}), Showings(X_{th}, X'_{sc}, X_{ti}), Provides(X_{th}, \text{"Coffee"}, X_{pr})$$

*Einbettung:*

$$\nu_1 = \{X_{ti} \mapsto \text{"The Birds"},\ X'_{ac} \mapsto \text{"Hedren"},\ X_{th} \mapsto \text{"Rex"},\ X'_{sc} \mapsto 1,\ X_{pr} \mapsto 2.35\}$$
$$\nu_2 = \{X_{ti} \mapsto \text{"The Birds"},\ X'_{ac} \mapsto \text{"Taylor"},\ X_{th} \mapsto \text{"Rex"},\ X'_{sc} \mapsto 1,\ X_{pr} \mapsto 2.35\}$$
$$\ldots$$

□

**Beispiel 4.5**

*Formulieren Sie die Anfrage aus Beispiel 4.4 als Anfrage gegen die angegebenen nicht-normalisierten Views.* □

**Beispiel 4.6**

*Zwei Personen wollen zusammen "The Birds" anschauen. Welche Snack-Varianten sind möglich?* □

**Beispiel 4.7**

*Formulieren Sie die Tableau-Anfrage zu Beispiel 4.3.* □

## 4.3 Äquivalenz von Tableaux

**Definition 4.4**

*Seien $\mathcal{T}$ und $\mathcal{T}'$ Tableaux über einem Datenbankschemas $\mathbf{R}$.*

- $\mathcal{T} \sqsubseteq \mathcal{T}'$ *falls für alle Instanzen $\mathcal{I}$ von $\mathbf{R}$, $\mathcal{T}(\mathcal{I}) \subseteq \mathcal{T}'(\mathcal{I})$,*
- $\mathcal{T} \equiv \mathcal{T}'$ *falls $\mathcal{T} \sqsubseteq \mathcal{T}'$ und $\mathcal{T}' \sqsubseteq \mathcal{T}$.* $\square$

Homomorphiesatz für Tableaux

**Satz 4.4**

*Seien $\mathcal{T}$ und $\mathcal{T}'$ Tableaux über $\mathbf{R}$. Dann ist $\mathcal{T} \sqsubseteq \mathcal{T}'$ gdw. es einen Homomorphismus $\theta : Var(\mathcal{T}') \to Var(\mathcal{T})$ (Enthaltenseinsabbildung) von $\mathcal{T}'$ nach $\mathcal{T}$ gibt, so dass*

- $\theta(\mathcal{T}') \subseteq \mathcal{T}$ *(Enthaltenseinseigenschaft)*
- *wenn $V$ eine ausgezeichnete Variable ist, dann ist auch $\theta(V)$ eine ausgezeichnete Variable* $\square$

# OPTIMIERUNG VON TABLEAUANFRAGEN: SUBSUMPTION

Voraussetzung:

- alle nicht-ausgezeichneten Variablen in $w_2$ treten nur einmal im Tableau auf

Eine Zeile $w_1$ eines Tableaux subsumiert eine andere Zeile $w_2$, wenn

- falls $\pi[A]w_2$ eine ausgezeichnete Variable ist, dann ist $\pi[A]w_1$ dieselbe Variable

$SUB(\mathcal{T}) := \mathcal{T}$ ohne die subsumierten Zeilen

**Satz 4.5**

$SUB(\mathcal{T}) \equiv \mathcal{T}$ $\square$

Beweis: einfache Anwendung des Homomorphiesatzes

... soweit bereits für allgemeine Anfragen behandelt.
Weiter: Ausnutzung von Abhängigkeiten

## Relative Äquivalenz

**Definition 4.5**

*Sei $\mathcal{F}$ eine Familie von Instanzen eines Datenbankschemas $\mathbf{R}$ und seien $Q_1, Q_2$ Anfragen.*

- *$Q_1$ ist relativ zu $\mathcal{F}$ in $Q_2$ enthalten, wenn $Q_1(\mathcal{I}) \subseteq Q_2(\mathcal{I})$ für jede Instanz $\mathcal{I} \in \mathcal{F}$. (Notation: $Q_1 \subseteq_{\mathcal{F}} Q_2$)*
- *$Q_1$ und $Q_2$ sind äquivalent relativ zu $\mathcal{F}$, wenn $Q_1 \subseteq_{\mathcal{F}} Q_2$ und $Q_2 \subseteq_{\mathcal{F}} Q_1$. (Notation: $Q_1 \equiv_{\mathcal{F}} Q_2$)* □

Für eine Menge $\Sigma$ von Abhängigkeiten ist $sat(\mathbf{R}, \Sigma)$ eine solche Familie

- $Q_1$ ist relativ zu $\Sigma$ in $Q_2$ enthalten, wenn $Q_1 \subseteq_{sat(\mathbf{R},\Sigma)} Q_2$. (Notation: $Q_1 \subseteq_{\Sigma} Q_2$)
- Analog $Q_1 \equiv_{\Sigma} Q_2$.

# ANFRAGEOPTIMIERUNG

**Ziel:** Sei eine Tableau-Anfrage $T$ und eine Menge $\Sigma$ von Abhängigkeiten gegeben. Erzeuge aus $T$ eine Tableau-Anfrage $T'$, so dass $T \equiv_\Sigma T'$ und $T'$ effizienter ausgewertet werden kann.

