QSQL^p: Eine Erweiterung der probabilistischen Many-World-Semantik um Relevanzwahrscheinlichkeiten

Sebastian Lehrack, Sascha Saretz und Ingo Schmitt

Brandenburgische Technische Universität Cottbus
Institut für Informatik, Postfach 10 13 44
D-03013 Cottbus, Germany
{slehrack,sascha.saretz}@informatik.tu-cottbus.de,
schmitt@tu-cottbus.de

Zusammenfassung. Die traditionelle Auswertung einer Datenbankanfrage ermittelt für jedes Tupel entweder den Wahrheitswert *Wahr* oder *Falsch*. Für viele Anwendungsszenarien ist diese Auswertungssemantik zu restriktiv, insbesondere wenn ein differenzierteres Anfrageergebnis benötigt wird. Ein etablierter probabilistischer Ansatz zum Erreichen dieser Ausdifferenzierung ist die Verwendung sogenannter Relevanzwahrscheinlichkeiten: Mit welcher Wahrscheinlichkeit ist ein Dokument oder ein Datenobjekt bezüglich einer gestellten Anfrage relevant?
Neben den IR-motivierten Relevanzwahrscheinlichkeiten hat sich in der Datenbankforschung das Gebiet der probabilistischen Datenbanken etabliert. Auch hier wird ein striktes, deterministisches Auswertungsmodell als nicht mehr ausreichend angesehen. In probabilistischen Datenbanksystemen werden daher mehrere mögliche Zustände für ein und dasselbe System in einer gemeinsamen Datenbank verwaltet.
Die vorliegende Arbeit verbindet diese beiden probabilistischen Ansätze zu einem semantisch reicheren Anfragemodell.

1 Motivation

Die traditionelle Auswertung einer Datenbankanfrage ermittelt für jedes Tupel entweder den Wahrheitswert *Wahr* oder *Falsch*. Alle wahren Tupel bilden daraufhin die Ergebnismenge der Anfrage. Für viele Anwendungsszenarien ist diese Auswertungssemantik zu restriktiv, insbesondere wenn ein differenzierteres Anfrageergebnis benötigt wird. Die Ausdifferenzierung des Ergebnisses setzt oft eine Aussage über den Grad der Erfüllung einer gestellten Anfrage voraus. Ein etablierter Ansatz, welcher vor allem im Bereich des Information Retrievals weit verbreitet ist, drückt den Erfüllungsgrad mittels sogenannter Relevanzwahrscheinlichkeiten aus [20]: *Mit welcher Wahrscheinlichkeit wird ein Dokument oder ein Datenobjekt bezüglich einer gestellten Anfrage vom Anwender als relevant eingestuft?* Die Entscheidung, ob ein betrachtetes Dokument oder Datenobjekt für den Anwender relevant oder nicht relevant ist, wird in dem hier betrachteten Kontext in den Erfüllungsgrad einer logikbasierten Anfrage übertragen. Ein

zentraler Bestandteil dieser Art von Anfragen sind Ähnlichkeitsprädikate, z.B. *Preis möglichst um 100 Euro* oder *Ort nahe Cottbus*, deren reelle Auswertungsergebnisse aus dem Interval $[0; 1]$ als Relevanzwahrscheinlichkeiten interpretiert werden können.

Neben den Relevanzwahrscheinlichkeiten aus dem Bereich des Information Retrievals hat sich in der Datenbankforschung ebenfalls das Gebiet der probabilistischen Datenbanken etabliert. Auch hier wird ein striktes, deterministisches Auswertungsmodell als nicht mehr ausreichend angesehen. Insbesondere wenn Daten automatisch extrahiert werden oder aus verschiedenen Quellen stammen, existiert oft eine Unsicherheit über die Genauigkeit der so gewonnenen Daten. Neben der Unsicherheit von Daten sind menschliche Bewertungen oder Beobachtungen, welche auf Grund ihrer inhärenten Subjektivität oft mit einem Konfidenzwert annotiert werden, ein typisches Anwendungsfeld für probabilistische Datenbanken. Das vorherrschende Anfrage- und Datenmodell ist dabei die sogenannte *Many-World-Semantik*. Hier werden mehrere mögliche Zustände für ein und dasselbe System in einer gemeinsamen Datenbank verwaltet.

Die vorliegende Arbeit verbindet diese beiden probabilistischen Ansätze zu einem semantisch reicheren Anfragemodell. Insbesondere liefert sie Beiträge zu folgenden Schwerpunkten:

- die Erweiterung der Many-World-Semantik um Relevanzwahrscheinlichkeiten in einem erweiterten probabilistischen Anfragemodell,
- das Konzept einer differenzierten Normalisierung von probabilistischen Anfragen, sowie
- die praktische Umsetzung des entwickelten probabilistischem Anfragemodells durch die SQL-Erweiterung QSQL^p.

In den sich anschließenden Kapiteln sollen Beispielanfragen aus einem durchgängigen Szenario betrachtet werden. Das hier verwendete Beispielszenario beschäftigt sich mit der Beobachtung von Vögeln (Ornithologie). Hierfür werden die beiden Relationen *VBeob* (Vogelbeobachtung, siehe Abb. 1) und *VArt* (Vogelart, siehe Abb. 2) eingeführt. Für jedes Tupel der Relation Vogelbeobachtung ist ein individueller Konfidenzwert hinterlegt (Attribut *Pr*). Dagegen sind in der Relation Vogelart einzelne Eigenschaften, wie die Verbreitungsregion (Attribut *Region*) und ein charakteristisches Foto (Attribut *Bild*) der jeweiligen Vogelart abgespeichert.

2 Anfragemodelle

Um die graduelle Erfüllung von Anfragen zu ermöglichen wurden in der Vergangenheit verschiedene Ansätze entwickelt, so z.B. die Fuzzy Logik [22] von Zadeh, eine Vielzahl probabilistischer Verfahren (siehe Kapitel 6) und ein quantenlogisches Auswertungsmodell von Schmitt [18]. In diesem Kapitel soll gezeigt werden, wie die Anfrageergebnisse des quantenlogischen Auswertungsmodells als Relevanzwahrscheinlichkeiten interpretiert werden können, um diese anschließend mit der Many-World-Semantik zu kombinieren.

VBeob (Vogelbeobachtung)			
Art	*Ort*	*Zeit*	*Pr*
Star	Cottbus	September	0.9
Fink	Berlin	Juni	0.5
Amsel	Cottbus	Mai	0.4
Star	Cottbus	August	0.3
Drossel	Berlin	Juni	0.4

Abb. 1. Relation *VBeob*

VArt (Vogelart)		
Art	*Region*	*Bild*
Star	Mitteldeutschland	$\boxdot_1$
Fink	Norddeutschland	$\boxdot_2$
Amsel	Mitteldeutschland	$\boxdot_3$
Star	Süddeutschland	$\boxdot_4$

Abb. 2. Relation *VArt*

2.1 Relevanzwahrscheinlichkeiten im quantenlogischen Auswertungsmodell

Im Folgendem wird eine kurze Einführung in die Arbeitsweise des quantenlogischen Auswertungsmodells gegeben. Für eine tiefergehende Darstellung wird auf [10] und [18] verwiesen, wobei in [10] lediglich ein mathematisches Grundverständnis vorausgesetzt wird.

