Inverse im Forschungsdatenmanagement

Eine Kombination aus Provenance Management, Schema- und Daten-Evolution

Tanja Auge
Lehrstuhl für Datenbank- und Informationssysteme
Institut für Informatik
Universität Rostock
tanja.auge@uni-rostock.de

Andreas Heuer
Lehrstuhl für Datenbank- und Informationssysteme
Institut für Informatik
Universität Rostock
heuer@informatik.uni-rostock.de

ABSTRACT

Die Sammlung, Erfassung, Speicherung, Verfolgung und Archivierung wissenschaftlicher Daten ist Aufgabe des Forschungsdatenmanagements. Neben der Auswertungsanfrage und dem -ergebnis selbst soll auch der verwendete Ausschnitt der Originaldatenbank archiviert werden. So können zur Absicherung replizierbarer Forschung die Auswertungsanfragen zu einem späteren Zeitpunkt erneut gestellt werden, um das Ergebnis zu reproduzieren.

Falls sich die Daten oder das Schema der Forschungsdatenbank häufig verändern, müsste man nun nach jeder auf dem Datenbestand vorgenommenen Auswertung die Originaldatenbank „einfrieren“ und sichern. Um dies zu vermeiden und somit massiv replizierte Datenbestände zu erzeugen, wollen wir mit Hilfe von Provenance-Management-Techniken die minimale Teildatenbank berechnen, die gesichert werden muss, um das Anfrageergebnis wieder erzeugen zu können. Dazu wollen wir Techniken der ***why***- und ***how***-Provenance kombinieren mit der Theorie der Schema-Abbildungen für Datenintegration und Datenaustausch, hier insbesondere der inversen Schemaabbildungen von Fagin.

In diesem Beitrag stellen wir das gesamte Forschungsprojekt im Überblick vor und konzentrieren uns dann auf die ersten konkreten Teilergebnisse: die Klassifizierung von inversen Anfragen bei der Berechnung der ***why***-Provenance. Dabei werden wir die Fälle unterscheiden, ob wir die Inverse nur mit Hilfe von Ergebnis und Auswertungsanfrage berechnen können, ob die Inverse zumindest einen Datenbankausschnitt berechnet, der homomorph auf die Originaldatenbank abgebildet werden kann, oder ob für die Berechnung der Inversen zusätzliche Provenance-Informationen (Polynome der ***how***-Provenance) gespeichert werden müssen.

Categories and Subject Descriptors

Information Systems [**Data Management Systems**]: Data Provenance; Information Systems [**Information Integration**]: Data Exchange

Keywords

CHASE-Algorithmus, Data Provenance, Schema-Evolution, Daten-Evolution, CHASE-Inverse, Ergebnisäquivalenz

30^{th} GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 22.05.2018 - 25.05.2018, Wuppertal, Germany.

1. PROBLEMSTELLUNG

Unter dem Begriff *Provenance Management* versteht man die Rückverfolgbarkeit eines Anfrageergebnisses bis zu den relevanten Originaldaten [4]. Dies beinhaltet sowohl die Originaldaten an sich (***where-***Provenance) als auch den zurückgelegten Weg (***why-*** und ***how-***Provenance).

Anwendung findet diese Fragestellung etwa im *Forschungsdatenmanagement*. Das Ziel besteht hierbei in der Verfolgung und Archivierung von Daten, die bei wissenschaftlichen Projekten, Experimenten oder Beobachtungen erfasst worden sind. Der Weg von der Datenerhebung bis zur Publikation soll so nachvollziehbar, rekonstruierbar und replizierbar gehalten werden. Da die Forschungsdatenbank sich ständig ändert und somit eine bitemporale Datenbank [18] darstellt, muss die Evolution von Daten und Schemata mit dem Provenance Management zusammenspielen. Leider werden bisher die Data-Provenance-Forschungen immer auf einer festen Datenbank durchgeführt. Zwei Forschungsziele des Projektes bestehen daher in der Berechnung einer minimalen Teildatenbank der Original-Forschungsdatenbank sowie der Vereinheitlichung von Provenance und Evolution.

Berechnung einer minimalen Teildatenbank: Die berechnete minimale Teildatenbank soll die Ergebnisse der Auswertungsanfrage unter verschiedenen Randbedingungen rekonstruieren können, etwa:

- Die Tupelanzahl der Originalrelation bleibt erhalten.
- Die Teildatenbank kann homomorph auf das Original abgebildet werden.

Konkret stellt sich die Frage, welche zusätzlichen Informationen bei archiviertem Ergebnis und gemerkter Auswertungsanfrage für die Rekonstruktion der minimalen Teildatenbank benötigt werden. Reicht es, eine minimale Zeugenmenge aufzuheben (***why***-Provenance, [6]) und/oder die zugehörigen Provenance-Polynome zu berechnen (***how***-Provenance, [14])? Oder ist es notwendig, ganze Tupel bzw. Datenbankausschnitte direkt zu speichern?