## 4.4 “The Chase”

- Gegeben: Datenbankschema $\mathbf{R}$, Abhängigkeiten $\Sigma$, und ein Tableau $\mathcal{T}$
- Idee: “Anwenden” von Abhängigkeiten aus $\Sigma$ auf $\mathcal{T}$
- Invariante/Korrektheit:

$$\mathcal{T}_{vorher}(\mathcal{I}) = \mathcal{T}_{nachher}(\mathcal{I})$$

  für alle $\mathcal{I} \in Sat(\mathbf{R}, \Sigma)$

## Beispiel

**Beispiel 4.8**

*Gegeben sei das Relationsschema $R(A, B, C, D)$ sowie die Abhängigkeiten $\bowtie$ [AB, BCD] und A $\rightarrow$ C, und die Anfrage*

$$ans(W, X, Y, Z) \leftarrow r(W, X, Y, Z'), r(W', X, Y', Z)$$

*Ausgangstableau*

| $\mathcal{T}$ | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| | $W$ | $X$ | $Y$ | $Z'$ |
| | $W'$ | $X$ | $Y'$ | $Z$ |

*Anwendung von $\bowtie$ [AB, BCD]: Das AB-Paar $\nu(WX)$ muss auch mit dem BCD-Tupel $\nu(XY'Z)$ vorkommen:*

| $\mathcal{T}'$ | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| | $W$ | $X$ | $Y$ | $Z'$ |
| | $W'$ | $X$ | $Y'$ | $Z$ |
| | $W$ | $X$ | $Y'$ | $Z$ |

*Anwendung von A $\rightarrow$ C: Wenn $A = \nu(W)$, dann $C = \nu(Y)$*

| $\mathcal{T}''$ | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| | $W$ | $X$ | $Y$ | $Z'$ |
| | $W'$ | $X$ | $Y'$ | $Z$ |
| | $W$ | $X$ | $Y$ | $Z$ |

*Die 1. und 2. Zeile in $\mathcal{T}''$ werden von der 3. subsumiert, können also gestrichen werden.* $\square$

# REGELN

Annahme: lineare Ordnung $<$ auf den Variablen von $\mathcal{T}$

- $X, Y, Z < X', X'', \ldots, Y', Y'', \ldots$

## Funktionale Abhängigkeiten

- $R(U)$ ein Relationsschema, $\sigma = V \to W$ mit $V, W \subseteq U$ eine FA,
- $a, b \in \mathcal{T}$ Zeilen des Tableaus
- mit demselben $V$-Wert: $\pi[V](a) = \pi[V](b)$ ,
- $a(W) \neq b(W)$. Sei weiter o.B.d.A. $a(W) < b(W)$.

Anwendung von $\sigma$ auf $(\mathcal{T}, t)$:

- Substitution $\theta = \mathrm{id}_{b(W)}^{a(W)}$ ("$a(W)$ für $b(W)$")
- neue Tableauanfrage: $\theta(\mathcal{T}, t)$

**Beispiel 4.9**

*Schema:* $Movie(Title, Director, Actor)$

*Abhängigkeit:* $Title \rightarrow Director$

*Anfrage:* $ans(T, D, A) \leftarrow movie(T, D', A),\ movie(T, D, A')$

*Start:*

| $Title$ | $Director$ | $Actor$ |
|---|---|---|
| $T$ | $D'$ | $A$ |
| $T$ | $D$ | $A'$ |

*Anwendung* $Title \rightarrow Director$*:*

| $Title$ | $Director$ | $Actor$ |
|---|---|---|
| $T$ | $D$ | $A$ |
| $T$ | $D$ | $A'$ |

*(1) subsumiert (2):*

| $Title$ | $Director$ | $Actor$ |
|---|---|---|
| $T$ | $D$ | $A$ |

*Vereinfachte Anfrage:* $ans(T, D, A) \leftarrow movie(T, D, A)$ □

## Verbundabhängigkeiten

- $R(U)$ ein Relationsschema, $\sigma = \bowtie [X_1, \ldots, X_n]$ mit $X_i \subseteq U$ eine VA,
- $u$ ein Tupel über $R$ so daß es $u_1, \ldots, u_n \in \mathcal{T}$ gibt mit $\pi_{X_i}(u_i) = \pi_{X_i}(u)$ für $1 \leq i \leq n$.

Anwendung von $\sigma$ auf $(\mathcal{T}, t)$:

- Neue Tableauanfrage: $(\mathcal{T} \cup \{u\}, t)$.

**Beispiel 4.10**

*Verbundabhängigkeit* $\bowtie [ABC, ADE]$.

| $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|
| $V$ | $W$ | $X$ | $Y'$ | $Z'$ |
| $V$ | $W'$ | $X'$ | $Y$ | $Z$ |

| $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|
| $V$ | $W$ | $X$ | $Y'$ | $Z'$ |
| $V$ | $W'$ | $X'$ | $Y$ | $Z$ |
| $V$ | $W$ | $X$ | $Y$ | $Z$ |

□

## Inklusionsabhängigkeiten

- $R(U_1)$ und $S(U_2)$ Relationsschemata,
- $\sigma = R[X] \subseteq S[Y]$ mit $X \subseteq U_1$ und $Y \subseteq U_2$ eine IA
- $u \in \mathcal{T}(R)$,
- es gibt kein $v \in \mathcal{T}(S)$ so dass $\pi[Y](v) = \pi[X](u)$

Anwendung von $\sigma$ auf $(\mathcal{T}, t)$:

- Sei $w$ ein Tupel über $S$ so dass $\pi[Y](w) = \pi[X](u)$ und $w$ überall sonst neue (nicht-ausgezeichnete) Variablen, die bzgl. $\leq$ größer als alle Variablen in $\mathcal{T}$ sind.
- Neue Tableauanfrage $(\mathcal{T}', t)$:
  - $\mathcal{T}'(P) = \mathcal{T}(P)$ für $P \in \mathbf{R} - \{S\}$,
  - $\mathcal{T}'(S) = \mathcal{T}(S) \cup \{w\}$.