Die Grundidee dieses Ansatzes ist die Anwendung eines mathematischen Vektorraummodells aus der Quantenmechanik und -logik. Die abgefragten Tupel, sowie die gestellte Anfrage werden dabei als Bestandteile dieses Vektorraums modelliert. So werden z.B. die Attributwerte des abgefragten Tupels in die *Richtung eines normierten Vektors* abgebildet. Die gestellte Anfrage erzeugt dagegen ein eingebetteten Vektor*unter*raum, welcher auch als *Anfrageraum* bezeichnet wird. Der Anfrageraum verkörpert die gesamte Anfragesemantik. Das Auswertungsergebnis wird dann durch den minimalen einschließenden Winkel zwischen Tupelvektor und Anfrageraum bestimmt. Dabei bedeutet ein Winkel von 0° eine maximale Ähnlichkeit und ein Winkel von 90° repräsentiert eine maximale Unähnlichkeit zwischen dem betrachteten Tupel und der formulierten Anfrage. Setzt man den Winkel in die quadrierte Kosinus-Funktion ein, ergibt sich ein reeller Wert zwischen 0 (für 90°) und 1 (für 0°). Dieser Wert, welcher auch als *Score-Wert* bezeichnet wird, kann demnach als Ähnlichkeitsmaß interpretiert werden.

Neben dieser geometrischen Deutung existiert eine weitere Interpretation für den berechneten Score-Wert. Die Berechnung des Score-Wertes bezüglich eines Anfrageraumes genügt den Eigenschaften eines additiven Wahrscheinlichkeitsmaßes [11].

Damit drückt der Score-Wert aus, wie wahrscheinlich es ist, dass der betrachtete Tupelvektor komplett im angefragten Anfrageraum liegt. In diesem Fall würde ein einschließender Winkel von 0° und ein Score-Wert von $cos^2(0°) = 1$ vorliegen. Somit kann der Score-Wert auch als Relevanzwahrscheinlichkeit eines Tupels gegenüber einer Anfrage aufgefasst werden, was voraussetzt, dass die komplette Erfüllung der Anfrage den betrachteten Tupels als relevant einstuft. Das Wahrscheinlichkeitsmaß wird dabei über die Konstruktion des Tupelvektors und des Anfragevektorraumes definiert.

Interessanterweise kann die Berechnung der Relevanzwahrscheinlichkeiten für ein Tupel t, die in unserem Modell eine geometrische Interpretation besitzen, auf

die logische Struktur einer Anfrage c und die Anwendung der bekannten Aggregationsfunktionen für Wahrscheinlichkeiten unabhängiger Ereignisse zurückgeführt werden:

$$\begin{aligned} eval(t,c) &= SF_i(t,c) \qquad \text{falls } c \text{ ein Ähnlichkeits}\textit{prädikat} \text{ ist,} \\ eval(t,c_1 \wedge c_2) &= eval(t,c_1) * eval(t,c_2) \\ eval(t,c_1 \vee c_2) &= eval(t,c_1) + eval(t,c_2) - eval(t,c_1 \wedge c_2) \\ eval(t,\neg c) &= 1 - eval(t,c) \end{aligned}$$

Die Auswertung atomarer Ähnlichkeitsprädikate wird mittels sogenannter *Scoring-Funktionen* (SF_i) durchgeführt. Sie ermitteln einen reellen Wert aus dem Interval $[0;1]$, der als Relevanzwahrscheinlichkeit bezüglich des jeweiligen Ähnlichkeitsprädikates interpretiert werden kann. Ähnlichkeitsprädikate werden gemäß den verwendeten Auswertungsregeln als unabhängige Ereignisse verstanden. Sie dürfen deshalb innerhalb einer Anfrage nicht mehrfach mit unterschiedlichen Vergleichskonstanten auftreten. Damit wäre z.B. eine Kombination der Ähnlichkeitsprädikate *Ort in der Nähe von Cottbus* und *Ort in der Nähe von Berlin* unzulässig, da diese offensichtlich korrelieren. So könnten sie z.B. nicht gleichzeitig auf 1 (vollständig erfüllt) ausgewertet werden, da es sich um geographisch unterschiedliche Städte handelt.

Des Weiteren wird für die semantisch korrekte Anwendung der obigen Auswertungsfunktionen eine syntaktische Normalisierung der Anfrage notwendig, welche u.a. identische Teilbedingungen zusammenfasst und sich negierende Teilbedingungen eliminiert. Der in [18] vorgeschlagene Normalisierungsalgorithmus basiert auf bekannten, logischen Umformungsregeln, wie z.B. Idempotenz und Distributivität. Diese können hier angewendet werden, da es sich bei der zu Grunde liegenden mathematischen Strukur um eine Boolesche Algebra handelt.

In [14] wird aus diesem rein theoretischen Auswertungsmodell die Kalkülanfragesprache CQQL (Commuting Quantum Query Language) entwickelt. Sie erweitert den relationalen Bereichskalkül um die Behandlung von Ähnlichkeitsprädikaten und Anfragegewichtung. Ein typisches Anwendungsgebiet von CQQL sind Ähnlichkeitsprädikate, welche multimediale Inhalte einbeziehen. Im Kontext des eingeführten Beispielszenarios könnte eine Anfrage folgendermaßen lauten: *Bestimme die Relevanzwahrscheinlichkeit einer Vogelart bezüglich eines Vorgabebildes (VBild), falls sie in der Region Mitteldeutschland ansässig ist.* Die formalisierte CQQL-Anfrage ist gegeben durch:

$$\begin{aligned} &\{(Art, Region, Bild) \mid VArt(Art, Region, Bild) \wedge \\ &Region = \text{Mitteldeutschland} \wedge Bild \approx_{BV} \text{VBild}\}. \end{aligned}$$

Diese Anfrage besitzt mit ($Bild \approx_{BV}$ VBild) ein Ähnlichkeitsprädikat, welches durch eine spezielle Scoring-Funktion für Bildvergleiche ($\approx_{BV}$) ausgewertet wird.

Allgemein gesprochen wird die Unsicherheit des Anfrageergebnisses auf die Vagheit in der Anfrageformulierung zurückgeführt, wogegen die angefragten Daten selbst als gesichert vorausgesetzt werden: *Eine unsichere Anfrage wird auf einer sicheren Datengrundlage ausgeführt.*

2.2 Many-World-Semantik

Ein weit verbreitetes Semantikmodell für probabilistische Datenbanken ist die Many-World-Semantik [1]. Ausgangspunkt sind eine oder mehrere Tabellen, über welche die Menge aller möglichen Instanzen (hier als Welten oder Zustände bezeichnet) der entsprechenden Relationenschematas betrachtet wird. Die Ausgangstabellen können somit entsprechend ihrer Relationenschemata eine maximal mögliche Menge von Tupeln besitzen. Jede Untermenge dieser maximalen Tupelmenge repräsentiert einen möglichen Zustand der Tabelle. Als Beispiel soll eine Tabelle mit maximal zwei Tupeln betrachtet werden $R(A_1) = \{(1), (2)\}$. Die möglichen vier Zustände lauten hier: $R_{Z_1}(A_1) = \{(1), (2)\}$, $R_{Z_2}(A_1) = \{(1)\}$, $R_{Z_3}(A_1) = \{(2)\}$ und $R_{Z_4}(A_1) = \{\}$. Einer dieser Zustände stellt die Realität dar. Welcher genau dies ist, ist jedoch unbekannt.

Vielmehr wird über Menge der Zustände ein Wahrscheinlichkeitsmaß definiert. Es drückt aus, mit welcher Wahrscheinlichkeit $Pr(Z_i)$ ein bestimmter Zustand Z_i der reale Zustand ist. Zustände können hierbei auch eine Wahrscheinlichkeit von Null besitzen.

Die Wahrscheinlichkeiten der einzelnen Zustände werden anhand der Tupel, welche in dem jeweiligen Zustand existieren definiert. Hierfür ist jedem Tupel t_i eine Eintrittswahrscheinlichkeit $Pr(t_i)$ zugeordnet, die ausdrückt, mit welcher Wahrscheinlichkeit es in der Realität vorkommt.