Die Berechnung einer inversen Anfrage Q^{-1}, mit deren Hilfe die gesuchte minimale Teildatenbank bestimmt werden soll, ist abhängig vom Typ der Originalanfrage Q sowie allen zusätzlich gemerkten Informationen. So kann für die Projektion etwa eine *ergebnisäquivalente CHASE-Inverse* [3], für die

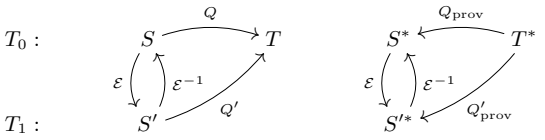


Abbildung 1: Provenance und Evolution mit

- minimalen Teildatenbanken $S^* \subseteq S$, $T^* \subseteq T$, $S'^* \subseteq S'$
- Q: Auswertungsanfrage
- $\mathcal{E}$: Schemaevolution
- Q_{prov}: Provenanceanfrage

Projektion ohne Duplikateliminierung eine *relaxte CHASE-Inverse* und für die Kopie eine *exakte CHASE-Inverse* [12] angegeben werden (siehe Tabelle 4). Die in den Randbedingungen geforderte Homomorphie ist eine sehr starke Eigenschaft, welche im Verlauf weiterer Untersuchungen gezielt abgeschwächt werden soll. Ein erster Ansatz ist die Definition ergebnisäquivalenter CHASE-Inversen (siehe Abschnitt 3.3). Diese vernachlässigen die Homomorphie gänzlich, was wiederum zu einem starken Informationsverlust führen kann. So können ganze (vielleicht für andere Anfragen notwendige) Tupel verloren gehen, welche bei der Verwendung einer relaxten CHASE-Inversen zumindest als Nulltupel noch erhalten blieben. Andere Ansätze führen beispielsweise Konzepthierarchien oder eine Vergröberung der Datenbankbestände ein, wie es etwa für intensionale Provenance-Antworten notwendig ist [22].

Vereinheitlichung von Provenance und Evolution: Bisherige Provenance-Anfragen Q_{prov} (***where***-, ***why***- und ***how***-Provenance) erfolgen stets auf einer gegebenen festen Datenbank S sowie einer Auswertungs-Anfrage Q. Die Kombination von Provenance mit Schema- und Daten-Evolution soll die Auswertung von Provenance-Anfragen bei sich ändernden Daten und Schemata ermöglichen (siehe Abbildung 1). Mit Hilfe der Inversenbildung $\mathcal{E}^{-1}$ kann wiederum eine zu archivierende minimale Teildatenbank S'^* der neuen Originaldatenbank S' bestimmt oder (falls möglich) aus einem alten Datenbestand zum alten Schema rückberechnet werden. Das mittels CHASE berechnete Auswertungsergebnis (siehe Abschnitt 3.1) soll daher in einem zweiten Schritt, dem BACKCHASE, um eine inverse Provenance-Anfrage Q'_{prov} erweitert werden.

Formal: Nach der Schemaevolution $\mathcal{E} : S \to S'$ kann die Anfrage Q' als Komposition der ursprünglichen Anfrage Q sowie der inversen Evolution $\mathcal{E}^{-1}$ direkt berechnet werden:

$$Q'(S') = (Q \circ \mathcal{E}^{-1})(S) = \mathcal{E}^{-1}(Q(S)).$$

Die neue Provenanceanfrage Q'_{prov} ergibt sich analog als

$$Q'_{\text{prov}}(T^*) = (\mathcal{E} \circ Q_{\text{prov}})(T^*).$$

Es genügt daher, sich eine der beiden minimalen Teildatenbanken S^* oder S'^* zu merken. Die jeweils andere lässt sich mit Hilfe der Inversen leicht berechnen.

Die Universität Rostock vereint ihre Forschungen zum Thema Provenance Management im Projekt **METIS in PArADISE** [17]. Konkret beschäftigt sich dieses Projekt mit Provenance Management bei der Auswertung von Sensordatenmengen für die Entwicklung von Assistenzsystemen.

METIS: Das Langzeitprojekt METIS (Management, Evolution, Transformation und Integration von Schemata) vereint die Verfahren zur Integration, Transformation und Evolution von Daten. Auch die Frage der Datenherkunft wird in diesem Projekt untersucht.

PArADISE: In der Projektgruppe PArADISE (Privacy AwaRe Assistive Distributed Information System Environment) werden Techniken zur Auswertung von großen Mengen von Sensordaten entwickelt.

Eine Anwendung von **METIS in PArADISE** im Bereich des Forschungsdatenmanagements wird zusammen mit dem Leibniz-Institut für Ostseeforschung in Warnemünde (IOW) entwickelt (siehe [5]). Die Notwendigkeit zur Integration von Provenance- und Evolutions-Fragestellungen wurde gerade in dieser Kooperation geboren.

Im Rest dieses Beitrages werden wir die Formalisierung der Auswertungsanfragen und ihrer Inversen mit Hilfe des CHASE einführen sowie für einfache Auswertungsanfragen verschiedene Typen von Inversen angeben. Die Formalisierung der Techniken über den CHASE soll uns gerade ermöglichen, die Ergebnisse der Schema-Evolutions-Theorie, der Theorie des Datenaustausches und der Data Provenance zu vereinheitlichen.

2. STAND DER FORSCHUNG

Die Idee unserer Arbeit basiert auf der Kombination dreier verschiedenen Grundtechniken.

Provenance: Die Grundlage für unsere Provenance-Analysen bildet [16] mit einem allgemeinen Überblick über die Begriffsbildungen sowie die verschiedenen Klassifizierungen von Provenance-Anfragen (***why***, ***where***, ***how*** und ***why not***). Die Konstruktion von (minimalen) Zeugenbasen und Provenance-Polynomen zur Beantwortung der ***why***- bzw. ***how***-Provenance basieren auf den Techniken aus [6] und [1, 14, 15]. Die Übertragung der Problematik auf die Spieltheorie wird in [19] zur Analyse der ***why not***-Provenance benutzt.

