Ein begriffsbasierter Ansatz zur semantischen Extraktion von Datenbankschemata

Henri Mühle, Hannes Voigt, Wolfgang Lehner

Database Technology Group
Technische Universität Dresden

Abstract: Die durch das rasante Anwachsen digitaler Datenbestände in Volumen und Vielfalt notwendig gewordene effiziente Verwaltung der erhobenen Datenbestände, bringt herkömmliche Datenbankmethoden an ihre Grenzen. Ein modelliertes Datenbankschema zur Grundstrukturierung der Datenbank kann längst nicht mehr statisch rigide modelliert werden. Vielmehr werden schemaflexible Datenbanken benötigt, die ihr Schema entsprechend an Änderungen im Datenbestand anpassen können. Da das Datenbankschema basierend auf einer konzeptuellen Datenbanksicht modelliert wird, präsentieren wir einen Ansatz, der die Formale Begriffsanalyse als Modellierungsmethode einsetzt. Die Formale Begriffsanalyse greift genau diese begriffsorientierte Weltsicht auf. Damit können wir Schemaextraktion und weiterführende Problemstellungen mit wohl verstandenen und gut untersuchten Mechanismen behandeln. Im Rahmen dieses Beitrages stellen wir ein begriffsbasiertes Verfahren zur Schemaextraktion vor, das sich genau diese konzeptuelle Weltsicht zu Nutze macht.

1 Einleitung

Mit dem rasanten Anwachsen digitaler Datenbestände geht die Anforderung nach einer effizienten Verwaltung der Bestände in immer neuen Anwendungsfeldern einher. Im Allgemeinen wird dafür auf die über Jahrzehnte bewährte Datenbanktechnologie zurückgegriffen. So sind Datenbanksysteme einer steten Diversifizierung ihres Anwendungsgebiets unterworfen. Dabei treten immer wieder die Grenzen ihrer zugrundeliegenden Konzeption zu Tage.

Das Basiskonzept eines jeden bewährten Datenbanksystems ist ein modelliertes Datenbankschema, welches die Grundstrukturierung für Datenablage und -anfrage vorgibt. Bei der Modellierung eines Datenbankschemas wird eine Begriffsbildung durchgeführt, indem gleichartige Datenwerte zu einem strukturellen Merkmal abstrahiert und diese zu Begriffen zusammen gefasst werden. In der Entity-Relationship-Modellierung erfolgt die Begriffsbildung mittels Attributen und Entitytypen; im konstruktiven Schemaentwurf mittels Neben- und Hauptprädikatoren. Ein Datenbankschema fasst nun die zur Beschreibung der zu verwaltenden Daten gebildeten Begriffe zusammen und macht sie dem Datenbanksystem verständlich. Das Datenbanksystem orientiert sich dann beim Aufbau seiner physischen Ablage stark am Datenbankschema, also an den gebildeten Begriffen. Damit erreicht man Redundanzfreiheit zur Vermeidung von Änderungsanomalien, sowie eine Eingrenzung der zu lesenden Daten für die Beantwortung von Anfragen an diese Begriffsstruktur. Wird zum Beispiel durch eine Anfrage nach einer Person gesucht, so liest das Datenbanksystem ausschließlich Datensätze die dem Begriff „Person“ genügen. Das Datenbankschema lässt sich so als eine vereinbarte Begriffswelt zwischen Anwendung und Datenbanksystem verstehen.

Entscheidend ist, dass die Begriffsbildung außerhalb des Datenbanksystems stattfindet, die Ratio der Begriffsbildung dem System also verborgen bleibt. Dies hat zum einen zur Folge, dass eine Begriffsumbildung stets ein Eingreifen des Modellierers erfordert und zum anderen, dass das entwickelte Datenbankschema als fix betrachtet wird. Änderungen und Anpassungen am Schema sind der Sonderfall und in der Durchführung meist aufwändig, da stets auch abhängige Daten angefasst und angepasst werden müssen. Das ist aber unproblematisch solange die Begriffsbildung vorab stattfinden kann und eine stabile Begriffswelt als Vereinbarung zwischen Anwendung und Datenbanksystem zum Resultat hat.