Prinzipiell ist die Many-World-Semantik nicht auf eine bestimmte Klasse von Wahrscheinlichkeitsmaßen festgelegt. Um jedoch eine möglichst einfache Berechnung der Zustandswahrscheinlichkeiten $Pr(Z_i)$ zu gewährleisten, werden die Eintrittswahrscheinlichkeiten der Tupel $Pr(t_i)$ als untereinander unabhängig angenommen. Dies bedeutet, die Eintrittswahrscheinlichkeit eines bestimmten Tupels ändert sich nicht mit dem Vorhandensein oder dem Nicht-Vorhandensein eines beliebigen anderen Tupels. Somit ergibt sich die Wahrscheinlichkeit eines Zustandes als $Pr(Z_i) = \Pi_{t_i \in Z_i}(Pr(t_i)) * \Pi_{t_i \notin Z_i}(1 - Pr(t_i))$.

Mit Eintrittswahrscheinlichkeiten für Tupel lassen sich u.a. besonders gut Beobachtungen und Bewertungen modellieren, welche einer bestimmten Unsicherheit bzw. Subjektivität unterliegen. Die Eintrittswahrscheinlichkeiten/Konfidenzwerte solcher Beobachtungen bzw. Bewertungen werden meist durch Expertenwissen bestimmt, das sich meist nur sehr unzureichend in Funktionen oder automatischen Verfahren abbilden lässt.

In dem eingeführten Beispielszenario stellen die Tupel der Tabelle *VBeob* solche subjektiven Beobachtungen dar. Die im Attribut *Pr* hinterlegten Eintrittswahrscheinlichkeiten sind von dem jeweiligen Beobachter auf Basis seines eigenen individuellen Erfahrungshorizonts bestimmt worden.

Eintrittswahrscheinlichkeiten von Tupeln aus einer Datenrelation stellen singuläre Basisereignisse dar. Dem gegenüber stehen *komplexe* Ereignisse, welche im Zuge der Anfrageauswertung aus der Kombination von Basisereignissen konstruiert werden.

Eine typische Many-World-Anfrage mit komplexen Ereignissen könnte wie folgt lauten: *Bestimme alle Zweier-Kombinationen von unterschiedlichen Vogelarten, die am selben Ort beobachten worden sind.* Wenn man die Beispielanfrage

auf die Tabelle *VBeob* anwendet ergibt sich u.a. die Kombination Star und Amsel. Das Eintreten dieser Kombination stellt ein komplexes Ereignis dar, welches sich aus zwei gleichzeitig eintretenden unabhängigen Basisereignissen zusammensetzt: $Pr((\text{Star}, \text{Amsel}, \text{Cottbus})) = Pr((\text{Star}, \text{Cottbus}, \text{September}))*$
$Pr((\text{Amsel}, \text{Cottbus}, \text{Mai})) = 0.36$

Zusammenfassend kann festgestellt werden, dass im Gegensatz zum vorherigen Semantikmodell hier die Daten als unsicher betrachtet werden: *Eine sichere Anfrage wird auf einer unsicheren Datengrundlage ausgeführt.*

2.3 Die Erweiterung der Many-World-Semantik um Relevanzwahrscheinlichkeiten

Die Kombination der beiden oben beschriebenen Semantikmodelle ergibt eine erweiterte Klasse von Anfragen. Ausgehend von einem Tupel in einer bestimmten Welt kann nun zusätzlich die Relevanz dieses Tupels bezüglich einer Ähnlichkeitsanfrage betrachtet werden. Als Beispiel soll folgende Anfrage gestellt werden: *Bestimme alle Vogelarten, welche beobachtet worden sind und zusätzlich möglichst ähnlich einem Vorgabebild (VBild) sind.* Die Bedingung kann wie folgt formalisiert werden:

$$VBeob(Art, Ort, Zeit) \wedge VArt(Art, Region, Bild) \wedge Bild \approx_{BV} \text{VBild}.$$

In dieser Beispielanfrage wird die Eintrittswahrscheinlichkeit der Beobachtung mit der Relevanzwahrscheinlichkeit der Beobachtung bezüglich des Ähnlichkeitsprädikates $Bild \approx_{BV}$ VBild verknüpft.

Die Kombination beider Anfrageparadigmen wird immer dann interessant, wenn konstruierte Datenobjekte mit komplexen Eintrittsereignissen assoziiert werden und auf den Attributwerten dieser Datenobjekte logikbasierte Ähnlichkeitsanfragen ausgeführt werden. Es wird somit eine Verbindung zwischen einer *subjektiven* Quantifizierung von Ereignissen und der *objektiven* Berechnung von Ähnlichkeitswerten realisiert.

In Anlehnung an die beiden vorangegangenen Abschnitte kann folgender Grundsatz für die Kombination von Relevanzwahrscheinlichkeiten und Many-World-Semantik formuliert werden: *Eine unsichere Anfrage wird auf einer unsicheren Datengrundlage ausgeführt.*

3 CQQL^p - Die probabilistische Erweiterung der Anfragesprache CQQL

Im vorherigen Kapitel wurde die erweiterte Anfrageklasse vorgestellt, welche sich aus der Kombination von Relevanzwahrscheinlichkeiten und der Many-World-Semantik ergibt.

Die technische Berechnung der kombinierten Wahrscheinlichkeiten basiert auf einem integrierten Wahrscheinlichkeitsmaß, welches auf einem Produktwahrscheinlichkeitsraum zwischen der Menge aller möglichen Welten und der Menge aller Anfrageräume definiert wird [11].

Die daraus resultierende probabilistische Erweiterung von CQQL wird als CQQL^p bezeichnet. In den folgenden Abschnitten werden grundlegende Konzepte von CQQL^p vorgestellt. Eine genaue Defintition von CQQL^p wird in [9] gegeben.

3.1 Probabilistische Relationen und probabilistische Relationenprädikate

Als erster Schritt wird das Konzept der *probabilistischen Relation* in den Sprachumfang von CQQL^p eingeführt. In probabilistischen Relationen besitzt jedes Tupel eine individuelle Eintrittswahrscheinlichkeit. Die Eintrittswahrscheinlichkeit stellt dabei kein explizites Attribut dar, d.h. sie kann nicht direkt manipuliert werden. Die definierten Eintrittswahrscheinlichkeiten werden als untereinander unabhängig vereinbart.

Bisher konnte eine CQQL-Formel aus drei verschiedenen Typen von Prädikaten bestehen [14]: (1) Relationenprädikate (z.B. $R_1(X_1, X_2)$), (2) Boolesche Prädikate (z.B. $X_1 = 2$ oder $X_2 < 5$) und (3) Ähnlichkeitsprädikate (z.B. $X_3 \approx 4$). Für die Auswertung von probabilistischen Relationen wird in CQQL^p der neue Typ der *probabilistischen Relationenprädikate* (Notation: $R_i^{\approx}(X_1, \ldots, X_n)$) eingeführt. Wird ein solches probabilistisches Relationenprädikat auf ein bestimmtes Tupel angewendet, ist der entsprechende Rückgabewert die Eintrittswahrscheinlichkeit dieses Tupels, falls es sich in der Relation befindet. Andernfalls wird der Wert 0 zurück gegeben.