CHASE: Leider werden Provenance-Anfragen derzeit nur auf festen Datenbanken benutzt. Um die Technik mit der Evolution von Schemata und Daten kombinieren zu können, wollen wir die Auswertungs-Anfragen Q sowie die Provenance-Anfragen Q_{prov} mit Hilfe von s-t tgds und egds (siehe unten) darstellen und mit Hilfe des CHASE konzeptuell umsetzen. Seine ursprüngliche Anwendung hat der CHASE im Datenbankentwurf [20], der auf den Tableaudefinitionen in [1] basiert. Der CHASE wurde dann später vor allem für die Transformation von Datenbanken im Bereich der Datenintegration, des Datenaustausches und der Schema-Evolution benutzt. In diesen Fällen dient der CHASE dann zur Darstellung von Schemabbildungen.

Schemaabbildungen und inverse Abbildungen. Fagin et al. beschäftigen sich in ihren Arbeiten [9, 12] ausführlich mit der Definition des CHASE-Algorithmus für s-t tgds und egds sowie der Konstruktion von CHASE-inversen Schemaabbildungen [10, 11]. Eine Implementierung des CHASE für diesen Anwendungsfall liefert etwa die Software Llunatic [13]. In [8] wurden diese Techniken auch für die Schema-Evolution verwendet. Die Nutzung inverser Schemaabbildungen für das Provenance Management, speziell für Provenance-Anfragen, ist bisher noch nicht vorgenommen worden.

3. DIE TECHNIKEN

Die Bearbeitung der beiden oben vorgestellten Forschungsziele soll durch die Kombination der gerade vorgestellten Techniken erfolgen, (1) dem CHASE-Algorithmus zur Auswertung von Anfragen, (2) der Ermittlung von ***why***- und

ID	Titel	Matrikelnummer	Studiengang	
1234	Big Data	4711	Informatik	t_1
2467	Provenance	4711	Informatik	t_2

Tabelle 1: APS-Relation mit zugehörigen Tupelidentifikatoren t_1 und t_2

how-Provenance und (3) der Berechnung von CHASE-inversen Abbildungen zur Rekonstruktion einer minimalen Teilmenge der Originaldatenbank.

3.1 Der CHASE

Der CHASE ist ein in der Datenbanktheorie universell einsetzbares Werkzeug. Es findet Anwendung etwa bei der Darstellung von Abhängigkeiten innerhalb einer Datenbank oder eines relationenalgebraischen Ausdrucks, der Implikation von Abhängigkeiten, dem Äquivalenznachweis von Datenbankschemata unter gegebenen Abhängigkeiten sowie der Behandlung (Ersetzung oder Säuberung) von Nullwerten in Datenbanken. Die Idee dieses Algorithmus kann dabei wie folgt zusammengefasst werden: Für ein Objekt $\bigcirc$ und eine Menge von Abhängigkeiten $\star$ (beispielsweise eine Menge von FDs und JDs) arbeitet der CHASE die Abhängigkeiten $\star$ in das Objekt $\bigcirc$ ein, sodass $\star$ implizit in $\bigcirc$ enthalten ist, und somit $\bigcirc$ die Abhängigkeiten $*$ nicht verletzt. Bildhaft ergibt sich:

$$\text{chase}_\star(\bigcirc) = \circledast.$$

Seine ursprüngliche Anwendung hat der CHASE im Datenbankentwurf [20], der auf den Tableaudefinitionen in [1] basiert. Der CHASE kann aber nicht nur mit FDs und JDs ($\star$) auf Tableaus ($\bigcirc$), sondern auch mit tgds/s-t tgds und egds ($\star$) auf Datenbanken ($\bigcirc$) definiert werden. Dies beschreiben unter Anderem Fagin et al. in [12, 9].

Eine *tuple-generating dependency (tgd)* ist eine Sequenz der Form

$$\forall x : (\phi(x) \rightarrow \exists y : \psi(x, y))$$

mit Konjunktionen $\phi(x)$ und $\psi(x, y)$ von Atomen aus x bzw. x und y. Ist $\psi(x)$ eine Konjunktion von Atomen über einem Quellschema S und $\phi(x, y)$ eine Konjunktion von Atomen über einem Zielschema T, so heißt eine solche Sequenz auch *source-to-target tuple-generating dependency (s-t tgd)*. Eine *equality-generating dependency (egd)* ist für zwei Variablen x_1, x_2 aus x definiert durch

$$\forall x : (\phi(x) \rightarrow (x_1 = x_2)).$$

Der *CHASE* einer Instanz I über einer Abhängigkeitsmenge Σ ist schließlich eine (endliche oder unendliche) Sequenz von *tgd-* und *egd-Regeln*, welche neue Tupel erzeugen oder (markierte) Nullwerte durch Konstanten bzw. (markierte) Nullwerte mit kleinerem Index ersetzen kann.

Zur Veranschaulichung des CHASE sei exemplarisch die Datenbank einer Universität gegeben. Diese enthält unter Anderem eine Relation mit den Namen, Studiengängen und Matrikelnummern ihrer Studenten sowie eine weitere Relation, welche alle abgeschlossenen Abschlussarbeiten listet. Die zugehörigen Schemata sind definiert

STUD(Matrikelnummer, Name, Studiengang)

und

AA(ID, Titel, Vertiefung, Matrikelnummer, Note).