In vielen neuen Anwendungsgebieten gestalten sich jedoch beide Bereiche, die Begriffsbildung im Vorhinein und die Vereinbarung einer stabilen Begriffswelt, als schwierig bis unmöglich. Die Bildung von Begriffen vor dem eigentlichen Betrieb einer Datenbank setzt voraus, dass alle Daten vorab strukturell bekannt sind und sich zu Begriffen vereinheitlichen lassen. Für die Vereinbarung einer stabilen Begriffswelt ist zwingend Voraussetzung, dass die Anwendung über ein stabiles, also wenig veränderliches, Weltverständnis verfügt. In vielen Anwendungen sind jedoch weder die Daten vorab vollständig strukturell bekannt, noch existiert ein stabiles Weltverständnis seitens der Anwendung.

Ein Beispiel für solche Anwendungen, sind sogenannte Multi-Tenant-Systeme. Diese hosten eine Anwendung für unterschiedliche Mandanten (Tenants) mit unterschiedlichsten Bedürfnissen. Zwar ergibt sich vorab aus der Anwendung eine gemeinsame Basis-Begriffswelt, jedoch wird diese oft an die Bedürfnisse einzelner Mandanten angepasst. Zudem sind die Bedürfnisse späterer Mandanten nur in eingegrenztem Maße bekannt, so dass sie sich begrifflich schwer im Vorhinein erfassen lassen. Jeder Mandant bringt ein Stück weit sein eigenes Weltverständnis in die Gesamtanwendung mit ein. Gerade bei Geschäftsanwendungen ist das Weltverständnis durch sich verändernde gesetzliche Vorgaben und Rahmenbedingungen ständigen Anpassungen unterworfen. Ein Multi-Tenant-System kumuliert dies und ist so einer sehr instabilen Begriffswelt ausgesetzt. [AGJ+08, For08]

Als zweites Beispiel sollen hier Anwendungen zur Unterstützung von Wissensarbeitern dienen. Wissensarbeiter erkunden Datenbestände nach neuen Erkenntnissen. Ihr Vorgehen folgt nicht immer festen Pfaden und Algorithmen. Aus einer Erkenntnis entstehen neue Fragestellungen, denen der Wissensarbeiter nachgeht. In jedem Schritt zieht er, in Abhängigkeit von Verfügbarkeit und Eignung für die Fragestellung, neue Daten heran. Welche Daten der Wissensarbeiter verwendet und welche strukturelle Form diese haben kann vorab nicht bekannt sein, da es sich erst im Laufe des Arbeitsprozesses ergibt. Ziel eines Wissensarbeiters ist es gerade ein Weltverständnis aufzubauen bzw. auszuweiten, dementsprechend ist das Weltverständnis seitens der Anwendung per se instabil. [End08]

Um Datenbanksysteme zu einem effizienten Umgang mit einer flexiblen Begriffswelt zu befähigen, sehen wir es als unerlässlich an, das Datenbanksystem selbst zur Begriffsbildung zu befähigen. Mit der Formalen Begriffsanalyse stehen wohl verstandene und gut untersuchte Konzepte, Formalismen und Algorithmen bereit, um automatisiert eine Begriffsbildung vorzunehmen. In dieser Arbeit betrachten wir als einen ersten Schritt, wie die Formale Begriffsanalyse grundsätzlich zur strukturellen Organisation von Daten in einem Datenbanksystem eingesetzt werden kann. Darauf aufbauend können dann weiterführende Mechanismen entwickelt werden, die diese Begriffsbildung im Zuge einer Schemaevolution ausnutzen.

Dazu stellen wir in Abschnitt 2 die notwendigen Begrifflichkeiten der Formalen Begriffsanalyse vor. Das Verfahren selbst gliedert sich dann in drei Schritte: einen Abstraktionsschritt (Abschnitt 2.1), einen Kollabierungsschritt (Abschnitt 2.2 und Abschnitt 2.3) und einen Extraktionsschritt (Abschnitt 2.4). Abschließend geben wir eine Zusammenfassung (Abschnitt 3) und einen Ausblick auf nachfolgende Arbeiten (Abschnitt 4).