**Beispiel 4.11**

*Abhängigkeiten:* $R_1.AB \subseteq R_2.EF$ *und* $R_2 \models \bowtie [AB, BC]$.

*Anfrage:* $ans(X,Y,Z) \leftarrow R_1(X,Y),\ R_2(X,Y',Z),\ R_2(X',Y,Z)$

*Start:*

| A | B |
|---|---|
| $X$ | $Y$ |

| E | F | G |
|---|---|---|
| $X$ | $Y'$ | $Z$ |
| $X'$ | $Y$ | $Z$ |

*Anwendung* $R_1.AB \subseteq R_2.EF$*:*

| A | B |
|---|---|
| $X$ | $Y$ |

| E | F | G |
|---|---|---|
| $X$ | $Y'$ | $Z$ |
| $X'$ | $Y$ | $Z$ |
| $X$ | $Y$ | $Z'$ |

*Anwendung* $R_2 \models \bowtie [AB, BC]$*:*

| A | B |
|---|---|
| $X$ | $Y$ |

| E | F | G |
|---|---|---|
| $X$ | $Y'$ | $Z$ |
| $X'$ | $Y$ | $Z$ |
| $X$ | $Y$ | $Z'$ |
| $X$ | $Y$ | $Z$ |

$R_2$ : *Subsumierte Zeilen 1-3 streichen.*

*Vereinfachte Anfrage:* $ans(X,Y,Z) \leftarrow R_1(X,Y),\ R_2(X,Y,Z)$. □

# KORREKTHEIT

**Satz 4.6**

*Wenn eine Anfrage $T'$ aus einer Anfrage $T$ durch Anwendung einer der Chase-Regeln zu einer Menge $\Sigma$ von Abhängigkeiten hervorgeht, gilt*

$$T' \equiv_\Sigma T$$

□

Beweis: Für jede einzelne Regel

- Anwendung von FAs
- Anwendung von VAs
- Anwendung von IAs
- (Anwendung von Subsumption/Homomorphiesatz)

$$T_{vorher}(\mathcal{I}) = T_{nachher}(\mathcal{I})$$

für alle $\mathcal{I} \in Sat(\mathbf{R}, \Sigma)$

Anwendung der Regeln erweitert ein Tableau so, dass eine Instanz über den im Tableau vorkommenden Variablen/Konstanten entsteht, die $\Sigma$ erfüllt.

## Beispiel/Aufgabe

Komplettieren Sie das untenstehende Tableau für $\Sigma = \{A \twoheadrightarrow BC,\ CD \twoheadrightarrow BE\}$

Hinweis: Bestimmen Sie äquivalente Verbundabhängigkeiten.

| $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|
| $a$ | $b$ | $c$ | $d$ | $e$ |
| $a$ | $b'$ | $c$ | $d'$ | $e$ |
| $a'$ | $b$ | $c$ | $d$ | $e'$ |

## CHASE-FOLGEN

**Definition 4.6**

*Eine* **Chase-Folge** *eines Tableaus $\mathcal{T}$ bzgl. $\Sigma$ ist eine (möglicherweise unendliche) Folge*

$$(\mathcal{T}_0 = \mathcal{T},\ \mathcal{T}_1,\ \ldots)$$

*so dass für jedes $i \geq 0$ das Tableau $\mathcal{T}_{i+1}$ durch Anwendung einer Abhängigkeit aus $\Sigma$ auf $\mathcal{T}_i$ entsteht.* □

**Lemma 4.1**

*Sei $R$ ein Relationsschema und $\Sigma$ eine Menge von Abhängigkeiten und $\mathcal{I} \in Sat(R, \Sigma)$.*
*Sei $(\mathcal{T}_0 := \mathcal{T},\ \mathcal{T}_1, \ldots)$ eine Chase-Folge zu $\Sigma$.*

*Dann gilt*

- *Für jede Zeile $u \in \mathcal{T}_i$: Sei $u'$ die entsprechende Zeile in $\mathcal{T}_{i+1}$.*
  1. *$u'$ subsumiert $u$,*
  2. *Für jede Auswertung $\nu$ von $\mathcal{T}_i$ so dass $\nu(\mathcal{T}_i) \subseteq \mathcal{I}$ ist, gilt $\nu(u) = \nu(u')$.*
- *Für das ganze Tableau gilt*
  1. *$\mathcal{T}_{i+1} \sqsubseteq \mathcal{T}_i$*
  2. *$\mathcal{T}_i \sqsubseteq_\Sigma \mathcal{T}_{i+1}$*
  3. *also: $\mathcal{T}_{i+1} \equiv_\Sigma \mathcal{T}_i$*
- *im Verhältnis zum Anfangszustand gilt*

$$\mathcal{T}_i \equiv_\Sigma \mathcal{T}$$

□

## Eigenschaften von Chase-Folgen

- Eine Chase-Folge heißt **terminal**, wenn sie endlich ist und keine Abhängigkeit mehr anwendbar ist;
- das letzte Element ist das Resultat und wird mit $chase(\mathcal{T}, \Sigma)$ bezeichnet.

Zu zeigen? was?

## ... ZU ZEIGEN IST

- Ist $chase(\mathcal{T}, \Sigma)$ wohldefiniert?
- d.h., existiert es (Folge ist endlich) und ist das Ergebnis eindeutig?