Die Relation APS mit dem Schema

APS(ID, Titel, Matrikelnummer, Studiengang),

welche die Abschlussarbeiten pro Studiengang zusammenfasst, soll nun in die Relationen AA und STUD migriert werden. Hierzu seien folgende Abhängigkeiten gegeben (eine s-t tgd, eine tgd sowie eine egd):

$$\begin{aligned}\Sigma_{\text{ST}} &= \{\delta_1 : \text{APS}(i,t,m,s) \\ &\qquad \rightarrow \exists N, V, G : \text{STUD}(m,N,s) \ \wedge \ \text{AA}(i,t,V,m,G)\} \\ \Sigma_{\text{T}} &= \{\delta_2 : \text{AA}(i,t,v,m,g) \rightarrow \exists\, N, S : \text{STUD}(m,N,S), \\ &\qquad \delta_3 : \text{AA}(m,n_1,s) \wedge \ \text{AA}(m,n_2,s) \rightarrow n_1 = n_2\}\end{aligned}$$

Der CHASE liefert nun, angewandt auf den Datenbestand aus Tabelle 1, nach Integration der s-t tgd δ_1 die Tupelmenge

$$\begin{aligned}I' = \ &\{\text{STUD}(4711,\, N_1,\, \text{Informatik}), \\ &\text{AA}(1234,\, \text{Big Data},\, V_1,\, 4711,\, G_1), \\ &\text{STUD}(4711,\, N_2,\, \text{Informatik}), \\ &\text{AA}(2467,\, \text{Provenance},\, V_2,\, 4711,\, G_2)\}\end{aligned}$$

mit den (markierten) Nullwerten N_1, N_2, V_1, V_2, G_1 und G_2. Bevor δ_2 sowie δ_3 eingearbeitet werden, muss geprüft werden, ob I' die Abhängigkeiten aus Σ_{T} bereits erfüllt, d.h. $I' \models \Sigma_{\text{T}}$. In diesem Fall wäre ein weiterer *CHASE-Schritt*, also die Anwendung einer tgd- oder egd-Regel, nicht mehr notwendig und der CHASE selbst erfolgreich. Die tgd δ_2 ist bereits erfüllt, sodass die Eliminierung überflüssiger Nullwerte — das sogenannte *cleaning* — mittels δ_3 bereits die Ergebnisinstanz

$$\begin{aligned}I'' = \ &\{\text{STUD}(4711,\, N_1,\, \text{Informatik}), \\ &\text{AA}(1234,\, \text{Big Data},\, V_1,\, 4711,\, G_1), \\ &\text{AA}(2467,\, \text{Provenance},\, V_2,\, 4711,\, G_2)\}\end{aligned}$$

liefert. Implementiert ist diese Art des CHASE etwa in der Software **Llunatic** [13].

3.2 Data Provenance

Typischerweise werden vier Provenance-Anfragen und vier Provenance-Antworten (siehe Tabelle 2) unterschieden [16]. Die ersten drei Anfragen können dabei bzgl. ihres Informationsgehaltes ($\preceq$) geordnet werden:

$$\textbf{\textit{where}} \preceq \textbf{\textit{why}} \preceq \textbf{\textit{how}}.$$

Mit anderen Worten, aus dem Ergebnis der ***how***-Provenance können die ***why***- sowie die ***where***-Provenance abgeleitet werden. Gleiches gilt für die ***why***- und ***where***-Provenance (siehe Abbildung 2). Der formale Beweis für die Reduktion der ***why***- auf die ***how***-Provenance kann in [7] nachgelesen werden. Die ***why not***-Anfrage kann in diese „Informationskette“ nicht eingeordnet werden, da sie, anders als die ***why***-, ***where***- oder ***how***-Anfrage keine extensionale, sondern eine anfrage- oder modifikationsbasierte Antwort liefert.

Die ***how***-Provenance einer Anfrage Q kann als Provenance-Polynom [14, 2, 15], die ***why***-Provenance in Form von (minimalen) Zeugenbasen [6] und die ***where***-Provenance als Liste

where-Provenance (Tabellenname)	***why***-Provenance (Zeugenbasis)	***how***-Provenance (Polynom)
APS	$\{\{t_1\}, \{t_2\}, \{t_1, t_2\}\}$	$(t_1 \cdot (t_1 + t_2)) + (t_2 \cdot (t_1 + t_2))$

Abbildung 2: Reduktion von ***why***, ***where*** und ***how***

Anfrage-Typ	Fragestellung
where	Woher kommen die Daten?
why	Warum dieses Ergebnis?
how	Wie kommt das Ergebnis zustande?
why not	Warum fehlt ein bestimmtes Element?

Antwort-Typ	Ergebnis
extensional	Tupel aus den Originaldaten
intensional	Beschreibung der Daten
anfragebasiert	Selektionsprädikate
modifikationsbasiert	Vorschlag zur minimalen Änderung der Auswertungsanfrage

Tabelle 2: Provenance-Anfragen und -Antworten

von Tupelidentifikatoren oder Tabellennamen der am Ergebnis beteiligten Tupel angegeben werden. Die Anfrage

$$Q = \pi_{\mathrm{m}}(\pi_{\mathrm{i,s}}(APS) \bowtie \pi_{\mathrm{m,s}}(APS)),$$

welche auf einem Beispiel aus [15] basiert, liefert die Matrikelnummer 4711. Die zugehörigen Provenance-Informationen können dabei wie folgt angegeben werden:

- ***how***-Provenance:
  - Provenance-Polynom: $(t_1 \cdot (t_1 + t_2)) + (t_2 \cdot (t_1 + t_2))$ mit $+$ für Alternativen und $\cdot$ für Kombinationen
- ***why***-Provenance:
  - Zeugenbasis: $\{\{t_1\}, \{t_2\}, \{t_1, t_2\}\}$
  - minimale Zeugenbasis: $\{t_1\}$ oder $\{t_2\}$
- ***where***-Provenance:
  - Name der am Ergebnis beteiligten Tabelle: APS
  - Menge der am Ergebnis beteiligten Tupel: $\{t_1, t_2\}$

Für die Angabe einer extensionalen Provenance-Antwort genügt somit die Berechnung der ***how***-Provenance. Die Berechnung und Speicherung dieser Polynome erfolgt etwa über die schrittweise Verfolgung der algebraischen Grundoperationen der Anfrage oder über graphentheoretische Ansätze. Im Bereich der Spieltheorie existieren für die Beantwortung der ***why***-, ***why not***- oder ***how***-Provenance beispielsweise die sogenannten *Provenance Games* [19].

3.3 CHASE-Inverse

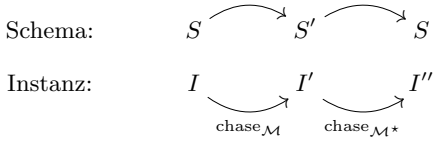
Exakte, klassische und relaxte CHASE-Inverse sind spezielle Formen von inversen Abbildungen [12]. Erstmalig formuliert wurden inverse Schemaabbildungen von Fagin [10]. Er definiert die Inverse einer Schemaabbildung $\mathcal{M} : S \to S'$ als eine Schemaabbildung $\mathcal{M}^* : S' \to S$, deren Komposition $\mathcal{M} \circ \mathcal{M}^*$ der Identitätsabbildung entspricht.

Inverse Abbildungen finden etwa bei der Schemaevolution Anwendung. Während eine *exakte CHASE-Inverse* stets die Originaldatenbank rekonstruiert, liefert die *klassische CHASE-Inverse* nur ein zur Originaldatenbank äquivalentes Ergebnis. Kann eine Inverse nicht angegeben bzw. definiert werden, wird aber dennoch eine Art inverse Abbildung benötigt, kann auf die sogenannte Quasi-Inverse zurückgegriffen werden [11]. Sie liefert nicht mehr die Quellinstanz selbst, dafür aber eine hierzu äquivalente Instanz oder zumindest eine äquivalente Teilinstanz.

Für eine Anfrage Q und eine Quellinstanz I liefert der CHASE eine Ergebnisinstanz $I' = \mathrm{chase}_{\mathcal{M}}(I)$, wenn gilt:

- Q wird als Schemaabbildung $\mathcal{M} = (S, S', \Sigma)$ mit Quell- und Zielschema S und S' sowie einer Menge von Abhängigkeiten Σ aufgefasst;
- I ist Quellinstanz in S.
- I' ist Zielinstanz in S'.

Die *Urinstanz* $I'' = \mathrm{chase}_{\mathcal{M}^*}(I') = \mathrm{chase}_{\mathcal{M}^*}(\mathrm{chase}_{\mathcal{M}}(I))$ ist somit das Ergebnis einer Rückanfrage Q' auf die Ergebnisinstanz I'. Dabei enthält I'' ganze oder auf bestimmte Attribute des Quellschemas eingeschränkte (und mit (markierten) Nullwerten aufgefüllte) Tupel aus I.



Das CHASE&BACKCHASE-Verfahren zur Bestimmung einer CHASE-inversen Schemaabbildung $\mathcal{M}^* = (S', S, \Sigma')$ zu $\mathcal{M} = (S, S', \Sigma)$ lässt sich daher wie folgt beschreiben:

- CHASE: Berechne den CHASE von I bzgl. $\mathcal{M}$ als Sequenz von s-t tgd- und egd-Regeln.
- BACKCHASE: Berechne den CHASE von I' bzgl. $\mathcal{M}^*$ als Sequenz von s-t tgd- und egd-Regeln.

Enthält die Urinstanz I'' Tupel mit (markierten) Nullwerten, deren restliche Attributwerte mit den Attributwerten eines Tupels der Quellinstanz I übereinstimmen, beispielsweise $(1, 3, 5) \in I$ und $(1, 3, n_1) \in I''$, so heißt I'' *Ausschnitt* der Instanz I, kurz $I'' \preceq I$. Mit anderen Worten, es existiert ein Homomorphismus h, welcher die Tupel aus I'' auf die Tupel aus I abbildet. Für das obige Beispiel gilt dann $1 \mapsto 1, 3 \mapsto 3$ und $n_1 \mapsto 5$. Enthält I'' keine (markierten) Nullwerten und sind alle Tupel aus I'' auch Tupel in I, schreibt man $I'' \subseteq I$.

Die *relaxte CHASE-Inverse* ist eine Abschwächung der klassischen CHASE-Inversen. Sie fordert keine Äquivalenzbeziehung zwischen Quell- und Urinstanz I'', dafür aber Ergebnisäquivalenz sowie die Existenz eines Homomorphismus von der Urinstanz I'' in die Quellinstanz I. Eine Schemaabbildung $\mathcal{M} = (S, S', \Sigma)$ heißt *ergebnisäquivalent* bzgl. $\mathcal{M}$ (kurz: $I \leftrightarrow_{\mathcal{M}} J$), wenn für zwei Instanzen I, J über S gilt:

$$\mathrm{chase}_{\mathcal{M}}(I) \leftrightarrow \mathrm{chase}_{\mathcal{M}}(J).$$

Die *ergebnisäquivalente CHASE-Inverse* fordert nur noch Ergebnisäquivalenz und ist somit die schwächste CHASE-Inverse. Insgesamt ergibt sich die Reduktion:

$$\begin{aligned} \text{ergebnisäquivalent} &\preceq \text{relaxt} \\ &\preceq \text{klassisch} \\ &\preceq \text{exakt.} \end{aligned}$$

Diese Reduktion bildet die hinreichende Bedingung für die Existenz einer CHASE-Inversen. Die notwendige Bedingung folgt aus der jeweiligen Definition der CHASE-Inversen. Tabelle 3 fasst diese Bedingungen zusammen.