2 Finden von Schemakandidaten mit Hilfe Formaler Kontexte

Die Formale Begriffsanalyse ist ein mathematisches Teilgebiet, das sich der Mathematisierung von „Begriff“ und „Begriffshierarchie“ widmet [GW96]. Zentrale Elemente der Formalen Begriffsanalyse sind sogenannte **formale Kontexte**. Darunter versteht man Tripel (G, M, I), bestehend aus einer Menge G von **Gegenständen**, einer Menge M von **Merkmalen** und einer Inzidenzrelation $I \subseteq G \times M$, die beschreibt, ob ein Gegenstand $g \in G$ ein Merkmal $m \in M$ **hat**. Zur intuitiven Veranschaulichung formaler Kontexte werden Kreuztabellen verwendet, also Tabellen, deren Zeilen Gegenstände und deren Spalten Merkmale repräsentieren und in deren Zellen ein Kreuz steht, wenn der korrespondierende Gegenstand das korrespondierende Merkmal aufweist.

In diesen sehr allgemeinen Strukturen lassen sich nun **formale Begriffe** bilden. Das sind Paare (A, B) maximaler Teilmengen $A \subseteq G, B \subseteq M$, sodass jeder Gegenstand in A jedes Merkmal in B besitzt und gleichermaßen jedes Merkmal in B von jedem Gegenstand in A besessen wird. Formal findet man diese Begriffe mit Hilfe der folgenden Ableitungsoperatoren

$$A' := \{m \in M \mid \forall\, g \in A : gIm\}$$

$$B' := \{g \in G \mid \forall\, m \in B : gIm\}$$

sodass für einen Begriff (A, B) stets $A' = B$ und $B' = A$ gilt. Man nennt A den **Begriffsumfang** und B den **Begriffsinhalt**. Auf der Menge aller Begriffe $\mathfrak{B}(G, M, I)$ eines Kontextes (G, M, I) lässt sich eine Ordnungsrelation wie folgt definieren:

$$(A_1, B_1) \leq (A_2, B_2) :\Leftrightarrow A_1 \subseteq A_2 \quad (\Leftrightarrow B_1 \supseteq B_2)$$

Mit dieser Ordnung bilden die Begriffe eines Kontextes einen vollständigen Verband, den **Begriffsverband** $\underline{\mathfrak{B}}(G, M, I)$ des Kontextes (G, M, I) Unter allen Begriffen von (G, M, I) seien noch die Begriffe der Form $\gamma g := (g'', g')$ für $g \in G$ und $\mu m := (m', m'')$ für $m \in M$ ausgezeichnet, die sogenannten **Gegenstands-** bzw. **Merkmalbegriffe**. [GW96]

2.1 Überführung der Datenbank in einen formalen Kontext

Um eine schemabezogene Ablage der Datensätze einer Datenbank zu realisieren, ist es hilfreich das Datenbankschema zu kennen. Wird die Datenbank von vornherein sauber modelliert, liegt das Schema explizit vor und die Datenablage kann dementsprechend strukturiert werden. Moderne Anwendungen erzeugen allerdings zunehmend Datenmengen, die nicht explizit strukturiert sind.

Das **Datenbankschema** beschreibt die semantische Struktur der Datensätze und besteht aus einer Überdeckung der Datenbankattribute[1] durch *semantische Einheiten*.

Überführen wir eine Datenbank derart in einen formalen Kontext, dass wir jeden Datenbankeintrag, der verschieden von NULL ist, durch ein Kreuz repräsentieren, bieten uns die Begriffsinhalte des zugehörigen Begriffsverbandes gerade einen strukturierten Suchraum für eine solche Überdeckung. Betrachten wir als Beispiel den bereits abstrahierten Datenbestand aus Datensätzen der freien Datenbank Freebase[2] in Abbildung 1. Im zugehörigen Begriffsverband (Abbildung 2) findet man durch die Begriffe

[1] Eine **Überdeckung** einer Menge M ist eine Familie $\{M_t \mid t \in T\}$ von Teilmengen $M_t \subseteq M$ für eine beliebige Indexmenge T, so dass $\bigcup_{t \in T} M_t = M$.