Als Anwendungsbeispiel wird folgende Anfrage betrachtet: *Bestimme alle Vogelarten, welche in Cottbus im September beobachtet worden sind.* Die formalisierte Anfrage in CQQL^p lautet:

$$\{(Art, Ort, Zeit) \mid VBeob^{\approx}(Art, Ort, Zeit) \wedge Ort = \text{Cottbus} \wedge Zeit = \text{September}\}.$$

Die Auswertung der Anfrage ergibt für das Tupel (Star, Cottbus, September) der Relation *VBeob* eine Wahrscheinlichkeit von $eval(VBeob^{\approx}(\text{Star}, \text{Cottbus}, \text{September})) * eval(\text{Cottbus} = \text{Cottbus}) * eval(\text{September} = \text{September}) = 0.9 * 1 * 1 = 0.9$ und für das Tupel (Fink, Berlin, Juni) von $eval(VBeob^{\approx}(\text{Fink}, \text{Berlin}, \text{Juni})) * eval(\text{Berlin} = \text{Cottbus}) * eval(\text{Juni} = \text{September}) = 0.5 * 0 * 0 = 0$.

3.2 Probabilistische Normalisierung

Ein zentraler Bestandteil der CQQL-Auswertung ist die syntaktische Normalisierung von Anfragen. Sie garantiert die semantisch korrekte Aggregierung der Relevanzwahrscheinlichkeiten von Ähnlichkeitsprädikaten. So wird etwa die Beispielbedingung $(Ort \approx_{OV} \text{Cottbus}) \wedge (Ort \approx_{OV} \text{Cottbus})$ zu $(Ort \approx_{OV} \text{Cottbus})$ normalisiert, weil es sich semantisch um die Konjunktion ein und derselben Bedingung handelt ($\approx_{OV}$ ist Ähnlichkeitsoperator für ein Ortsvergleich). Die direkte Auswertung der unnormalisierten Anfrage würde eine falsche Relevanzwahrscheinlichkeit von $eval(Ort \approx_{OV} \text{Cottbus}) * eval(Ort \approx_{OV} \text{Cottbus})$ anstatt von $eval(Ort \approx_{OV} \text{Cottbus})$ ergeben.

Die Normalisierung von Ähnlichkeitsprädikaten wird nun auf probabilistische Relationenprädikate übertragen. Dadurch wird z.B. gewährleistet, dass Eintrittswahrscheinlichkeiten gleicher Tupel nicht mehrfach in die Gesamtwahrscheinlichkeit eingehen. Als Beispiel wird der Schnitt der Relation *VBeob* mit sich selbst betrachtet: $\{(Art, Ort, Zeit) \mid VBeob^{\approx}(Art, Ort, Zeit) \wedge VBeob^{\approx}(Art, Ort, Zeit)\}$. Sobald man ein konkretes Tupel mit Hilfe dieser unnormalisierten Bedingung auswertet, erkennt man, dass die Eintrittswahrscheinlichkeiten ein und desselben Tupels zweimal in die Gesamtwahrscheinlichkeit des Ergebnistupels eingehen würde. Dies widerspricht der probabilistischen Many-World-Semantik. Auch hier ist eine Normalisierung der Formel notwendig. In diesem Fall vereinfacht sich die Bedingung zu $VBeob^{\approx}(Art, Ort, Zeit)$.

3.3 Intra-Tupel versus Inter-Normalisierung

Im letzten Abschnitt wurde zum einen die Normalisierung von Ähnlichkeitsprädikaten und zum anderen die Normalisierung von probabilistischen Relationenprädikaten vorgestellt. Die erste Normalisierung garantiert die korrekte Aggregierung von Relevanzwahrscheinlichkeiten, die zweite ist dagegen dafür verantwortlich, dass Eintrittswahrscheinlichkeiten semantisch richtig zusammengefasst werden.

Betrachtet man die Normalisierung von Ähnlichkeitsprädikaten genauer, erkennt man, dass sich die zu normalisierenden Ereignisse auf Attributwerte genau eines Tupels bzw. genau einer Variablenbelegung beziehen. Dies entspricht exakt dem quantenlogischen Auswertungsmodell, da hier die Auswertung für einen einzelnen Vektor gegenüber einem Anfrageraum definiert wird. Eine Interaktion zwischen verschiedenen Vektoren innerhalb der Auswertung ist nicht vorgesehen. Daher kann die Normalisierung von Ähnlichkeitsprädikaten als *Intra-Tupel*-Normalisierung bezeichnet werden. Sie wirkt nur innerhalb eines Tupels bzw. einer Variablenbelegung.

Die Normalisierung von probabilistischen Relationenprädikaten unterstützt dagegen die Bildung von komplexen Ereignissen, welche die Eintrittswahrscheinlichkeiten von konstruierten Tupeln verkörpern. Komplexe Ereignisse dieser Art beziehen sich definitionsgemäß auf mehrere Basistupel bzw. Variablenbelegungen. Demnach findet eine *Inter*-Normalisierung zwischen mehreren Tupeln bzw. Variablenbelegungen statt. Eine typische Operation, die eine Inter-Normalisierung notwendig macht, ist die Projektion. Hier können mehrere Ausgangstupel zu einem Ergebnistupel verdichtet werden. Die Wahrscheinlichkeit des Ergebnistupel ergibt sich aus einer disjunktiven Verknüpfung der Wahrscheinlichkeiten der jeweiligen Ausgangstupel. Dies bedeutet, dass mindestens eines der Ausgangsereignisse eingetreten sein muss, um das Ereignis des verdichteten Tupels zu erzeugen [6].

Die durch die Projektion erzeugte Disjunktion muss jedoch mit einer Inter-Normalisierung behandelt werden, da Basisereignisse mehrfach in den möglicherweise komplexen Ereignissen der Ausgangstupel vorliegen können.

Als Beispiel soll die folgende Anfrage betrachtet werden: *Bestimme alle Vogelarten, welche in der Nähe von Berlin oder in der Nähe von Berlin beobachtet worden sind.* Die formalisierte Variante dieser Anfrage lautet:

$$\{(Art) \mid \exists Ort : \exists Zeit : VBeob^{\approx}(Art, Ort, Zeit) \wedge \\ (Ort \approx_{OV} \text{Berlin} \vee Ort \approx_{OV} \text{Berlin})\}.$$

Das doppelte Auftreten eines Ähnlichkeitsprädikates kann z.B. durch die automatisierte Generierung von Anfragen oder durch die Anwendung von Sichten auftreten.

Bei der Auswertung der Beispielanfrage muss sowohl eine Intra-Tupel- als auch eine Inter-Normalisierung durchgeführt werden. Zunächst wird die Intra-Tupel-Normalisierung auf die Bedingung $(Ort \approx_{OV} \text{Berlin} \vee Ort \approx_{OV} \text{Berlin})$ angewendet: $(Ort \approx_{OV} \text{Berlin})$. Somit ergeben sich im ersten Schritt für das Tupel (Star,Cottbus,September) die Wahrscheinlichkeit $eval(VBeob^{\approx}(\text{Star}, \text{Cottbus}, \text{September}) \wedge \text{Cottbus} \approx_{OV} \text{Berlin})$ und für das Tupel (Star,Cottbus,August) die Wahrscheinlichkeit $eval(VBeob^{\approx}(\text{Star}, \text{Cottbus}, \text{August}) \wedge \text{Cottbus} \approx_{OV} \text{Berlin})$. Anschließend muss eine Inter-Normalisierung auf die Projektion[1] des Attributes *Art* durchgeführt werden. Da sich die Inter-Normalisierung nur auf probabilistische Relationenprädikate bezieht, verändert sie die disjunktiv konstruierte Formel für das Ergebnistupel (Star) hier nicht mehr:

$$eval((VBeob^{\approx}(\text{Star}, \text{Cottbus}, \text{September}) \wedge \text{Cottbus} \approx_{OV} \\ \text{Berlin}) \vee (VBeob^{\approx}(\text{Star}, \text{Cottbus}, \text{August}) \wedge \text{Cottbus} \approx_{OV} \text{Berlin})) = \\ (1 - (1 - (0.9 * 0.7))(1 - (0.3 * 0.7))) = 0.7077,$$

wenn $eval(\text{Cottbus} \approx_{OV} \text{Berlin})$ als 0.7 angenommen wird[2].