Eine allgemeine Untersuchung der wichtigsten Basis-Operationen auf die Existenz CHASE-inverser Abbildungen bildet den Kern der Masterarbeit „Umsetzung von Provenance-Anfragen in Big-Data-Analytics-Umgebungen“ [3]. So können für die Operationen der Relationenalgebra bis auf wenige Ausnahmen (Selektion auf Ungleichheit, Differenzbildung)

CHASE-Inverse	Hinreichende Bedingung	Notwendige Bedingung
Exakt	-	$I'' = I$
Klassisch	Exakte CHASE-Inverse	$I'' \leftrightarrow I$
Relaxt	Klassische CHASE-Inverse	$I'' \preceq I$
Ergebnisäquivalent	Relaxte CHASE-Inverse	$I'' \leftrightarrow_{\mathcal{M}} I$

Tabelle 3: Hinreichende und notwendige Bedingungen für die Existenz von CHASE-Inversen

exakte, relaxte oder ergebnisäquivalente CHASE-Inverse angegeben werden (siehe Tabelle 4, Spalte 2). Die Existenz einer exakten CHASE-Inversen (=) kann dabei nur für einige wenige relationale Operationen wie das Kopieren oder den natürliche Verbund ohne Duplikate nachgewiesen werden. Während für die Aggregatfunktionen SUM und AVG keine CHASE-Inverse gefunden werden kann (xxx), sind die meisten Operationen ergebnisäquivalent ($\leftrightarrow$) oder relaxt CHASE-invers ($\sim$).

Die Hinzunahme zusätzlicher Provenance-Informationen ermöglicht die Angabe stärkerer CHASE-Inversen (siehe Tabelle 4, Spalte 4). Die hierfür benötigten Provenance-Polynome und (minimalen) Zeugenbasen entsprechen den Definitionen aus Abschnitt 3.2. So kann im Falle der Projektion mit Duplikaten durch die Verwendung von Data Provenance statt einer ergebnisäquivalenten CHASE-Inversen eine relaxte und für die Vereinigung sogar eine exakte CHASE-Inverse konstruiert werden. Für andere Operationen wie etwa die Projektion ohne Duplikate, die Selektion auf $<, \leq, =, \geq, >$ oder die Maximum- bzw. Minimumbildung können trotz Provenance-Untersuchungen keine Verbesserungen gefunden werden. Die Kopieroperation, die Umbenennung, der natürliche Verbund ohne Duplikate sowie die arithmetischen Operationen $+, -, \cdot$ und : liefern zudem bereits ohne Provenance-Aspekte exakte CHASE-Inverse. Für die Selektion auf Ungleichheit sowie die Differenzbildung können trotz Provenance keine CHASE-Inversen gefunden werden. Anders verhält es sich bei den Aggregatfunktionen SUM und AVG. Hier kann nur mit Hilfe der Provenance-Polynome überhaupt eine CHASE-inverse Abbildung angegeben werden.

Eine konkrete Angabe der inversen Abbildung $\mathcal{M}^*$ ist für einige der Grundoperationen ebenfalls möglich. Die ergebnisäquivalente CHASE-Inverse entspricht stets der Identitätsabbildung (Selektion, Mengenoperationen) oder im Falle von Relationenschemaänderungen der Erweiterung um (markierte) Nullwerten (null values, NV) etwa bei den Operationen Projektion, natürlicher Verbund sowie den Aggregatfunktionen MIN und MAX. Die relaxte CHASE-Inverse kann neben den (markierten) Nullwerten sogar ganze Nulltupel (Aggregatfunktion COUNT) erzeugen. Der natürliche Verbund, die arithmetischen Operationen $+$, $-$, $\cdot$ und : sowie die Umbenennung lassen sich durch die Projektion, die arithmetischen Operationen $-$, $+$, : und $\cdot$ sowie die Rückbenennung invertieren. Dargestellt sind diese Erkenntnisse in der dritten und fünften Spalte der Tabelle 4.

Für die Komposition $\mathcal{M} = \mathcal{M}_1 \circ ... \circ \mathcal{M}_n$ der obigen Operationen ergibt sich die inverse Funktion

$$\mathcal{M}^* = (\mathcal{M}_1 \circ ... \circ \mathcal{M}_n)^{-1} = \mathcal{M}_n^* \circ ... \circ \mathcal{M}_2^* \circ \mathcal{M}_1^*$$

als Komposition der inversen Teiloperationen $\mathcal{M}_1^*, ..., \mathcal{M}_n^*$. Der Typ der Inversen $\mathcal{M}^*$ entspricht dabei dem Typ der schwächsten Teilinversen $\mathcal{M}_i^*$ $(i = 1, ..., n)$. Für die Anfrage

$$Q = \pi_{\mathrm{m}}(\pi_{\mathrm{i,s}}(\mathrm{APS}) \bowtie \pi_{\mathrm{m,s}}(\mathrm{APS}))$$