[2] `http://www.freebase.com`

	Also known as	Date of Birth	Country of Nationality	Height	Weight	Position	Religion	President Number	Date founded	Country	Time Zone(s)	Population
Michael Jordan	×	×		×		×						
LeBron James	×	×		×	×	×						
Arnold Schwarzenegger	×	×		×	×		×					
Michael Schumacher		×	×									
Barack Obama	×	×					×	×				
Leeds	×										×	×
Berlin	×								×	×	×	×
New York City	×								×			×
Chicago	×								×		×	×

Abbildung 1: Ein formaler Kontext basierend auf Freebase-Datensätzen

eine Aufteilung der Datenbank in achtzehn logisch-strukturelle Einheiten. Der Begriffsverband bietet zudem eine visuelle Darstellung der Beziehungen zwischen diesen strukturellen Einheiten. Man sieht z. B., dass der grau markierte Begriff zum Datensatz *Chicago* ein Oberbegriff zu *Berlin* ist. Er prägt eine Teilmenge der Attribute seines Unterbegriffes aus, verallgemeinert diesen also.

Die Beschriftung des Verbandes ergibt sich so, dass ein Begriff ein Merkmallabel erhält, wenn er der größte Begriff ist, dessen Inhalt dieses Merkmal umfasst. Dual erhält er ein Gegenstandslabel, wenn er der kleinste Begriff ist, zu dessen Umfang dieser Gegenstand gehört. Die Merkmallabel oberhalb und die Gegenstandslabel unterhalb eines Begriffes ergeben dessen Zusammensetzung. Der markierte Begriff umfasst also gerade die Datensätze *Berlin* und *Chicago*, auf denen die Attribute *Population*, *Time Zone(s)*, *Date founded* und *Also known as* gemeinsam ausgeprägt sind.

Die Datenbank aus Abbildung 1 besitzt offensichtlich zwei semantische Einheiten, *Person* und *Stadt*. Natürlich bietet die Gesamtheit aller Begriffsinhalte eine Überdeckung der Merkmalmenge. Diese ist allerdings potentiell viel zu groß[3], als dass sie zur Strukturierung einer schemabezogenen Ablage in Frage kommt. Zudem bietet diese Überdeckung keine semantische Trennung der Datensätze, da die einzelnen Datensätze im Normalfall zu verschiedenen Begriffumfängen gehören. Unser Ziel ist es also, mit Hilfe einer Merkmalüberdeckung $\mathcal{M} := \{M_t \mid t \in T\}$ die Gegenstände so zu gruppieren, dass man jedem M_t eine Menge G_t von Datensätzen zuordnen kann, so dass $\mathcal{G} := \{G_t \mid t \in T\}$ eine Partition[4] der Gegenstandsmenge ist. T ist hierbei eine beliebige Indexmenge.

Bei der Erzeugung des Kontextes aus der Datenbank setzen wir nur dann Kreuze, wenn die Attributausprägung des jeweiligen Datensatzes explizit bekannt ist. Das bedeutet aber *nicht*, dass ein Datensatz ein Attribut, zu dem kein Kreuz existiert *nicht* hat. Es kann auch sein, dass er dieses Attribut zwar semantisch besitzt, der Attributwert aber nicht bekannt ist und der Datensatz das Attribut somit mit NULL ausprägt. Man spricht hierbei auch von *Unknown NULL-Values* bzw. von *Non-Applicable NULL-Values*. Auf einer logischen Ebene sind diese Datensätze natürlich als eigenständig zu betrachten, auf einer darüber stehenden semantischen Ebene können sie aber durchaus als Einheit angesehen werden. Genau solche semantischen Einheiten suchen wir.

[3]Zu einem formalen Kontext (G, M, I) kann es höchstens $2^{|M|}$ Begriffsinhalte geben.

