Kapitel 5: Grundlagen von Anfragesprachen

Sprachparadigmen

- Relationenalgebra (in der Vorlesung)
- Relationenkalkül (siehe Literatur)

empfohlene Lektüre:

A Relational Model of Data for Large Shared Data Banks

E. F. Codd
IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on *n*-ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

In: Communications of the ACM, Volume 13 , Issue 6 (June 1970).

Kann gegoogelt werden.

5.1 Das relationale Datenmodell

Überblick

- Relationale Datenbanken repräsentieren Zustände einer Miniwelt durch Relationen.
- Wir unterscheiden die *Definition* der Relationen von den konkreten zeitabhängigen *Inhalten* (Zuständen).
- Wir reden von einem *Schema* einer Relation, wenn wir die Definition meinen, und von einer *Instanz* einer Relation, wenn wir eine entsprechende Menge von Tupeln meinen.

Objekte, Beziehungen, Attribute

- Objekte und Beziehungen werden durch ihre Eigenschaften repräsentiert. Eigenschaften werden im relationalen Modell als *Attribute* bezeichnet.
- Sei $X = \{A_1, \ldots, A_k\}$ eine endliche *Attributmenge*, wobei $k \geq 1$.
- Jedes Attribut $A \in X$ besitzt einen nicht-leeren *Wertebereich* $dom(A)$.
- Die Vereinigung aller Wertebereiche ergibt sich dann zu $dom(X) = \cup_{A \in X} dom(A)$.

Tupel

- Die Eigenschaften der Objekte und Beziehungen werden zu *Tupeln* zusammengefasst.
- Ein *Tupel* μ über Attributmenge X ist eine Abbildung

$$\mu : X \longrightarrow dom(X),$$

  wobei $(\forall A \in X)\mu(A) \in dom(A)$.
- Sei Tup(X) im folgenden die Menge aller Tupel über X.

Beispiel

$\mu_1 =$ {`MatrNr` → 1223, `Name` → Hans Eifrig, `Adresse` → Seeweg 20, `Semester` → 2}

$\mu_2 =$ {`MatrNr` → 3434, `Name` → Lisa Lustig, `Adresse` → Bergstraße 11, `Semester` → 4}

$\mu_3 =$ {`MatrNr` → 1234, `Name` → Maria Gut, `Adresse` → Am Bächle 1, `Semester` → 2}

Tupel als Abbildungen versus Tupel als Vektoren

$$\mu_1 = \{\texttt{MatrNr} \to 1223, \texttt{Name} \to \text{Hans Eifrig}, \texttt{Adresse} \to \text{Seeweg 20}, \texttt{Semester} \to 2\}$$

$$\mu_2 = \{\texttt{MatrNr} \to 1223, \texttt{Adresse} \to \text{Seeweg 20}, \texttt{Semester} \to 2, \texttt{Name} \to \text{Hans Eifrig}\}$$

μ'_1 = `(1223,Hans Eifrig,Seeweg 20,2)`

μ'_2 = `(1223,Seeweg 20,2,Hans Eifrig)`

$\mu_1 = \mu_2$, aber $\mu'_1 \neq \mu'_2$.

Relation

- Eine *Relation* r über einer Attributmenge X ist eine *endliche* Menge $r \subseteq \text{Tup}(X)$.
- Sei R ein *Relationsbezeichner*.
  Ein *(Relations)-Schema* zu R hat die Form $R(X)$. X ist hier eine endliche Attributmenge, das so genannte *Format* des Schemas.
  Anstatt $R(\{A_1, \ldots, A_k\})$ schreiben wir auch $R(A_1, \ldots, A_k)$.
  k ist die *Stelligkeit* des Relationsbezeichners.

  Auch:

$$R(A_1 : dom(A_1), \ldots, A_k : dom(A_k))$$

Datenbank

- Ein *(relationales) Datenbank-Schema* $\mathcal{R}$ ist gegeben durch eine Menge von (Relations-) Schemata,

$$\mathcal{R} := \{R_1(X_1), \ldots, R_m(X_m)\},$$

  bzw. $\mathcal{R} = \{R_1, \ldots, R_m\}$.
- Eine *Instanz* $\mathcal{I}$ zu einem relationalen Datenbankschema $\mathcal{R} = \{R_1, \ldots, R_m\}$ ist eine Menge von endlichen Relationen $\mathcal{I} := \{r_1, \ldots, r_m\}$, wobei $r_i \subseteq \text{Tup}(X_i)$ *Instanz* zu R_i für $1 \leq i \leq m$.