Operation $\mathcal{M}$	ohne Provenance		mit Provenance	
	Inversen-Typ	Inverse $\mathcal{M}^\star$	Inversen-Typ	Inverse $\mathcal{M}^\star$
$r(\mathcal{R})$	$=$	Identität	$=$	Identität
$\beta_{A_j \leftarrow A_i}(r(\mathcal{R}))$	$=$	$\beta_{A_i \leftarrow A_j}$	$=$	$\beta_{A_i \leftarrow A_j}$
$\pi_{A_i}(r(\mathcal{R}))$	$\leftrightarrow$	Erweiterung um NV in A_i	$\leftrightarrow$	Erweiterung um NV in A_i
	$\sim$	Erweiterung um NV in A_i	$\leftrightarrow$	Erweiterung um NV in A_i + Rekonstruktion verlorener Tupel
$r_1(\mathcal{R}_1) \bowtie r_2(\mathcal{R}_2)$	$=$ $\sim$	Projektion auf Attribute aus $\mathcal{R}_i$ Projektion auf Attribute aus $\mathcal{R}_i$	$=$ $\sim$	Projektion auf Attribute aus $\mathcal{R}_i$ Projektion auf Attribute aus $\mathcal{R}_i$
$\sigma_{A_i \theta c}(r(\mathcal{R}))$	$\sim$ $\sim$	Identität	$\sim$ $\sim$	Identität
$\sigma_{A_i \theta A_j}(r(\mathcal{R}))$ mit $\theta \in \{<, \leq, =, \geq, >\}$		Identität		Identität
$\sigma_{A_i \neq c}(r(\mathcal{R}))$	xxx	xxx	xxx	xxx
$\sigma_{A_i \neq A_j}(r(\mathcal{R}))$	xxx	xxx	xxx	xxx
$r_1(\mathcal{R}_1) \cup r_2(\mathcal{R}_2)$	$\sim$ $\sim$	Identität	$=$ $\sim$	Selektion
$r_1(\mathcal{R}_1) \cap r_2(\mathcal{R}_2)$		Identität		Identität
$r_1(\mathcal{R}_1) - r_2(\mathcal{R}_2)$	xxx	xxx	xxx	xxx
$\mathbf{MAX}_{A_i}(r(\mathcal{R}))$ / $\mathbf{MIN}(r(\mathcal{R}))$	$\sim$	Erweiterung um NV in A_j, $i \neq j$	$\sim$	Erweiterung um NV in A_j, $i \neq j$
$\mathbf{COUNT}_{A_i}(r(\mathcal{R}))$	$\leftrightarrow$	Erzeugen von Nulltupeln	$\leftrightarrow$	Erzeugen von Nulltupeln
$\mathbf{SUM}_{A_i}(r(\mathcal{R}))$	xxx	xxx	$=$	Rekonstruktion verlorener Attributwerte in A_i
$\mathbf{AVG}_{A_i}(r(\mathcal{R}))$	xxx	xxx	$=$	Rekonstruktion verlorener Attributwerte in A_i
$r(\mathcal{R})$ θ α mit $\theta \in \{+, -, \cdot, /\}$	$=$	$-$ / $+$ / : / $\cdot$	$=$	$-$ / $+$ / : / $\cdot$
$r(\mathcal{R})$ **MOD** α	$\sim$	Identität	$\sim$	Identität

Tabelle 4: Grundoperation und ihre exakten (=), relaxten ($\leftrightarrow$) oder ergebnisäquivalenten ($\sim$) CHASE-Inversen (Ausschnitt aus [3])

ergibt sich somit eine ergebnisäquivalente CHASE-Inverse, denn sowohl die Projektion mit Duplikatbildung, als auch der natürliche Verbund mit dangling tuples liefern die Existenz einer solchen Inversen. Existiert zu einer der Teiloperationen $\mathcal{M}_i$ keine CHASE-Inverse, so auch nicht zu $\mathcal{M}$.

Die Anwendung dieser theoretischen Überlegungen auf ein praxisbezogenes Beispiel, das Hidden-Markov-Modell, kann in [3] nachgelesen werden. Basierend auf der Darstellung des Hidden-Markov-Modells in Form von SQL-Anweisungen [21], können die benötigten Operationen Addition und Subtraktion, skalare Multiplikation und Division sowie die Matrix-Vektor- und Matrix-Matrix-Multiplikation auf die Existenz von CHASE-inversen Abbildungen untersucht werden. Insgesamt ergibt sich für die Komposition dieser Operationen eine ergebnisäquivalente CHASE-Inverse.

4. ZUSAMMENFASSUNG

Im Forschungsdatenmanagement soll neben der Auswertungsanfrage und dem -ergebnis selbst auch der verwendete Ausschnitt der Originaldatenbank archiviert werden. Falls sich die Daten oder das Schema der Forschungsdatenbank häufig verändern, müsste man nach jeder auf dem Datenbestand vorgenommenen Auswertung die Originaldatenbank „einfrieren“ und sichern. Um dies zu vermeiden, berechnen wir mit Hilfe von Provenance-Management-Techniken die minimale Teildatenbank, die gesichert werden muss, um das Anfrageergebnis wieder erzeugen zu können. Dazu haben wir Techniken der ***why***- und ***how***-Provenance mit der Theorie der inversen Schema-Abbildungen für Datenintegration und Datenaustausch kombiniert. Basis für die Transformationen auf den Daten zur Evolution von Schema und Daten als auch für die Transformation, die die Auswertungsanfrage darstellt, ist der CHASE. Damit können wir drei verschiedene Teilgebiete der Theorie zusammenführen.