[4]Eine **Partition** einer Menge G ist eine Überdeckung in disjunkte Mengen.

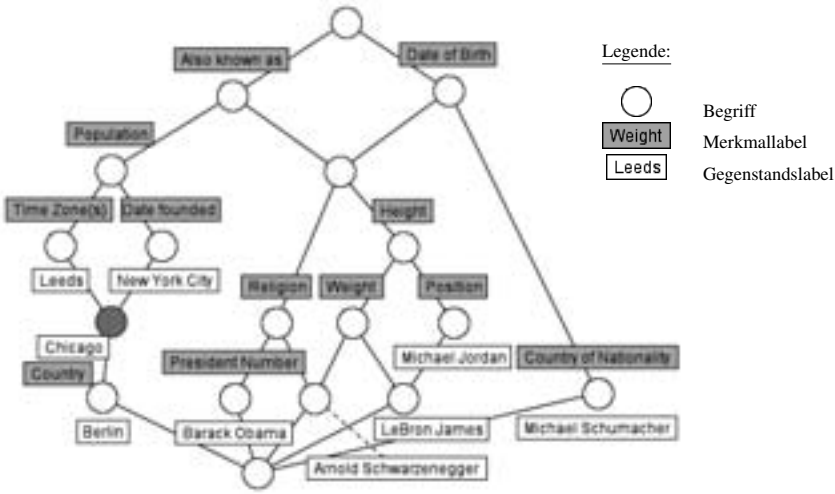

Abbildung 2: Der Begriffsverband zum Kontext aus Abbildung 1

2.2 Eine umfangsbezogene Ähnlichkeitsrelation auf den Merkmalen

Wir wollen unserem Ansatz die Annahme zu Grunde legen, dass ein Merkmal $n \in M$ eines Kontextes (G, M, I) mit einem gegebenen Merkmal $m \in M$ eine **semantische Einheit** bildet, wenn die Summe der Kreuze aus m' und $n' \cap m'$ einen gewissen Prozentsatz der Rechteckfläche $m' \times (m'' \cup n)$ überdeckt. Formal definieren wir damit auf den Merkmalen eines formalen Kontextes (G, M, I) eine Ähnlichkeitsrelation $\sim$ zu einem Schwellwert $t \in [0, 1]$, so dass für $m, n \in M$ gilt

$$m \sim n :\Leftrightarrow m' \cap n' \neq \emptyset \wedge \frac{|m'| \cdot |m''| + |m' \cap n'|}{|m'| \cdot |m'' \cup n|} \geq t$$

Außerdem schließen wir eine Ähnlichkeit zweier Merkmale aus, wenn ihre Merkmalumfänge disjunkt sind (sie also auf keinem Datensatz gemeinsam ausgeprägt sind). Eine Erweiterung auf Merkmalmengen (insbesondere Begriffsinhalte) erfolgt direkt.

Diese Art der Vergröberung begünstigt „große Begriffe“. Zu diesen lassen sich wesentlich einfacher ähnliche Merkmale finden, da die bereits ausgefüllte Kreuzfläche durch den Begriff beliebig groß werden kann. Haben wir zu einem Begriff (A, B) ein Merkmal $n \in M$ gefunden, mit $B \sim n$, dann erzeugen wir einen „gröberen Kontext“ $(G, M, \tilde{I})$, wobei

$$\tilde{I} := I \cup \{(g, n) \mid g \in A\}$$

Offenbar ist $(A, B \cup \{n\})$ dann ein Begriff von $(G, M, \tilde{I})$.

Basierend auf dieser Ähnlichkeit stellen wir im nächsten Abschnitt eine Kollabierung des Begriffsverbandes durch Vergrößern der Begriffe vor.