5.2 Relationenalgebra

Basisoperatoren

- Attribute aus Relationen herausstreichen: *Projektion* π.
- Tupel aus Relationen auswählen: *Selektion* σ.
- Relationen miteinander verknüpfen: *Verbund* $\bowtie$.
- Relationen wie Mengen verarbeiten: *Vereinigung* $\cup$, *Differenz* $-$.

Beispiel Projektion einer Relation

$$r = \begin{array}{ccc} A & B & C \\ \hline a & b & c \\ a & a & c \\ c & b & d \end{array} \qquad \pi[A, C]r = \qquad \begin{array}{cc} A & C \\ \hline a & c \\ c & d \end{array}$$

Projektion einer Relation

- Sei $r \subseteq \mathrm{Tup}(X)$ eine Relation und $\emptyset \subset Y \subseteq X$.
- Der Ausdruck $\pi[Y]r$ heißt *Projektion* der Relation r auf Y. Es gilt:

$$\pi[Y]r := \{\mu \in \mathrm{Tup}(Y) \mid \exists \mu' \in r, \text{ so dass } \mu(A) = \mu'(A), A \in Y.\}.$$

Selektionsbedingung

- Seien $A, B \in X$, $a \in dom(A)$ und sei $\theta \in \{=, \neq, <, \leq, >, \geq\}$ ein arithmetischer Vergleichsoperator.
- Eine (atomare) *Selektionsbedingung* α (bezüglich X) ist ein Ausdruck der Form $A\,\theta\,B$, bzw. $A\,\theta\,a$, bzw. $a\,\theta\,A$.
- Ein Tupel $\mu \in \text{Tup}(X)$ *erfüllt* eine Selektionsbedingung α, wenn gerade $\mu(A)\,\theta\,\mu(B)$, bzw. $\mu(A)\,\theta\,a$, bzw. $a\,\theta\,\mu(A)$.
- Atomare Selektionsbedingungen können mittels $\wedge$, $\vee$, $\neg$, $($, $)$ zu Formeln verallgemeinert werden.

Beispiel

$$
\begin{aligned}
&X = \{A, B, C\}.\\
&\mu_1 = (A \to 2, B \to 2, C \to 1),\ \mu_2 = (A \to 2, B \to 3, C \to 2)\\
&\alpha_1 = (A = B),\ \alpha_2 = ((B > 1) \wedge (C > 1))
\end{aligned}
$$

Welche Tupel erfüllen welche Selektionsbedingungen?

Beispiel Selektion

$$r = \begin{array}{ccc} A & B & C \\ \hline a & b & c \\ d & a & f \\ c & b & d \end{array} \qquad \sigma[B = b]r = \qquad \begin{array}{ccc} A & B & C \\ \hline a & b & c \\ c & b & d \end{array}$$

Selektion

- Sei $r \subseteq \mathsf{Tup}(X)$ eine Relation und α eine Selektionsbedingung zu X.
- Der Ausdruck $\sigma[\alpha]r$ heißt *Selektion* der Relation r bezüglich α. Es gilt:

$$\sigma[\alpha]r := \{\mu \in \mathsf{Tup}(X) \mid \mu \in r \wedge \mu \text{ erfllt } \alpha\}.$$

Beispiel Vereinigung und Differenz

$r =$

A	B	C
a	b	c
d	a	f
c	b	d

$s =$

A	B	C
b	g	a
d	a	f

$r \cup s =$

A	B	C
a	b	c
d	a	f
c	b	d
b	g	a

$r =$

A	B	C
a	b	c
d	a	f
c	b	d

$s =$

A	B	C
b	g	a
d	a	f

$r - s =$

A	B	C
a	b	c
c	b	d

Vereinigung und Differenz

- Seien X, Y Attributmengen, wobei $X = Y$ und seien weiter $r \subseteq \mathrm{Tup}(X), s \subseteq \mathrm{Tup}(Y)$ zwei entsprechende Relationen.
- $$r \cup s = \{\mu \in \mathrm{Tup}(X) \mid \mu \in r \vee \mu \in s\}.$$
  $$r - s = \{\mu \in \mathrm{Tup}(X) \mid \mu \in r, \text{ wobei } \mu \notin s\}.$$

Beispiel Verbund

$r =$

A	B	C
1	2	3
4	5	6
7	8	6

$s =$

C	D
3	1
6	2
4	5

$r \bowtie s =$

A	B	C	D
1	2	3	1
4	5	6	2
7	8	6	2

Verbund

- Seien X, Y Attributmengen; XY sei im Folgenden eine Kurzschreibweise für $X \cup Y$.
- Seien weiter $r \subseteq \mathrm{Tup}(X), s \subseteq \mathrm{Tup}(Y)$ zugehörige Relationen.
- Der *(natürliche) Verbund* $\bowtie$ von r und s ist dann definiert:

$$r \bowtie s := \{\mu \in \mathrm{Tup}(XY) \mid \mu[X] \in r \wedge \mu[Y] \in s\}.$$