# EXKURS: REDUKTIONSSYSTEME

Für gegebenes $\mathbf{R}$, $\Sigma$ bilden die Chase-Regeln ein **Reduktionssystem** $\mathcal{R} = (\mathcal{D}, \rightarrow)$ (i.a. ist auf $\mathcal{D}$ bereits eine "einfache" $\equiv$-Relation gegeben):

- $\mathcal{D}$: Domain (hier: Menge der Tableau-Anfragen über $\mathbf{R}$)
- $\rightarrow \subseteq \mathcal{D} \times \mathcal{D}$ rekursiv aufzählbare Relation (i.a. durch Regeln gegeben)
- Äquivalenz "$\Leftrightarrow$": reflexive, transitive, symmetrische Hülle von $\rightarrow$

- Anmerkung: Der GYO-Algorithmus ist ebenfalls ein Reduktionssystem

## Eigenschaften: Terminierung

**Definition 4.7**

*Sei $\mathcal{R} = (\mathcal{D}, \rightarrow)$ ein Reduktionssystem.*

- *Die Ableitungsrelation $\rightarrow$ ist* wohlfundiert*, wenn es keine unendliche Folge $(d_i)_{i \in \mathbb{N}} \in \mathcal{D}^{\mathbb{N}}$ gibt so dass $d_i \rightarrow d_{i+1}$ für alle $i$ gilt.*
- *Elemente $d$ so dass*

$$d \rightarrow d' \Rightarrow d \equiv d'$$

*gilt, heissen* irreduzible *Elemente.* □

Wohldefiniertheit erlaubt Anwendung von Induktion:

Sei $\mathcal{E}$ eine Eigenschaft über $\mathcal{D}$. Wenn $\mathcal{E}(d)$ für alle irreduziblen Elemente gilt, und

$$\forall d \in \mathcal{D} : ((\forall d' \in \mathcal{D} : d \rightarrow d' \Rightarrow \mathcal{E}(d')) \Rightarrow \mathcal{E}(d))$$

gilt, so gilt $\quad \forall d \in \mathcal{D} : \mathcal{E}(d')$.

## Eigenschaften: Konfluenz, Church-Rosser

Wie testet man die Äquivalenz “$d_1 \Leftrightarrow d_2$” zweier Elemente?

(Beachte Äquivalenz $\equiv$ auf $\mathcal{D}$ vs. Äquivalenz $\Leftrightarrow$ auf $\mathcal{D}$ bzgl $\mathcal{R}$.)

**Definition 4.8**

*Eine Relation “$\rightarrow$” besitzt die* **Church-Rosser-Eigenschaft** *(CRE), wenn:*

- $d_1 \Leftrightarrow d_2$ *genau dann, wenn es* $d_1' \equiv d_2'$ *gibt, so daß* $d_1 \xrightarrow{*} d_1'$ *und* $d \xrightarrow{*} d_2'$. $\square$

Kritisch ist hier die Richtung von links nach rechts.

**Definition 4.9**

*Eine Relation “$\rightarrow$” ist* **konfluent**, *wenn:*

- *für alle* $d, d_1, d_2$, *wenn aus* $d \xrightarrow{*} d_1$ *und* $d \xrightarrow{*} d_2$ *folgt, dass es* $d_1' \equiv d_2'$ *gibt, so daß* $d_1 \xrightarrow{*} d_1'$ *und* $d_2 \xrightarrow{*} d_2'$. $\square$

Beide Eigenschaften sind äquivalent.

## Eigenschaften: Minimale Elemente, Normalform

- Wenn $\mathcal{R}$ konfluent ist, gibt es zu jedem Element gibt es höchstens ein irreduzibles Element.
- Wenn $\mathcal{R}$ wohlfundiert ist, ist es konfluent genau dann, wenn es zu jedem Element *genau* ein irreduzibles Element gibt.

**Definition 4.10**

*Wenn es zu jedem Element $d$* genau *ein irreduzibles Element gibt, so ist dieses Element die* Normalform *von $d$ (bzw. der gesamten Äquivalenzklasse (bzgl. $\Leftrightarrow$) zu $d$).* $\square$

Wünschenswert also:

- Wohlfundiertheit/Terminierung und
- Konfluenz/Church-Rosser

- Anmerkung: GYO terminiert (offensichtlich) und ist im positiven Fall konfluent (nicht so offensichtlich, vgl. Übung 11.6.)

## EIGENSCHAFTEN DES CHASE

**Satz 4.7**

*Sei $\Sigma$ eine Menge von FA's und VA's.*

*Wohlfundiertheit: Jede Chase-Folge zu $\mathcal{T}$ und $\Sigma$ ist endlich und kann zu einer terminalen Folge verlängert werden.* $\square$

Beweis:

- es werden keine neuen Variablen eingeführt $\rightsquigarrow$ nur $k := \#Variablen^{\#Attribute}$ viele Tupel (Zeilen) möglich $\rightsquigarrow$ nur $2^k$ verschiedene Tableaux.

Zu zeigen: kein Tableau tritt in einer Chase-Folge unendlich oft auf:

- FA-Regel: mit jeder Anwendung verschwindet eine Variable
  Also nur endlich oft möglich
- Annahme: es gibt eine unendliche Folge. Dann gibt es ein $i$, so dass ab dem $i$-ten Schritt nur noch VA-Regeln angewendet werden.
  Jede solche Anwendung erzeugt ein neues Tupel. Widerspruch zu der Feststellung, dass nur endlich viele Tupel möglich sind.