2.3 Kollabierung der Begriffswelt

Um den in Abschnitt 2.4 beschriebenen Extraktionsschritt eindeutig durchführen zu können, müssen wir einen irreduziblen Kontext voraussetzen. Ein Kontext heißt

Algorithmus 1 Der Algorithmus zur Kontextvergröberung

Require: Kontext (G, M, I), Threshold t
1: $(G, M, \tilde{I}) := (G, M, I)$
2: **repeat**
3: $(G, M, \hat{I}) := (G, M, \tilde{I})$
4: $\mathcal{B} := \emptyset$
5: **for all** $(A, B) \in$ SEARCHSPACE$(G, M, \tilde{I})$ **do**
6: **for all** $m \in M \setminus B$ **do**
7: **if** $m' \cap A \neq \emptyset \wedge \frac{|A| \cdot |B| + |A \cap m'|}{|A| \cdot |B \cup \{m\}|} \geq t$ **then**
8: $\mathcal{B} := \mathcal{B} \cup \big(A, B \cup \{m\}\big)$
9: **end if**
10: **end for**
11: **end for**
12: $(G, M, \tilde{I}) :=$ CREATEFROMCONCEPTS$(\mathcal{B})$
13: **until** $(G, M, \tilde{I}) = (G, M, \hat{I})$
14: **return** CREATESCHEMACONTEXT$(G, M, \tilde{I})$

irreduzibel, wenn es keinen Gegenstand gibt, dessen Inhalt sich als Durchschnitt anderer Gegenstandsinhalte darstellen lässt und die duale Forderung für die Merkmalumfänge gilt.

Binden wir dieses Vorgehen nun in einen iterativen Algorithmus ein, dann vergrößern wir sukzessive die Inzidenzrelation und führen dabei eine Art semantisches Clustering der Gegenstände durch. Unser Algorithmus folgt dabei einem Greedy-Ansatz, so dass in jedem Iterationsschritt möglichst viele Merkmale an die Begriffe angeheftet werden.

Ein Algorithmus, der alle Begriffe durchläuft und zu jedem Begriff die ähnlichen Merkmale herausfindet und darauf basierend sukzessive den Kontext vergröbert (Algorithmus 1), bietet in jedem Schritt eine neue Konfiguration von logischen Einheiten an, die einer Kollabierung der vorherigen Konfiguration entspricht. Im Idealfall konvergiert dieses Verfahren auf eine Konfiguration semantischer Einheiten hin. Der Algorithmus bricht ab, wenn keine derartige Vergröberung des Kontextes mehr möglich ist, also wenn entweder $\tilde{I} = G \times M$ oder wenn die gefundenen Merkmalinhalte, die echt kleiner als der größte Begriff $\top := (\emptyset', \emptyset'')$ sind, paarweise disjunkt sind.

Der Aufruf CREATEFROMCONCEPTS$(\mathcal{B})$ (Zeile 12) erzeugt aus einer Menge $\mathcal{B}$ von Paaren (A, B) mit $A \subseteq G, B \subseteq M$ einen formalen Kontext $(G, M, \tilde{I})$ mit $\tilde{I} := \{A \times B \mid (A, B) \in \mathcal{B}\}$. Die gewünschte, minimale Merkmalüberdeckung findet man dann über die Atome des Begriffsverbandes zum gröbsten Kontext. Ein Begriff heißt **Atom**, wenn er direkter oberer Nachbar des kleinsten Begriffes $\bot := (\emptyset'', \emptyset')$ ist. Die Inhalte der Atome eines Verbandes bilden offenbar stets eine minimale, nichttriviale Merkmalüberdeckung. Hierüber lässt sich auch ein alternatives Abbruchkriterium definieren, indem man eine maximale Anzahl semantischer Einheiten festlegt und den Algorithmus abbricht, wenn die Anzahl der Atome diesen Wert erreicht oder erstmals unterschreitet. Der Aufruf CREATESCHEMACONTEXT$(G, M, \tilde{I})$ in Zeile 14 erzeugt aus dem kollabierten Kontext $(G, M, \tilde{I})$ den Kontext der semantischen Einheiten, dessen Beschreibung in Abschnitt 2.4 folgt.