Der *(natürliche) Verbund* $\bowtie$ von r und s ist definiert:

$$r \bowtie s := \{\mu \in \mathsf{Tup}(XY) \mid \mu[X] \in r \wedge \mu[Y] \in s\}.$$

Verbund fortgesetzt

Seien X_i, $1 \leq i \leq n$ Formate.

- Sei $X_1 \cap X_2 = \emptyset$

  $r_1 \times r_2 := r_1 \bowtie r_2$.
- $\bowtie_{i=1}^{n} r_i := \{\mu \in \mathsf{Tup}(\cup_{i=1}^{n} X_i) \mid \mu[X_i] \in r_i, 1 \leq i \leq n\}$.

Beispiel Umbenennung

$X = \{A, B, C\}$, $Y = \{D, E, C\}$ und $\delta = \{A \to D, B \to E, C \to C\}$.

$r =$

A	B	C
a	b	c
d	a	f
c	b	d

$\delta[X, Y]r =$

D	E	C
a	b	c
d	a	f
c	b	d

Umbenennung

- Seien $X = \{A_1, \ldots, A_k\}$, $Y = \{B_1, \ldots, B_k\}$ Formate.
- Sei δ eine umkehrbar eindeutige Abbildung von X nach Y, wobei $dom(A) = dom(\delta(A))$. Gilt $\delta(A) = B$, so schreiben wir $A \to B$.
- Sei $r \subseteq \text{Tup}(X)$ eine Relation zu X.
- Die Umbenennung $\delta[X, Y]$ bezüglich r ist wie folgt:

$$\delta[X, Y]r := \{\mu \in \text{Tup}(Y) \mid \quad \exists \mu' \in r, \text{so dass } \mu'(A_i) = \mu(\delta(A_i)), 1 \leq i \leq k\}$$

Basisoperatoren

- Selektion, Projektion, Vereinigung, Differenz, Verbund und Umbenennung sind die Basisoperatoren der Relationenalgebra.
- Die Anwendung dieser Operatoren auf Relationen liefert als Ergebnis wiederum eine Relation.
- Die zulässigen Ausdrücke der Relationenalgebra können ausgehend von den Basisoperatoren induktiv definiert werden.
- Wir können andere nützliche Operatoren definiern.

weitere Operatoren

Seien X_i, $1 \leq i \leq n$, Formate und seien $r_i \subseteq \mathrm{Tup}(X_i)$, $1 \leq i \leq n$, Relationen.

- *Durchschnitt*. Sei $X_1 = X_2$.

$$r_1 \cap r_2 := r_1 - (r_1 - r_2).$$

- *θ-Verbund*. Sei $X_1 \cap X_2 = \emptyset$ und sei α eine beliebige Selektionsbedingung über $X_1 \cup X_2$.

$$r \bowtie_\alpha s := \sigma[\alpha](r \bowtie s).$$

  Enthält α ausschließlich Gleichheitsvergleiche, dann redet man von einem *Equi-Verbund*.

Beispiel Division

$r_1 =$

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	d

$r_2 =$

C	D
c	d
e	f

$r_1 \div r_2 =$

A	B
a	b
e	d

Division

Seien X_1, X_2 Formate, $X_2 \subset X_1$, $Z = X_1 - X_2$ und weiter $r_2 \neq \emptyset$.

$$\begin{aligned} r_1 \div r_2 &:= \{\mu \in \mathrm{Tup}(Z) \mid \{\mu\} \times r_2 \subseteq r_1\} \\ &= \pi[Z]r_1 - \pi[Z](((\pi[Z]r_1) \times r_2) - r_1). \end{aligned}$$

Beispiel: Welche Studierenden belegen alle Kurse?