# Eigenschaften des Chase

**Satz 4.8**

*Sei $\Sigma$ eine Menge von FA's und VA's.*

- *Das Resultat einer terminalen Chase-Folge erfüllt (als Instanz gesehen) $\Sigma$.*
  *Beweis: falls nein, wende entsprechende Regel an.*
- *Konfluenz: Für gegebenes $\mathcal{T}$ haben alle (terminalen) Folgen dasselbe Resultat.*
  *Beweis: Routine, [Maier, Theorem 8.9]*
- *Damit ist* $chase(\mathcal{T}, t, \Sigma)$ *wohldefiniert.*
- *Alle Elemente einer Chase-Folge sind zueinander äquivalent (relativ zu $\Sigma$) und insbesondere gilt für jedes Tableau $\mathcal{T}$*

$$\mathcal{T} \equiv_\Sigma chase(\mathcal{T}, \Sigma) \ .$$

- *(Church-Rosser: Ist $\mathcal{T}_1 \equiv_\Sigma \mathcal{T}_2$, so ist $chase(\mathcal{T}_1, \Sigma) \equiv chase(\mathcal{T}_2, \Sigma)$.)* $\square$

## EIGENSCHAFTEN DES CHASE

Enthält $\Sigma$ IA's, so kann $(\mathcal{T}, t)$ bzgl. $\Sigma$ endliche terminale oder auch unendliche Chase-Folgen besitzen. Die Resultate zweier terminaler Folgen müssen nicht notwendigerweise isomorph sein:

- Durch IAs werden neue Variablen eingeführt, die sich über VAs beliebig verbreiten können (und zu wieder neuen IAs führen)

Im folgenden werden nur FAs und VAs betrachtet. Damit kann man sich außerdem auf einzelne Relationen beschränken.

## 4.5 Anwendungen des Chase

- bis jetzt: rein auf syntaktischer Basis
  - Abhängigkeiten
  - Anfrage

Äquivalente Mengen von Abhängigkeiten

**Satz 4.9**

*Seien $\Sigma \equiv \Sigma'$ zwei äquivalente Mengen von FAs und VAs über $R$; sei weiter $\mathcal{T}$ ein Tableau über $R$. Dann ist $chase(\mathcal{T}, \Sigma) = chase(\mathcal{T}, \Sigma')$.* $\square$

Beweis: Benutze $chase(\mathcal{T}, \Sigma) = chase(\mathcal{T}, \Sigma \cup \{\sigma\})$ falls $\Sigma \models \sigma$

$\Rightarrow$ hängt nur von der Semantik von $\Sigma$ ab.

## Äquivalente Anfragen

**Satz 4.10 (Church-Rosser-Eigenschaft des Chase)**

*Seien $\Sigma$ eine Menge von FAs und VAs. Dann gilt*

$$\mathcal{T}_1 \equiv_\Sigma \mathcal{T}_2 \qquad \textit{gdw.} \qquad chase(\mathcal{T}_1, \Sigma) \equiv chase(\mathcal{T}_2, \Sigma)\ . \qquad \square$$

Beweis (Reduktion auf “Konfluenz gdw. Church-Rosser”):

- Anwendungssemantik: $\mathcal{T}_1 \equiv_\Sigma \mathcal{T}_2$
- Nachweis von $\equiv_\Sigma$ durch das Chase-Reduktionssystem $\mathcal{R}_{chase} = (Tableaux, Chase - Regeln)$
- $\Leftrightarrow$ induziert durch $\mathcal{R}_{chase}$, also $\equiv_\Sigma = \Leftrightarrow$, insb. Anwendungssemantik: $\mathcal{T}_1 \Leftrightarrow \mathcal{T}_2$
- $\mathcal{T}_i \xrightarrow{*} \mathcal{T}_i'$ und $\mathcal{T}_i'$ irreduzibel gdw. $\mathcal{T}_i' \equiv chase(\mathcal{T}_i, \Sigma)$.
- $\mathcal{R}_{chase}$ ist konfluent (Satz 4.8), also auch Church-Rosser.
- Damit also $chase(\mathcal{T}_1, \Sigma) \equiv chase(\mathcal{T}_2, \Sigma)$ .

Beweis von Satz 4.10 “zu Fuß”:

offensichtlich: $\underbrace{chase(\mathcal{T}_1, \Sigma)}_{=: \mathcal{T}_1'} \equiv_\Sigma \mathcal{T}_1 \equiv_\Sigma \mathcal{T}_2 \equiv_\Sigma \underbrace{chase(\mathcal{T}_2, \Sigma)}_{=: \mathcal{T}_2'}$

Weiterhin erfüllen beide $\mathcal{T}_i'$ als Instanz $\Sigma$ (Satz 4.8).

zu zeigen: es gibt eine Enthaltenseins-Abbildung $\theta$ von $\mathcal{T}_2'$ nach $\mathcal{T}_1'$ (und analog umgekehrt):

Für die Zeile $t_1' := (X_1, X_2, \ldots, X_n)$ der ausgezeichneten Variablen in $\mathcal{T}_1'$ gilt $t_1' \in \mathcal{T}_1'(\mathcal{T}_1')$. Wegen Äquivalenz auf $Sat(\mathbf{R}, \Sigma)$ gilt

$$t_1' \in \mathcal{T}_1'(\mathcal{T}_1') \Rightarrow t_1' \in \mathcal{T}_1(\mathcal{T}_1') \Rightarrow t_1' \in \mathcal{T}_2(\mathcal{T}_1') \Rightarrow t_1' \in \mathcal{T}_2'(\mathcal{T}_1')$$

und aus $t_1' \in \mathcal{T}_2'(\mathcal{T}_1')$ bekommt man eine Abbildung $\theta$ so daß $\theta(\mathcal{T}_2') \subseteq \mathcal{T}_1'$.