4 Die probabilistische Anfragesprache QSQL^p

Die Anfragesprache SQL ist der etablierte Standard für den Zugriff auf objektrelationale Datenbanksysteme. Seit der Einführung von SQL in den 70er Jahren ist ihre praktische Relevanz kontinuierlich gestiegen. Aus diesem Grund werden die in Kapitel 3 vorgestellten Konzepte der Kalkülanfragesprache CQQL^p auf SQL übertragen. Dadurch werden sie in Form des SQL-Dialektes QSQL^p einer breiten Entwicklerschicht zugänglich gemacht. Der bisherige Funktionsumfang von SQL bleibt dabei vollständig in QSQL^p erhalten, d.h. alle SQL-Anfragen können auch in QSQL^p wie gewohnt formuliert und ausgewertet werden.

Tupel von probabilistischen Relationen besitzen eine individuelle Eintrittswahrscheinlichkeit. QSQL^p benutzt als Eintrittswahrscheinlichkeit automatisch die Werte des Attributes `probvalue`, falls es in der Relation vorhanden ist. Andernfalls wird für jedes Tupel implizit eine Eintrittswahrscheinlichkeit von 1 angenommen. Neben der expliziten Speicherung der Eintrittswahrscheinlichkeiten

[1] Im Kalkül wird eine Projektion mittels (mehrerer) Existenzquantoren ausgedrückt, welche die nicht projizierten Attribute binden.

[2] Wegen der DeMorgan-Umformungsregel gilt: $eval(A \vee B) = eval(\neg(\neg A \wedge \neg B)) = (1 - (1 - eval(A)) * (1 - eval(B)))$

können diese auch mittels von Unterabfragen berechnet werden. Die berechneten Wahrscheinlichkeiten befinden sich dann wiederum in dem Attribut `probvalue` der Ergebnisrelation.

Die Selektion von Tupeln aus einer oder mehreren Tabellen wird syntaktisch wie in SQL formuliert. So wird die logische Anfragebedingung, welche sich aus Booleschen Prädikaten, Ähnlichkeitsprädikaten, sowie den logischen Operatoren `and`, `or` und `not` zusammensetzt, ebenfalls in der `where`-Klausel einer Anfrage platziert. Gegenüber SQL können in QSQL^p zusätzlich Ähnlichkeitsbedingungen mittels des Ähnlichkeitsoperators $\approx$ formuliert werden. Eine Beispielanfrage in QSQL^p wird in Abschnitt 4.3 vorgestellt.

4.1 Der Auswertungsprozess von QSQL^p

Die interne Ergebnisberechnung einer QSQL^p-Anfrage wird mittels einer Transformation zwischen den folgenden drei Anfragesprachen realisiert: (1) QSQL^p zur Formulierung der Anfrage, (2) die Ähnlichkeitsalgebra QA^p zur Normalisierung und Optimierung, sowie (3) SQL-99 zur eigentlichen Berechnung des Ergebnisses innerhalb eines DBMS (siehe Abbildung 3).

In den nächsten Abschnitten wird die Normalisierung und Optimierung von QA^p-Ausdrücken skizziert. Eine exakte Definition der Ähnlichkeitsalgebra QA^p, sowie eine detaillierte Beschreibung der Abbildung von QSQL^p nach QA^p wird in [12] gegeben. Die verwendeten Prinzipien für die finale Abbildung nach SQL-99 wurden bereits in der bisherigen QSQL-Version eingesetzt und werden in [13] vorgestellt.

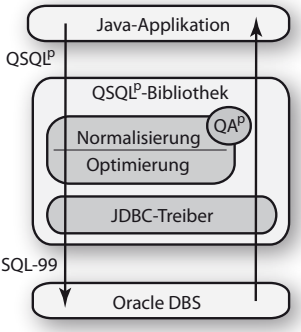

Abb. 3. Auswertungsprozess

4.2 Die Ähnlichkeitsalgebra QA^p

Das Kernstück der Auswertung von QSQL^p-Anfragen ist die Erzeugung von semantisch äquivalenten Ausdrücken in QA^p und deren Optimierung.

Die probabilistische Normalisierung von Prädikaten wurde bereits im Kontext von CQQL^p in Kapitel 3 diskutiert. Die dort entwickelten Konzepte werden nun auf die Ähnlichkeitsalgebra QA^p angewendet. Damit werden die Ähnlichkeitsalgebra QA^p und die Menge der *sicheren* CQQL^p-Anfragen gleichmächtig [12]. Eine CQQL^p-Anfrage gilt als sicher, wenn ihre Ergebnismenge endlich ist und sie darüber hinaus in endlicher Zeit berechnet werden kann.

Die Operatoren der Ähnlichkeitsalgebra QA^p werden in Tabelle 1 aufgeführt. Das Ergebnis eines jeden Operators ist ein Tupel (R, Pr), welches aus dem relationalen Datenanteil R und der Wahrscheinlichkeitsfunktion Pr besteht. Die

Funktion Pr ordnet jedem Tupel aus R eine Wahrscheinlichkeit zu. Die Berechnung von R wird dabei mit den bekannten Operatoren aus der Relationalen Algebra durchgeführt.

Probabilistische Auswertungsoperatoren werden gemeinhin in *extensionale* und *intensionale* Operatoren unterteilt (siehe z.B. [4], [6]). Dabei aggregieren extensionale Operatoren Wahrscheinlichkeiten ohne die zu Grunde liegenden (komplexen) Ereignisse zu berücksichtigen. Die richtige Semantik muss vielmehr durch die richtige Anordnung der Operatoren innerhalb des Ausdruckes garantiert werden. Dagegen besitzen intensionale Operatoren zur Berechnung der richtigen Ergebniswahrscheinlichkeiten eine interne Normalisierung. Diese stellt im Allgemeinen einen signifikanten Mehraufwand dar.

Operation	Semantik
(Prob.) Relation R, R^p	$R := R$ $Pr(t) := 1$ für R bzw. $Pr(t)$ wird gesetzt für R^p
Projektion -extens.- $\pi^e_{\mathcal{A}}(E_1)$	$R := \pi^{RA}_{\mathcal{A}}(R_1)$ $Pr(t) := 1 - \Pi_{\tilde{t} \in \{\tilde{t} \in R_1 \mid \tilde{t}[\mathcal{A}]=t\}}(1 - Pr(\tilde{t}))$
Projektion -intens.- $\pi^i_{(\mathcal{A},F)}(E_1)$	$R := \pi^{RA}_{\mathcal{A}}(R_1)$ $Pr(t) := eval(\mathtt{norm}_{inter}(\vee_{\tilde{t} \in \{\tilde{t} \in R_1 \mid \tilde{t}[\mathcal{A}]=t\}} F(\tilde{t})))$
Selektion $\sigma_F(E_1)$	$R := \{t \in R_1 \mid Pr(t) > 0\}$ $Pr(t) := Pr_1(t) * eval(\mathtt{norm}_{intra}(F(t)))$
Schnitt $E_1 \cap_{(\mathcal{A}_1,\mathcal{A}_2)} E_2$	$R := R_1 \bowtie^{RA}_{natural\ \beta_{(\mathcal{A}_1 \leftarrow \mathcal{A}_2)}} (R_2)$ $Pr(t) := Pr_1(t[R_1]) * Pr_2(t[R_2])$
Vereinigung $E_1 \cup_{(\mathcal{A}_1,\mathcal{A}_2)} E_2$	$R := R_1 \bowtie^{RA}_{full\ outer\ \beta_{(\mathcal{A}_1 \leftarrow \mathcal{A}_2)}} (R_2)$ $Pr(t) := \begin{cases} Pr_1(t[R_1]) + Pr_2(t[R_2]) - Pr_1(t[R_1]) * Pr_2(t[R_2]) & \text{falls } t[R_1] \in R_1 \wedge t[R_2] \in R_2 \\ Pr_1(t[R_1]) & \text{falls } t[R_1] \in R_1 \wedge t[R_2] \notin R_2 \\ Pr_2(t[R_2]) & \text{falls } t[R_1] \notin R_1 \wedge t[R_2] \in R_2 \end{cases}$
Differenz $E_1 -_{(\mathcal{A}_1,\mathcal{A}_2)} E_2$	$R := R_1 \bowtie^{RA}_{left\ outer\ \beta_{(\mathcal{A}_1 \leftarrow \mathcal{A}_2)}} (R_2)$ $Pr(t) := \begin{cases} Pr_1(t[R_1]) * (1 - Pr_2(t[R_2])) & \text{falls } t[R_1] \in R_1 \wedge t[R_2] \in R_2 \\ Pr_1(t[R_1]) & \text{falls } t[R_1] \in R_1 \wedge t[R_2] \notin R_2 \end{cases}$
Kreuzprodukt $E_1 \times E_2$	$R := R_1 \times^{RA} R_2$ $Pr(t) := Pr_1(t[R_1]) * Pr_2(t[R_2])$