Kurs(<u>KursNr</u>, Institut, Name, Beschreibung)
Belegung(<u>MatrNr</u>, <u>KursNr</u>, Semester, Note)

$$\pi[\mathtt{MatrNr}](\pi[\mathtt{MatrNr,KursNr}]\mathtt{Belegung} \div \pi[\mathtt{KursNr}]\mathtt{Kurs})$$

Äquivalenz

Zwei Ausdrücke der Algebra Q, Q' heißen *äquivalent*, $Q \equiv Q'$, genau dann, wenn für jede Instanz $\mathcal{I}$ der Datenbank für die Antworten $\mathcal{I}(Q)$ und $\mathcal{I}(Q')$ der Anfragen Q und Q' gilt:

$$\mathcal{I}(Q) = \mathcal{I}(Q').$$

Beispiel: Welche Kurse belegt Lisa Lustig?

```
Student(MatrNr, SName, Adresse, Semester)
Kurs(KursNr, Institut, KName, Beschreibung)
Belegung(MatrNr, KursNr, Note)
```

- $\sigma[SName = LisaLustig](Student \bowtie (Kurs \bowtie Belegung))$
- $\sigma[SName = LisaLustig]Student \bowtie (Kurs \bowtie Belegung)$
- $Kurs \bowtie (\sigma[SName = LisaLustig]Student \bowtie Belegung))$

Äquivalenz

Zwei Ausdrücke der Algebra Q, Q' heißen *äquivalent*, $Q \equiv Q'$, genau dann, wenn für jede Instanz $\mathcal{I}$ der Datenbank für die Antworten $\mathcal{I}(Q)$ und $\mathcal{I}(Q')$ der Anfragen Q und Q' gilt:

$$\mathcal{I}(Q) = \mathcal{I}(Q').$$

Beispiele

Sei attr(α) die in einer Selektionsbedingung α verwendete Menge von Attributen und seien $R, S, T \ldots$ Relationsbezeichner mit Formaten X, Y, Z.

- $Z \subseteq Y \subseteq X \Longrightarrow \pi[Z](\pi[Y]R) \equiv \pi[Z]R.$
- $\mathrm{attr}(\alpha) \subseteq Y \subseteq X \Longrightarrow \pi[Y](\sigma[\alpha]R) \equiv \sigma[\alpha](\pi[Y]R).$
- $R \bowtie R \equiv R.$

  Betrachte die Definition des Verbundes $R_1 \bowtie R_2$. Da hier $R = R_1 = R_2$, mit X Format von R, folgt $R \bowtie R = \{\mu \in Tup(X) | \mu[X] \in \mathcal{I}(R)\}$ für eine beliebige Instanz $\mathcal{I}(R)$. Damit: $R \bowtie R = \{\mu \in Tup(X) | \mu \in \mathcal{I}(R)\} = \mathcal{I}(R)$.
- $X = Y \Longrightarrow R \cap S \equiv R \bowtie S.$
- $\mathrm{attr}(\alpha) \subseteq X, \mathrm{attr}(\alpha) \cap Y = \emptyset \Longrightarrow \sigma[\alpha](R \bowtie S) \equiv (\sigma[\alpha]R) \bowtie S.$

Kapitel 6: Formaler Datenbankentwurf

- Die Schwierigkeiten der konzeptuellen Modellierung sind zu einem großen Teil dadurch begründet, dass sich die relevanten Strukturen einer Miniwelt erst in Diskussionen mit den Anwendern oder durch Analyse von Dokumenten erfassen lassen.
- Mit der Transformation eines konzeptuellen Schemas in ein relationales Schema, dem logischen Entwurf, ändert sich diese Situation.
- Die Konzepte des relationalen Datenmodells werden dahingehend erweitert, dass sich die Güte eines logischen Entwurfs formal überprüfen läßt.
- Grundlage hierfür sind sogenannte *funktionale Abhängigkeiten*.

6.1 Funktionale Abhängigkeiten

6.1.1 Definition

- Sei ein Relationsschema gegeben durch sein Format V und seien $X, Y \subseteq V$.
- Sei $r \in \mathrm{Rel}(V)$. r erfüllt eine *funktionale Abhängigkeit (FA)* $X \to Y$, wenn für alle $\mu, \nu \in r$ gilt:

$$\mu[X] = \nu[X] \Rightarrow \mu[Y] = \nu[Y].$$

  Wir sagen auch die FA $X \to Y$ gilt in r.
- Sei $\mathcal{F}$ eine Menge funktionaler Abhängigkeiten über V und $X, Y \subseteq V$. Die Menge aller Relationen r in $\mathrm{Rel}(V)$, die alle funktionalen Abhängigkeiten in $\mathcal{F}$ erfüllen, bezeichnen wir mit $\mathrm{Sat}(V, \mathcal{F})$.

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

Welche FA werden durch r erfüllt?