In diesem Beitrag haben wir erste konkrete Teilergebnisse zur Klassifizierung von inversen Anfragen bei der Berechnung der ***why***-Provenance vorgestellt: Die Berechnung der Inversen nur mit Hilfe von Ergebnis und Auswertungsanfrage, die Bestimmung der Inversen mit Hilfe zusätzlicher Provenance-Informationen (Provenance-Polynome der ***how***-Provenance) sowie die Definition eines Datenbankausschnitts, der homomorph auf die Originaldatenbank abgebildet werden kann.

Literatur

[1] A. V. Aho, C. Beeri und J. D. Ullman. “The Theory of Joins in Relational Databases”. In: *ACM TODS* 4.3 (1979), S. 297–314. DOI: 10.1145/320083.320091.

[2] Y. Amsterdamer, D. Deutch und V. Tannen. “Provenance for Aggregate Queries”. In: *PODS*. ACM, 2011, S. 153–164.

[3] T. Auge. *Umsetzung von Provenance-Anfragen in Big-Data-Analytics-Umgebungen*. Universität Rostock. Masterarbeit. 2017.

[4] I. Bruder, A. Heuer, S. Schick und S. Spors. “Konzepte für das Forschungsdatenmanagement an der Universität Rostock”. In: *LWDA 2017*. Bd. 1917, S. 165–175. URL: http://ceur-ws.org/Vol-1917/paper25.pdf.

[5] I. Bruder, M. Klettke, M. L. Möller, F. Meyer, A. Heuer, S. Jürgensmann und S. Feistel. “Daten wie Sand am Meer - Datenerhebung, -strukturierung, -management und Data Provenance für die Ostseeforschung”. In: *Datenbank-Spektrum* 17.2 (2017), S. 183–196.

[6] P. Buneman, S. Khanna und W. C. Tan. “Why and Where: A Characterization of Data Provenance”. In: *ICDT*. Bd. 1. Springer. 2001, S. 316–330.

[7] J. Cheney, L. Chiticariu und W. C. Tan. “Provenance in Databases: Why, How, and Where”. In: *Foundations and Trends in Databases* 1.4 (2009), S. 379–474. DOI: 10.1561/1900000006.

[8] C. Curino, H. J. Moon, A. Deutsch und C. Zaniolo. “Update Rewriting and Integrity Constraint Maintenance in a Schema Evolution Support System: PRISM++”. In: *PVLDB* 4.2 (2010), S. 117–128. URL: http://www.vldb.org/pvldb/vol4/p117-curino.pdf.

[9] R. Fagin, P. G. Kolaitis, R. J. Miller und L. Popa. “Data Exchange: Semantics and Query Answering”. In: *Theor. Comput. Sci.* 336.1 (2005), S. 89–124.

[10] R. Fagin. “Inverting Schema Mappings”. In: *ACM TODS* 32.4 (2007), S. 25. DOI: 10.1145/1292609.1292615.

[11] R. Fagin, P. G. Kolaitis, L. Popa und W. C. Tan. “Quasi-Inverses of Schema Mappings”. In: *ACM TODS* 33.2 (2008), 11:1–11:52. DOI: 10.1145/1366102.1366108.

[12] R. Fagin, P. G. Kolaitis, L. Popa und W. C. Tan. “Schema Mapping Evolution Through Composition and Inversion”. In: *Schema Matching and Mapping*. Springer, 2011, S. 191–222.

[13] F. Geerts, G. Mecca, P. Papotti und D. Santoro. “That’s All Folks! LLUNATIC Goes Open Source”. In: *PVLDB* 7.13 (2014), S. 1565–1568.

[14] T. J. Green, G. Karvounarakis und V. Tannen. “Provenance semirings”. In: *PODS*. ACM, 2007, S. 31–40.

[15] T. J. Green und V. Tannen. “The Semiring Framework for Database Provenance”. In: *PODS 2017*. 2017, S. 93–99. DOI: 10.1145/3034786.3056125.

[16] M. Herschel. “A Hybrid Approach to Answering Why-Not Questions on Relational Query Results”. In: *J. Data and Information Quality* 5.3 (2015), 10:1–10:29. DOI: 10.1145/2665070.

[17] A. Heuer. “METIS in PArADISE: Provenance Management bei der Auswertung von Sensordatenmengen für die Entwicklung von Assistenzsystemen”. In: *BTW Workshops*. Bd. 242. LNI. 2015, S. 131–136.

[18] T. Johnston. *Bitemporal Data – Theory and Practice*. Morgan Kaufmann, 2014.

[19] S. Köhler, B. Ludäscher und D. Zinn. “First-Order Provenance Games”. In: *CoRR* abs/1309.2655 (2013).

[20] D. Maier, A. O. Mendelzon und Y. Sagiv. “Testing Implications of Data Dependencies”. In: *ACM TODS* 4.4 (1979), S. 455–469. DOI: 10.1145/320107.320115.

[21] D. Marten und A. Heuer. “Machine Learning on Large Databases: Transforming Hidden Markov Models to SQL Statements”. In: *OJDB* 4.1 (2017), S. 22–42.

[22] J. Svacina. *Intensional Answers for Provenance Queries in Big Data Analytics*. Universität Rostock. Bachelorarbeit. 2016.