Tabelle 1. Übersicht der QA^p-Operatoren

4.3 Die Abbildung von QSQL^p nach QA^p

Da die Auswertung einer QSQL^p-Anfrage mittels QA^p-Ausdrücke geschieht, ist die Semantik von QSQL^p mittels der Abbildung von QSQL^p nach QA^p und der Definition der QA^p-Operatoren festgelegt. Dies wiederum bedingt die Gleichmächtigkeit zwischen der *Kernfunktionalität* von QSQL^p und dem sicheren CQQL^p-Kalkül, da bereits eine Äquivalenz zwischen QA^p und CQQL^p festgestellt wurde. Der Begriff Kernfunktionalität bezieht sich auf den Umstand, dass bestimmte SQL-Funktionalitäten wie die Gruppierung und die Multimengen-Semantik nicht direkt in eine Kalkülsprache, welche auf Prädikatenlogik 1. Stufe basiert, übertragen werden können.

Der Dreiklang von sicherem CQQL^p (Kalkül), QA^p (Algebra) und QSQL^p (SQL) spielt bei der Abbildung von QSQL^p nach QA^p eine wesentliche Rolle.

Der Ausgangspunkt für die folgenden Betrachtung ist eine in QSQL^p formulierte Anfrage. Als Grundlage für die Erzeugung eines entsprechenden gleichwertigen QA^p-Ausdrucks wird die Kalkülauswertung einer äquivalenten CQQL^p-Anfrage betrachtet.

In der Kalkülauswertung wird jede Variablenbelegung gegen eine normalisierte Bedingung F ausgewertet. Die Menge aller gebundenen Variablenbelegungen wird hier als R_{VB} bezeichnet. Die eigentlichen Ergebnistupel werden abschließend anhand einer Menge von Ausgabeattributen $\mathcal{A}$ gebildet. Übersetzt man dieses Vorgehen direkt in einen Algebraausdruck ergibt sich folgende Grundstruktur für die Auswertung: $\pi_{\mathcal{A}}(\sigma_F(R_{VB}))$.

In dem grundlegenden Algebraausdruck wird die Menge R_{VB} als Eingangsrelation benutzt. Offensichtlich kann diese Relation schnell anwachsen, da sie alle benötigten Variablenbelegungen als Tupel beinhaltet und Projektionen bzw. Selektionen, welche die Eingangsrelation verkleinern würden, erst abschließend durchgeführt werden. Eine direkte Auswertung dieses Ausdruckes ist demnach nicht praktikabel. Bevor im nächsten Abschnitt auf eine notwendige Optimierung eingegangen wird, steht hier zunächst die Generierung der Grundstruktur $\pi_{\mathcal{A}}(\sigma_F(R_{VB}))$ im Vordergrund.

Die übergebene QSQL^p-Anfrage wird hierfür in eine spezielle Datenstruktur, dem sogenannten *Select-From-Where-Baum*, überführt. Er stellt die Grundlage für den Abbildungsalgorithmus zwischen QSQL^p und QA^p dar. Im SFW-Baum wird u.a. die syntaktische Struktur der QSQL^p-Anfrage nachgebildet. Dementsprechend sind die Knoten des Baumes entweder SFW-Blöcke, Relationen oder Relationsoperatoren ($\times, \cup, \cap, -$). Jeder SFW-Block besitzt (1) eine *Projektionsliste*, welche aus der `select`-Klausel generiert wird, (2) eine *logische Bedingung*, welche auf der `where`-Klausel basiert, und (3) Konnektoren zu weiteren möglichen Unterabfragen.

Als Beispiel wird die abstrakte QSQL^p-Anfrage aus Quelltext 4.1 betrachtet. Die Anfrage drückt den Schnitt zweier probabilistischer Tabellen T_1 und T_2 aus, wobei die bereinigten Relationenschemata (ohne Attribut Pr) der benutzten Tabellen $R(T_1) = (A_1, A_2, A_3)$ und $R(T_2) = (B_1, B_2)$ lauten. Die Anfrage beinhaltet u.a. die zwei Ähnlichkeitsbedingungen $B_1 \approx 1$ und $A_1 \approx 1$. Diese be-

```
select A1
from
     ( select A1, A2
       from T1
       where A3 ~ 3 and A2 > 2 )
  intersect
     ( select *
       from T2
       where B1 ~ 1 and B2 ~ 2 )
where A1 ~ 1
```

Quelltext 4.1. Beispielanfrage in QSQL^p

ziehen sich auf ein und dasselbe Attribut, wenn man die geschnittene Relation als Grundlage betrachtet. Diese Überlappung von Ähnlichkeitsprädikaten muss mittels einer Intra-Tupel-Normalisierung aufgelöst werden. Andernfalls wird auf den ersten Attributwert eines jeden Tupels aus T_2 die Bedingung *ähnlich 1* doppelt ausgeführt.

Der für die Beispielanfrage generierte SFW-Baum wird in Abbildung 4 gezeigt.

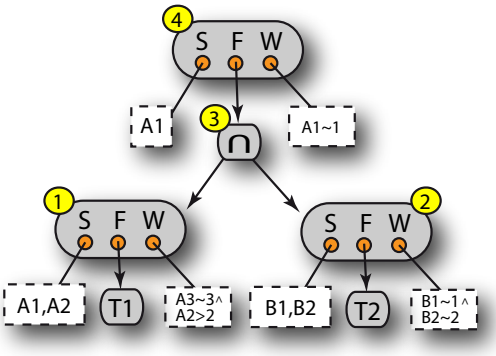

Abb. 4. SFW-Baum der Anfrage aus Quelltext 4.1

Mit Hilfe einer Traversierung des SFW-Baum werden knotenweise die drei Bestandteile der initialen Grundstruktur $\pi_{\mathcal{A}}(\sigma_F(R_{VB}))$ konstruiert, d.h. (1) die Attributmenge $\mathcal{A}$, (2) die Selektionsbedingung F und (3) der Algebraausdruck zur Konstruktion von R_{VB}.