Aus komplexitätstheoretischer Sicht ist dieser Algorithmus auf dem naiven Suchraum *aller* Begriffe allerdings äußerst unangenehm, da dies exponentiell viele sein können. Wir schlagen daher vor, als Suchraum nur die Merkmalbegriffe heranzuziehen. Anschaulich prüfen wir damit zunächst die Merkmalbegriffe untereinander auf semantische Ähnlichkeit und versuchen so den Begriffsverband von oben herab zu kollabieren. Da jeder Begriff (A, B) Unterbegriff aller Merkmalbegriffe (m', m'') mit $m \in B$ ist, prüft das eingeschränkte Verfahren also zunächst, ob ein Begriff überhaupt Teil einer semantischen Einheit in unserem Sinne ist, ehe dieser Begriff um weitere

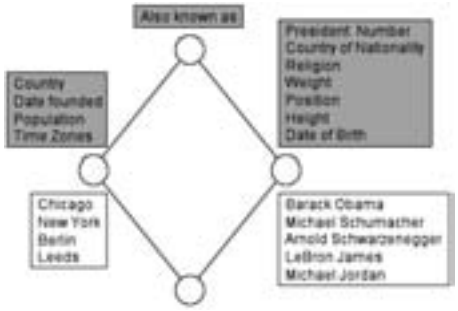

Abbildung 3: Der kollabierte Begriffsverband zum Kontext aus Abbildung 1 für $t = 0.8$

Merkmale angereichert wird. Wir erhalten also eine semantisch striktere Suche. Dadurch kann es allerdings passieren, dass man nicht den gleichen kollabierten Kontext erhält, wie im Ansatz über alle Begriffe. Dies wird durch die Bildung des Kontextes der semantischen Einheiten aber wieder ausgeglichen. Mit dieser Eingrenzung des Suchraumes verbessern wir die Komplexität dieses Algorithmus erheblich. Allgemein hat jede Iteration eine Komplexität von $O(|\text{SEARCHSPACE}| \cdot |M| \cdot |G|)$. Da es höchstens $|M|$ Merkmalbegriffe gibt, verringern wir die Komplexität also von $O(2^{|M|} \cdot |G| \cdot |M|)$ auf $O(|G| \cdot |M|^2)$.

2.4 Extraktion der semantischen Einheiten

Allgemein müssen wir zunächst festlegen, wie wir aus dem kollabierten Kontext $(G, M, \tilde{I})$ die gewünschten Überdeckungen $\mathcal{G}$ und $\mathcal{M}$ erhalten. Dazu wählen wir $\mathcal{M} := \left\{M_t \subseteq M \mid (M_t', M_t) \text{ Atom von } (G, M, \tilde{I})\right\}$. Für die Gegenstandsüberdeckung $\mathcal{G}$ wählen wir zu jedem Atom des kollabierten Kontextes genau die Gegenstände, deren Gegenstandsbegriffe oberhalb des Atoms liegen. Formal bedeutet das:

$$\mathcal{G} := \{G_t \subseteq G \mid t \in T\}, \text{ wobei } G_t := \{g \in G \mid \gamma g \geq (M_t', M_t)\}$$

T ist hierbei eine beliebige Indexmenge und γg bezeichnet den am Anfang von Abschnitt 2 eingeführten Gegenstandsbegriff zum Gegenstand g. Anschließend erzeugt man den **Kontext der semantischen Einheiten** (G, M, S) mittels

$$S := \{G_t \times M_t \mid t \in T\}$$

Wenn der kollabierte Kontext irreduzibel ist, also kein Gegenstandsbegriff als Supremum anderer Gegenstandsbegriffe darstellbar ist (für Merkmalbegriffe dual), ist auch der Kontext der semantischen Einheiten irreduzibel und liefert über seine Atomumfänge eine Partition der Gegenstandsmenge. Zudem erzeugt auch Algorithmus 1 aus irreduziblen Kontexten stets wieder irreduzible Kontexte, da keine neuen Begriffe zum Begriffsverband hinzugefügt, sondern lediglich vorhandene Begriffe miteinander vereinigt werden.