## 4.6 Anfrageoptimierung mittels Chase

- Beispiel: Folie 29

**Definition 4.11**

*Sei $\mathcal{T}$ ein Tableau.*

- $min(\mathcal{T})$ *ist das durch Anwendung des Homomorphiesatzes minimierte Tableau*
- $\mid \mathcal{T} \mid$ *ist die Kardinalität (Anzahl Zeilen) des Tableaus* $\mathcal{T}$. $\square$

**Beispiel 4.12**

*Sei* $\sigma = \bowtie [AB, BCD]$ *und* $\mathcal{T}$ *gegeben:*

| $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|
| $V$ | $W$ | $X'$ | $Y'$ |
| $V'$ | $W$ | $X$ | $Y$ |

$\sigma$ *anwenden:* $\mathcal{T}' =$

| $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|
| $V$ | $W$ | $X'$ | $Y'$ |
| $V'$ | $W$ | $X$ | $Y$ |
| $V$ | $W$ | $X$ | $Y$ |

- $\mathcal{T}$ *ist minimal und somit* $|\, min(\mathcal{T})\,| = 2$.
- $\mathcal{T}' = chase(\mathcal{T}, \sigma)$ *ist nicht minimal (1. und 2. Zeile von 3. subsumiert).* $|\, min(chase(\mathcal{T}, \sigma))\,| = 1 < |\, min(\mathcal{T})\,|$. $\square$

**Satz 4.11**

*Wenn* $\Sigma$ *nur FAs enthält, gilt*

$$|\; min(chase(\mathcal{T}, \Sigma)) \;| \leq |\; min(\mathcal{T}) \;| \qquad \square$$

## Negativbeispiel mit VAs

**Beispiel 4.13**

*Sei* $\sigma = \bowtie [AB, CD]$ *und* $\mathcal{T}$ *gegeben:*

| $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|
| $V$ | $W$ | $X$ | $Y'$ |
| $V'$ | $W'$ | $X'$ | $Y$ |

$\sigma$ *anwenden* $\mathcal{T}' =$

| $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|
| $V$ | $W$ | $X$ | $Y'$ |
| $V'$ | $W'$ | $X'$ | $Y$ |
| $V'$ | $W'$ | $X$ | $Y'$ |
| $V$ | $W$ | $X'$ | $Y$ |

- $(\mathcal{T}, t)$ *ist minimal und somit* $|\,min(\mathcal{T})\,| = 2$.
- $\mathcal{T}' = chase(\mathcal{T}, \sigma)$, $(\mathcal{T}', t')$ *minimal und somit* $|\,min(chase(\mathcal{T}, \sigma))\,| = 4 > |\,min(\mathcal{T})\,|$. $\square$

- Im allgemeinen aber gute Ergebnisse.

**Beispiel 4.14**

*Gegeben:* $\Sigma = \{B \rightarrow D,\ D \rightarrow C,\ \bowtie [AB, ACD]\}$ *und die Anfrage*

$$ans(W, X, Z) \leftarrow r(W', X, Y', Z'), r(W, X', Y', Z'), r(W, X'', Y'', Z)$$

$$q = (\pi[AB](\pi[BCD](R) \bowtie \pi[ACD](R))) \bowtie \pi[AD](R)\ .$$

| $\mathcal{T}$ | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| | $W'$ | $X$ | $Y'$ | $Z'$ |
| | $W$ | $X'$ | $Y'$ | $Z'$ |
| | $W$ | $X''$ | $Y''$ | $Z$ |

| $\mathcal{T}'$ | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| | $W'$ | $X$ | $Y'$ | $Z'$ |
| | $W$ | $X'$ | $Y'$ | $Z'$ |
| | $W$ | $X''$ | $Y''$ | $Z$ |
| | $W$ | $X'$ | $Y''$ | $Z$ |
| | $W$ | $X''$ | $Y'$ | $Z'$ |

| $\mathcal{T}''$ | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| | $W'$ | $X$ | $Y'$ | $Z'$ |
| | $W$ | $X'$ | $Y'$ | $Z$ |
| | $W$ | $X''$ | $Y''$ | $Z$ |
| | $W$ | $X'$ | $Y''$ | $Z$ |
| | $W$ | $X''$ | $Y'$ | $Z$ |

- $\mathcal{T}$ *ist minimal. Chase (2,3) mit* $\bowtie [AB, ACD]$ *erzeugt* $\mathcal{T}'$.
- *Chase mit zuerst* $B \rightarrow D$ *und dann* $D \rightarrow C$ *erzeugt* $\mathcal{T}''$.
- $\mathcal{T}''$ *wird minimiert zu*

| $\mathcal{T}'''$ | $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|---|
| | $W'$ | $X$ | $Y'$ | $Z'$ |
| | $W$ | $X'$ | $Y'$ | $Z$ |

□

**Beispiel 4.15**

*Sei* $P = (A, B, C)$ *und* $Q = (D, E, F)$ *und* $\Sigma = \{Q[DE] \subseteq P[AB], A \to B\}$
*Anfrage:* $ans(Y, Z) \leftarrow Q(X', Y, Z'),\ Q(X', Y', Z)$.