FA	gilt in r
$A \to B$	1
$A \to C$	1
$B \to A$	0
$B \to C$	0
$C \to A$	1
$C \to B$	1
$AB \to C$	1
$AC \to B$	1
$BC \to A$	1
$A \to \emptyset$	1
$A \to A$	1
$AB \to A$	1
$ABC \to A$	1
$\vdots$	$\vdots$
$\emptyset \to A$	0
$\emptyset \to B$	1
$\emptyset \to C$	0
$\emptyset \to AB$	0
$\vdots$	$\vdots$

6.1.2 Membership-Test

- $\mathcal{F}$ *impliziert* die funktionale Abhängigkeit $X \to Y$, $\mathcal{F} \models X \to Y$, wenn jede Relation $r \in \mathrm{Sat}(V, \mathcal{F})$ auch $X \to Y$ erfüllt.
- Die Menge $\mathcal{F}^+ = \{X \to Y \mid \mathcal{F} \models X \to Y\}$ nennen wir die *Hülle* von $\mathcal{F}$.
- Der Test $X \to Y \in \mathcal{F}^+$ ist der *Membership-Test*.

Armstrong-Axiome

Sei $r \in \mathsf{Sat}(V, \mathcal{F})$.

(A1) Reflexivität: Wenn $Y \subseteq X \subseteq V$, dann erfüllt r die FA $X \to Y$.

(A2) Augmentation: Wenn $X \to Y \in \mathcal{F}, Z \subseteq V$, dann erfüllt r auch die FA $XZ \to YZ$.

(A3) Transitivität: Wenn $X \to Y, Y \to Z \in \mathcal{F}$, dann erfüllt r auch die FA $X \to Z$.

(A1) erlaubt die Herleitung funktionaler Abhängigkeiten, ohne Bezug auf $\mathcal{F}$: *triviale funktionale Abhängigkeiten*.

Korrektheit und Vollständigkeit

- Die Armstrong-Axiome sind *korrekt* in dem Sinn, dass die mit ihnen herleitbaren funktionalen Abhängigkeiten in der Tat Elemente der Hülle $\mathcal{F}^+$ sind.
- Die Armstrong-Axiome sind auch *vollständig*, d.h., jede funktionale Abhängigkeit in $\mathcal{F}^+$ kann auch mit ihnen hergeleitet werden.

Membership-Test Variante 1:

Starte mit $\mathcal{F}$ und wende solange die Regeln (A1)–(A3) an, bis entweder $X \to Y$ hergeleitet, oder $\mathcal{F}^+$ hergeleitet und $X \to Y \notin \mathcal{F}^+$.

Ein solcher Algorithmus ist im Allgemeinen mindestens exponentiell in der Anzahl der Attribute in V, da für (A2) alle Teilmengen von V betrachtet werden müssen.

weitere Axiome

Seien $X, Y, Z, W \subseteq V$ und $A \in V$.

(A4) Vereinigung: Wenn $X \to Y, X \to Z \in \mathcal{F}$, dann erfüllt r auch die FA $X \to YZ$.

(A5) Pseudotransitivität: Wenn $X \to Y, WY \to Z \in \mathcal{F}$, dann erfüllt r auch die FA $XW \to Z$.

(A6) Dekomposition: Wenn $X \to Y \in \mathcal{F}, Z \subseteq Y$, dann erfüllt r auch die FA $X \to Z$.

(A7) Reflexivität: Wenn $X \subseteq V$, dann erfüllt r auch die FA $X \to X$.

(A8) Akkumulation: Wenn $X \to YZ, Z \to AW \in \mathcal{F}$, dann erfüllt r auch die FA $X \to YZA$.

Die Axiomensysteme {(A1), (A2), (A3)} und {(A6), (A7), (A8)} sind zueinander äquivalent.

Hierzu müssen wir zeigen, dass jedes Axiom der einen Menge durch die Axiome der anderen Menge simuliert werden kann.

Beispiel: (A6) kann durch (A1) und (A3) simuliert werden.

Seien $X, Y, Z \subseteq Y$.
Zu zeigen: $X \to Y, Z \subseteq Y \Rightarrow X \to Z$.

$$Z \subseteq Y \quad \stackrel{A1}{\Rightarrow} \quad Y \to Z$$

$$X \to Y, Y \to Z \quad \stackrel{A3}{\Rightarrow} \quad X \to Z$$

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

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

Membership-Test Variante 2:

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

XPlus-Algorithmus

```
XPlus(X, Y, F) boolean {
  result := X;
  WHILE (changes to result) DO
    FOR each X' → Y' ∈ F DO
       IF (X' ⊆ result) THEN result := result ∪ Y';
    end.
  IF (Y ⊆ 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

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

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

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

(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 $A \in V$ muss nur einmal betrachtet werden; 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).