Auf die Datenbank aus Abbildung 1 angewendet, erhalten wir für einen Schwellwert von $t = 0.8$ mit unserem Algorithmus nach nur drei Iterationen den Begriffsverband der semantischen Einheiten in Abbildung 3. In diesem erkennen wir eine exakte Partitionierung der Gegenstände entsprechend der eingangs genannten semantischen Schemaelemente. Durch die Verwendung der Formalen Begriffsanalyse erhalten wir also direkt eine leicht verständliche Visualisierung der extrahierten semantischen Einheiten.

3 Zusammenfassung

Wir haben in diesem Artikel einen begriffsbasierten Ansatz zur semantischen Schemaextraktion aus nicht explizit strukturierten Datenbanken vorgestellt. Im Gegensatz zu anderen, graphen- oder logikbasierten Extraktionsverfahren, wie [BDFS97, LMP00, NAM98], nutzen wir explizit die strukturellen Informationen der Datenbank um den Suchraum von vornherein einzuschränken. Die Formale Begriffsanalyse bietet uns in diesem Zusammenhang einen theoretisch fundierten Ansatz um die vorausgesetzte instabile Begriffswelt zu modellieren und basierend auf einer erlaubten Unschärfe zusammenzufassen. Wir kollabieren dazu den Begriffsverband der Datenstruktur sukzessive, bis wir eine vorgegebene Anzahl von semantisch verschiedenen Schemaelementen unterschreiten. Anschließend extrahieren wir aus dem kollabierten Verband die Datensätze und Attribute, die den jeweiligen Schemaelementen genügen. Um dies zu erreichen, müssen wir eine Irreduzibilität des Datenbestandes voraussetzen, sodass es keine Datensätze gibt, deren Struktur aus anderen Datensätzen herleitbar ist. Insbesondere betrifft das Vererbungshierarchien auf den Typen der Datensätze. (Z. B. ist die Struktur eines Supertypen stets aus dem strukturellen Durchschnitt all seiner Subtypen herleitbar.) Da diese Reduzierung aber lediglich Datensätze entfernt, deren Struktur bereits implizit in anderen Datensätzen enthalten ist, ändert sich der Begriffsverband und damit der Suchraum für unseren Algorithmus *nicht*.

Unser Ansatz bietet zudem eine Erkennung von beliebig unstrukturierten Datenbeständen, indem der Algorithmus einen Kontext mit vollständig ausgefüllter Kreuztabelle zurück gibt. In diesem Fall gehören die Datensätze alle der gleichen semantischen Einheit an, bzw. sind diesbezüglich nicht unterscheidbar.

4 Ausblick

Eine Stärke unseres Ansatzes liegt in der sehr allgemeinen Modellierung durch die Formale Begriffsanalyse. Damit können wir nicht nur die in Abschnitt 1 genannten Anwendungsfälle in einer einheitlichen Sprache formulieren, sondern auch den gesamten Formalisierungsapparat auf datenbankspezifische Probleme anwenden. Somit erhalten wir eine neue Sicht auf die Problemstellung und damit auch einen gänzlich neuen Lösungsraum.

Literatur

[AGJ+08] Stefan Aulbach, Torsten Grust, Dean Jacobs, Alfons Kemper und Jan Rittinger. Multi-Tenant Databases for Software as a Service: Schema-mapping Techniques. In *SIGMOD'08*, 2008.

[BDFS97] Peter Buneman, Susan B. Davidson, Mary F. Fernandez und Dan Suciu. Adding Structure to unstructured Data. In *ICDT'97*, 1997.

[End08] Endeca. Endeca Information Access Platform, 2008.

[For08] Force.com. The Force.com Multitenant Architecture, 2008.

[GW96] Bernhard Ganter und Rudolf Wille. *Formale Begriffsanalyse: Mathematische Grundlagen*. Springer, 1996.

[LMP00] Pierre-Alain Laur, Florent Masseglia und Pascal Poncelet. Schema Mining: Finding Structural Regularity among Semistructured Data. In *Principles of Data Mining and Knowledge Discovery*, 2000.

[NAM98] Svetlozar Nestorov, Serge Abiteboul und Rajeev Motwani. Extracting Schema from Semistructured Data. In *SIGMOD'98*, 1998.