Die Tabelle 2 beinhaltet die drei QA^p-Bestandteile $\mathcal{A}$, F und R_{VB} für die Knoten 1 bis 4 der Beispielanfrage. Die Formeln der beiden Knoten 1 und 2 ergeben sich zu F_1 und F_2. Die Bereichsvariablen X_i stammen aus einem globalen

	$\mathcal{A}$	F	R_{VB}
1	$\mathcal{A}_1 = \{X_1, X_2\}$	$F_1 = T_1^{\approx}(X_1, X_2, X_3) \wedge X_3 \approx 3 \wedge X_2 > 2$	T_1
2	$\mathcal{A}_2 = \{X_4, X_5\}$	$F_2 = T_2^{\approx}(X_4, X_5) \wedge X_4 \approx 1 \wedge X_5 \approx 2$	T_2
3	$\mathcal{A}_3 = \{X_1, X_2\}$	$F_3 = (T_1^{\approx}(X_1, X_2, X_3) \wedge X_3 \approx 3 \wedge X_2 > 2) \wedge$ $(T_2^{\approx}(X_1, X_2) \wedge X_1 \approx 1 \wedge X_2 \approx 2)$	$T_1 \cap_{(\mathcal{A}_1, \mathcal{A}_2)} T_2$
4	$\mathcal{A}_4 = \{X_1\}$	$F_4 = ((T_1^{\approx}(X_1, X_2, X_3) \wedge X_3 \approx 3 \wedge X_2 > 2) \wedge$ $(T_2^{\approx}(X_1, X_2) \wedge X_1 \approx 1 \wedge X_2 \approx 2)) \wedge X_1 \approx 1$	$T_1 \cap_{(\mathcal{A}_1, \mathcal{A}_2)} T_2$

Tabelle 2. Berechnung des initialen Grundausdruckes

Variablenschemata und repräsentieren die jeweiligen Attribute der zu Grunde liegenden Relationen T_1 und T_2. Die Relationen T_1 und T_2 wiederum erzeugen die probabilistischen Relationenprädikate $T_1^{\approx}$ und $T_2^{\approx}$. Sie werden genutzt um die entsprechenden Eintrittswahrscheinlichkeiten einfließen zu lassen. Logische Bedingungen aus der `where`-Klausel werden konjunktiv an die jeweiligen probilistischen Relationenprädikate gebunden.

Die beiden Zwischenformeln F_1 und F_2 werden in Knoten 3 zu der Formel F_3 kombiniert. Der Schnittoperator kann dabei direkt in eine Konjunktion zwischen F_1 und F_2 umgewandelt werden, wobei die beiden Variablenschemata einander angepasst werden müssen. Dadurch können äußere Bedingungen (hier: $A1 \approx 1$) auf die Tupel beider Eingangsrelationen wirken.

Die Attributmengen $\mathcal{A}_i$ ergeben sich direkt aus den Projektionsattributlisten der entsprechenden SFW-Blöcke. Der Algebraausdruck zur Berechnung von R_{VB} wird entsprechend den Abbildungsvorschriften aus [12] generiert.

Der initiale QA^p-Ausdruck ergibt sich dann zu $\pi^i_{(\{X_1\},F)}(T_1 \cap_{(\mathcal{A}_1,\mathcal{A}_2)} T_2)$, wobei $F = \mathtt{norm}_{intra}(F_4) = T_1^{\approx}(X_1, X_2, X_3) \wedge T_2^{\approx}(X_1, X_2) \wedge X_1 \approx 1 \wedge X_2 \approx 2 \wedge X_2 > 2 \wedge X_3 \approx 3$. In F ist nun die Überlappung der Ähnlichkeitsprädikate aufgelöst, da $B_1 \approx 1$ und $A_1 \approx 1$ jeweils auf $X_1 \approx 1$ abgebildet und mittels der Idempotenz-Regel zusammengefasst worden sind.

4.4 Optimierung in QA^p

Um ein starkes Anwachsen von R_{VB} zu vermeiden, muss der initiale Grundausdruck optimiert werden. Die Optimierung von QA^p-Ausdrücken setzt die Möglichkeit einer separaten Normalisierung von Teilausdrücken voraus. Dies bedeutet, dass zwischen zu trennenden Teilausdrücken keine Überlappungen von Ähnlichkeitsprädikaten existieren dürfen, die aufgelöst werden müssten.

Ein optimierter QA^p-Ausdruck kann extensionale, sowie intensionale Operatoren beinhalten. Ziel der Optimierung ist es einen Ausdruck zu erzeugen der möglichst auf die Anwendung von intensionalen Operatoren verzichtet, da diese einen internen Normalisierungsschritt (siehe Tabelle 1) notwendig machen. Zur Verdeutlichung des Optimierungspotential, soll ein optimierter Ausdruck für das eingeführte Beispiel in Abbildung 5 genutzt werden. Der optimierte Ausdruck enthält nur noch extensionale Operatoren. Die Normalisierung der Ausgangs-

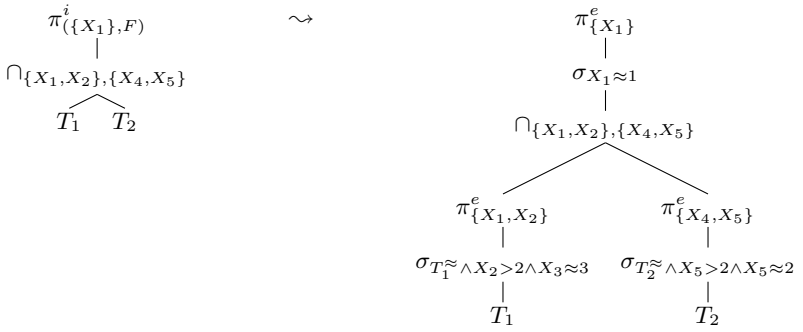

Abb. 5. Optimierung des initialen Algebrabaumes

anfrage verschiebt sich auf die gezielte Anwendung extensionaler Algebraoperatoren und den Einsatz entsprechender Selektionsbedingungen. Die konzeptionelle Konstruktion von R_{VB} vereinfacht sich durch den Einsatz extensionaler Projektionen zu einer einfachen Schnittoperation, wenn man den relationalen Datenanteil des Operators $\cap_{\{X_1,X_2\},\{X_4,X_5\}}$ (siehe Tab. 1) als natürlichen Verbund zwischen zwei Relationen mit gleichen Relationenschemata auflöst. Damit gleicht der erzeugte Ausdruck stark der ursprünglichen QSQL^p-Anfrage. Der gewonnene Effekt neben der Wahrscheinlichkeitsberechnung ist die Normalisierung der überlappenden Ähnlichkeitsprädikate $B_1 \approx 1$ und $A_1 \approx 1$, welche beide auf $X_1 \approx 1$ abgebildet worden sind.

5 Experimente

Zur Evaluierung der Performanz wurde das Beispiel aus Quelltext 4.1 mit den unoptimierten und optimierten Ausführungsplänen aus Abbildung 5 untersucht. Zu Grunde lagen zwei Familien von Tabellen T_1, T_2, welche jeweils $10^0, 10^1, \ldots, 10^6$ Tupel enthielten. Zur Überprüfung wurde ein Sun UltraSPARC IV 1.4 GHz mit 8 GB RAM genutzt. Bei Experiment 1 enthielt Tabelle T_1 konstant 10^4 Tupel. Wie in Abbildung Tabelle 3 zu erkennen ist, wächst die Laufzeit der nicht optimierten Anfrage linear mit der Größe von T_2, während die optimierte Anfrage deutlich weniger Zeit benötigt.

Bei Experiment 2 wuchsen beide Tabellen T_1 und T_2. In Tabelle 4 sieht man, dass die Laufzeit des optimierten Verfahrens in diesem Fall linear wächst, während die Laufzeit des nicht optimierten Verfahrens quadratisch wächst.