$\mathcal{T}$

$$\begin{array}{c|ccc} P & A & B & C \\ \hline & & & \end{array}$$

$$\begin{array}{c|ccc} Q & D & E & F \\ \hline & X' & Y & Z' \\ & X' & Y' & Z \end{array}$$

$\mathcal{T}_1$

$$\begin{array}{c|ccc} P & A & B & C \\ \hline & X' & Y & W' \\ & X' & Y' & W'' \end{array}$$

$$\begin{array}{c|ccc} Q & D & E & F \\ \hline & X' & Y & Z' \\ & X' & Y' & Z \end{array}$$

$\mathcal{T}_2$

$$\begin{array}{c|ccc} P & A & B & C \\ \hline & X' & Y & W' \\ & X' & Y & W'' \end{array}$$

$$\begin{array}{c|ccc} Q & D & E & F \\ \hline & X' & Y & Z' \\ & X' & Y & Z \end{array}$$

$\mathcal{T}_3$

$$\begin{array}{c|ccc} P & A & B & C \\ \hline & X' & Y & W' \end{array}$$

$$\begin{array}{c|ccc} Q & D & E & F \\ \hline & X' & Y & Z \end{array}$$

- $\mathcal{T}_1$ *ergibt sich durch Anwendung der IA*
- $\mathcal{T}_2$ *ergibt sich durch Anwendung der FA* ($Y' \mapsto Y$)
- $\mathcal{T}_3$ *ergibt sich durch Homomorphiesatz*
- *vereinfachte Anfrage:* $ans(Y, Z) \leftarrow Q(X', Y, Z)$.
- ($P$*-Anteil weglassen wegen* $Q(X, Y, Z') \to P(X, Y, W')$) $\square$

# Aufwand

- bei bekanntem Schema ist Chase polynomiell
- trotzdem erheblicher Aufwand als “Vorverarbeitung”
- lohnt sich bei
  - View-Definitionen (Übersetzung bei Compile-Time)
  - vorcompilierten Anfragen (vgl. JDBC *prepared statements*)
  - run-time: z.B. bei korrelierten Subqueries (die mehrmals ausgewertet werden müssen).

# 4.7 Implizierte Abhängigkeiten

## VERBUND-ABHÄNGIGKEITEN

- Gegeben: ein Relationsschema $R(U)$ und eine Menge $\Sigma$ von Abhängigkeiten über $U$.
- Frage: Ist die Zerlegung $\rho = \{R_1, \ldots, R_k\}$ mit $R_i \subseteq U$ verlustfrei?
- Umformulierung:
  - Garantiert $\Sigma$, dass *jedes* Tupel, das im Join $\pi_{R_1}(R) \bowtie \ldots \bowtie \pi_{R_k}(R)$ enthalten ist, auch in $R$ enthalten sein muss?
  - Äquivalent: $\Sigma \models \bowtie [R_1, \ldots, R_k]$?
- Eine geeignete Chase-Folge "probiert" genau dies systematisch aus.

**Satz 4.12**

*Sei $R(U)$ mit $U = (A_1, \ldots, A_n)$ ein Relationsschema.*
*Die Zerlegung $\rho = \{R_1, \ldots, R_k\}$ ist verlustfrei gdw. $(X_1, \ldots, X_n) \in chase(\mathcal{T}_\rho, \Sigma)$.* $\square$

Tableau $\mathcal{T}_\rho$:

- $n$ Spalten, $k$ Zeilen
  Spalte $j$ für Attribut $A_j$, Zeile $i$ für $R_i$
- $T_{(i,j)} = X_j$ sofern $A_j \in R_i$; sonst $T_{(i,j)} = Y_{ij}$
- $X_i$ sind die ausgezeichneten Variablen

Beispiel:

- $U = ABCD$,
- $\Sigma = \{A \to D, \; \bowtie [AB, BCD]\}$,
- $\rho = (AB, BC, AD)$

## Beispiel:

$U = ABCD,\ \Sigma = \{A \to D, \bowtie [AB, BCD]\},\ \rho = (AB, BC, AD)$

$$\mathcal{T}_\rho = \begin{array}{cccc} A & B & C & D \\ \hline W & X & Y' & Z' \\ W' & X & Y & Z'' \\ W & X''' & Y''' & Z \end{array}$$

$\bowtie [AB, BCD]$ auf (1) und (2):

| $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|
| $W$ | $X$ | $Y'$ | $Z'$ |
| $W'$ | $X$ | $Y$ | $Z''$ |
| $W$ | $X'''$ | $Y'''$ | $Z$ |
| $W$ | $X$ | $Y$ | $Z''$ |

$A \to D$ auf (3) und (4):

| $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|
| $W$ | $X$ | $Y'$ | $Z'$ |
| $W'$ | $X$ | $Y$ | $Z''$ |
| $W$ | $X'''$ | $Y'''$ | $Z$ |
| $W$ | $X$ | $Y$ | $Z$ |

Als Anfrage geschrieben:

$$? - R(W, X, Y', Z')\ , R(W', X, Y, Z'')\ , R(W, X''', Y''', Z)$$

kann unter $\Sigma$ zu $\quad ? - R(W, X, Y, Z) \quad$ vereinfacht werden.

## Anmerkungen

- Anwendung der Regel für Verbund-Abhängigkeit $\bowtie [\rho]$ würde *direkt* das Tupel $(X_1, \ldots, X_n)$ im Tableau $\mathcal{T}_\rho$ erzeugen.
- Genauso kann man mit dem Chase ein Gegenbeispiel erzeugen, wenn eine Zerlegung nicht verlustfrei ist:
  - $chase(\mathcal{T}_\rho, \Sigma)$ erzeugt eine Instanz, die $\Sigma$ erfüllt, und $\bowtie [\rho]$ nicht erfüllt (da das Tupel $(X_1, \ldots, X_n)$ fehlt.

# FUNKTIONALE ABHÄNGIGKEITEN

- Gegeben: ein Relationsschema $R(U)$, $U = \{A_1, \ldots, A_n\}$ und eine Menge $\Sigma$ von Abhängigkeiten
- Frage: Gilt $\Sigma \models V \to W$ für $V, W \subseteq U$?

Definiere $\mathcal{T}_V$ wie folgt:

- Zwei Zeilen, $n$ Spalten
- erste Zeile: überall ausgezeichnete Variablen,
- zweite Zeite: bei $V$ dieselben ausgezeichneten Variablen, sonst überall unterschiedliche nicht-ausgezeichnete Variablen

**Satz 4.13**

$\Sigma \models V \to W$ *gdw.* $chase(\mathcal{T}_V, \Sigma)$ *nur ausgezeichnete Variablen in den* $W$*-Spalten hat.* $\square$

**Korollar 4.1**

*Sei $R(U)$ ein Relationsschema und $\Sigma$ eine Menge von Abhängigkeiten über $U = (A_1, \ldots, A_n)$.*

*Für $X \subseteq U$ ist $X^+$ die Menge aller $A_i$ so dass die $A_i$-Spalte von $\mathcal{T}_X$ nur ausgezeichnete Variablen enthält.* □

## Zusammenfassung

- Implikation von Verbund-Abhängigkeiten und funktionalen Abhängigkeiten ist entscheidbar:
  - Testen von Verbundabhängigkeiten
  - Aufzählen der Attributhülle (funktionale Abhängigkeiten)
- Sonderfall von Verbundabhängigkeiten: mehrwertige Abhängigkeiten
  - können mit demselben Verfahren ebenfalls aufgezählt werden

## Mehrwertige Abhängigkeiten

**Satz 4.14**

*$\Sigma \models V \twoheadrightarrow W$ gdw. $chase(\mathcal{T}_V, \Sigma)$ eine Zeile enthält, in der ausgezeichnete Variablen genau in allen $WV^+$-Spalten stehen.* $\square$

Beispiel: $U = ABCDE,\ \Sigma = \{B \to C,\ \bowtie[ABC, CDE]\}$

$\mathcal{T}_B =$

| $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|
| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ |
| $Y_1$ | $X_2$ | $Y_3$ | $Y_4$ | $Y_5$ |

$B \to C$ auf (1) und (2):

| $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|
| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ |
| $Y_1$ | $X_2$ | $X_3$ | $Y_4$ | $Y_5$ |

$\bowtie[ABC, CDE]$ aus (1) und (2):

| $A$ | $B$ | $C$ | $D$ | $E$ |
|---|---|---|---|---|
| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ |
| $Y_1$ | $X_2$ | $X_3$ | $Y_4$ | $Y_5$ |
| $X_1$ | $X_2$ | $X_3$ | $Y_4$ | $Y_5$ |
| $Y_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ |

- $B^+ = \{BC\}$
- $B \twoheadrightarrow ADE,\ B \twoheadrightarrow \emptyset,\ B \twoheadrightarrow A,\ B \twoheadrightarrow DE.$

## Abhängigkeitsbasis

- $DEP(V) := mdsb(\{W \mid \Sigma \models V \twoheadrightarrow W\})$
  ($mdsb$ = minimum disjoint set basis)
- $\Sigma \models V \twoheadrightarrow W$ gdw. $W$ die Vereinigung von Mengen aus $DEP(V)$ ist.
- $Q := \{W \mid \Sigma \models V \twoheadrightarrow W$ and $X^+ \cap W = \emptyset\}$ wie oben berechnet.
- $DEP(V) = mdsb(Q) \cup \{\{A\} \mid A \in X^+\}$.

## im obigen Beispiel:

- $Q = \{\{ADE\}, \{A\}, \{DE\}\}$
- $mdsb(Q) = \{\{A\}, \{DE\}\}$
- $B^+ = \{B, C\}$
- $DEP(B) = \{\{A\}, \{B\}, \{C\}, \{DE\}\}$

## ZERLEGUNG VON VERBUND-ABHÄNGIGKEITEN IN MEHRWERTIGE ABHÄNGIGKEITEN

**Definition 4.12**

*Eine Verbund-Abhängigkeit* $\bowtie[\rho]$ ist azyklisch, *wenn der Hypergraph zu* $\rho$ *azyklisch ist.* $\square$

**Lemma 4.2**

*Sei* $R(U)$ *ein Relationsschema und* $\sigma = \bowtie[R_1, \ldots, R_n]$ *mit* $R_i \subseteq U$ *eine Verbund-Abhängigkeit. Die folgenden Aussagen sind äquivalent für* $V, W \subseteq U$*:*

- $\sigma \models V \twoheadrightarrow W$,
- *es gibt kein* $R_i$*, so dass* $R_i \cap W \neq \emptyset$ *und* $R_i \cap (U - VW) \neq \emptyset$.
- $W$ *ist die Vereinigung von Zusammenhangskomponenten des Hypergraphs* $\rho\,|_{U-V}$. $\square$

## AZYKLISCHE VERBUND-ABHÄNGIGKEITEN

**Satz 4.15**

*Eine Verbund-Abhängigkeit $\sigma$ ist azyklisch, wenn es eine Menge $\Sigma$ von mehrwertigen Abhängigkeiten gibt, die äquivalent zu $\sigma$ ist.* $\square$

Praktische Relevanz und Verbindung zu vorigen Kapiteln:

Bei Normalisierung einer *azyklischen* Verbund-Abhängigkeit läßt sich die ursprüngliche Relation als Join-View erstellen, dessen Berechnung durch Verwendung voller Reduzierer optimiert werden kann.