Das nicht optimierte Verfahren ist zwar semantisch korrekt, aber zu langsam. Das äquivalente optimierte Verfahren ist also trotz seiner benötigten komplexeren Konstruktion bei Anfragen auf große Tabellen zu bevorzugen.

6 Vergleichbare Ansätze

In der Literatur wurden eine Vielzahl von Systemen vorgeschlagen, welche die probabilistische Verarbeitung von relationalen Daten unterstützen. In dem Kon-

Anfragen	**Anzahl Tupel in T_2**						
	10^0	10^1	10^2	10^3	10^4	10^5	10^6
optimiert	0,5	0,5	1,5	1,5	2,0	9,0	82,3
nicht optimiert	0,5	0,6	3,5	30,1	297,3	-	-

Tabelle 3. Auswertungszeit in Sekunden bei 10^4 Tupel in T_1

Anfragen	**Anzahl Tupel in T_1 = Anzahl Tupel in T_2**						
	10^0	10^1	10^2	10^3	10^4	10^5	10^6
optimiert	0,5	0,5	0,5	0,5	2,0	16,8	162,7
nicht optimiert	0,5	0,5	0,5	3,5	297,3	-	-

Tabelle 4. Auswertungszeit in Sekunden

text von QSQL^p sollen vor allem Ansätze untersucht werden, die eine *logikbasierte Anfragesprache* in Form eines Kalküls, einer Algebra oder eines SQL-Dialektes anbieten.

Die betrachteten Systeme können bezüglich der Wahrscheinlichkeitsberechnung grob in zwei Klassen eingeteilt werden: extensionale und intensionale Ansätze. Die konzeptionellen Charakteristika von extensionalen und intensionalen Verfahren werden in [16] umfassend diskutiert.

Extensionale Systeme [3,2,5,4] können sehr effizient Wahrscheinlichkeiten berechnen, wenn die unterstützte Klasse von Anfragen oder die Klasse der verwendeten Wahrscheinlichkeitsmaße eingeschränkt wird.

Zum Beispiel nehmen Cavallo und Pittarelli in [3] an, dass Tupel in derselben Relation disjunkte Ereignisse darstellen. Barbara et. al. [2] verallgemeinern dieses Modell, sodass Tupel unabhängig und deren Attribute zusätzlich ungenau sein können, was zu disjunkten Eintrittswahrscheinlichkeiten auf Attributebene führt. Dabei muss jede Relation eine Menge von deterministischen Attributen besitzen, welche den Schlüssel der Relation bilden. Dey und Sarkar [5] verbessern dieses Modell, indem beliebige Schlüssel erlaubt werden. Es sind jedoch nur Projektionen erlaubt, welche auch den jeweiligen Schlüssel der angefragten Relation enthalten. In [4] wird für die Klasse der konjunktiven Anfragen ohne Selbstverbund sichere (d.h. semantisch korrekte) Ausführungspläne erzeugt. Die Ergebnisse von unsicheren Ausführungspläne werden approximativ angenähert. Keines dieser Systeme kann somit mit *beliebigen* Anfragen korrekt umgehen, da eine notwendige Normalisierung innerhalb des Auswertungsprozesses nicht durchgeführt wird.

QSQL^p berechnet für beliebige Anfragen korrekte Wahrscheinlichkeiten. Bezüglich der einsetzbaren Wahrscheinlichkeitsmaße ist es jedoch z.B. gegenüber [2,8,21] restriktiver, da momentan keine *disjunkte* Eintrittswahrscheinlichkeiten auf Tupel- bzw. Attributebene unterstützt werden. Dieser Nachteil wird in [11] konzeptionell aufgehoben und soll in einer späteren Version von QSQL^p umgesetzt werden.

Im Gegensatz zu extensionalen Ansätzen verarbeiten intensionale Systeme [6,8,21] während der Ergebnisberechnung Ereignisse oder Zufallsvariabeln. Ab-

schließend wird auf der Grundlage des finalen, normalisierten Ereignisses die eigentliche Ergebniswahrscheinlichkeit ermittelt. Dies garantiert wie in QSQL^p die Berechnung von semantisch korrekten Ergebniswahrscheinlichkeiten. Für intensionale Systeme wurden verschiedene Approximationsverfahren entwickelt um die Wahrscheinlichkeitsberechnung auf Kosten der Ergebnisgenauigkeit zu beschleunigen [15,17].

6.1 Logikbasierte Ähnlichkeitsbedingungen in probabilistischen Datenbanken

Neben der Art der Berechnung der Wahrscheinlichkeiten (extensional oder intensional) stellt sich vor allem die Frage der Ausdruckskraft bereits existierender Ansätze: Inwiefern ist es möglich in ihnen *beliebige logikbasierte Ähnlichkeitsanfragen* zu formulieren?

Insbesondere die wegweisenden Arbeiten [6] und [4] diskutieren explizit die Einbindung von Ähnlichkeitsprädikaten. Zur Abgrenzung gegenüber QSQL^p soll deshalb auf diese beiden Ansätze im Detail eingegangen werden.

Ähnlichkeitsprädikate als Built-In-Prädikate Fuhr und Röllecke schlagen in [6] vor die Scoring-Funktion eines Ähnlichkeitsprädikates mit Hilfe einer eigenständigen probabilistischen Relation zu modellieren. Diese wird dann gemäß der ursprünglichen Anfragestruktur mittels einer Verbundoperation in den Anfrageausdruck integriert. Als Bespiel soll folgende Anfrage betrachtet werden: *Bestimme alle Vogelarten, welche in der Nähe von Berlin beobachtet worden sind.* Für das Ähnlichkeitsprädikat Ort_1 *in der Nähe von* Ort_2 wird die probabilistische Relation SF_{OV} (Scoring-Funktion für Ortsvergleich) mit dem Relationenschemata (Ort_1, Ort_2, Pr) und der Tupelmenge $SF_{OV} = \{(\text{Cottbus}, \text{Berlin}, 0.7), (\text{Berlin}, \text{Berlin}, 1.0)\}$ vereinbart. Die Tupel beinhalten die Auswertung der Ortsvergleiche zwischen Cottbus und Berlin, sowie Berlin und Berlin.

Der PRA-Algebraausdruck (siehe [6]) für die Beispielanfrage lautet: $VBeob \bowtie_{Ort=Ort_1} \sigma_{Ort_2=\text{Berlin}}(SF_{OV})$. Somit werden die Eintrittswahrscheinlichkeiten der Tupel aus *VBeob* mit dem jeweiligen Ähnlichkeitswert des Ortsvergleichs aus SF_{OV}verbunden.

Problematisch bei diesem Vorgehen ist jedoch die Konstruktion von SF_{OV}. Sie verkörpert zwar ein Ähnlichkeitsprädikat, aber bezüglich der Auswertung stellt sie kein eigenständiges Konzept dar. Vielmehr unterliegt sie den gleichen Regeln, wie sie für alle probabilistische Relationen gelten. Somit müssen die Tupel unabhängige Basisereignisse darstellen damit die entsprechenden Aggregationsfunktionen angewendet werden können. Die Unabhängigkeit der Tupel ist in einer *SF*-Relation jedoch nicht gegeben. Fuhr und Röllecke schlagen deshalb vor, lediglich Anfragen zu benutzen, in denen keine Tupel aus gleichen *SF*-Relationen kombiniert werden. So darf z.B. eine bestimmte *SF*-Relation nicht mehr als einmal in einem Anfrageausdruck eingebunden werden und Projektionen können nicht mehr beliebig eingesetzt werden.

QSQL^p besitzt bezüglich der Anwendung von Ähnlichkeitsprädikaten mit unterschiedlichen Vergleichskonstanten eine vergleichbare Restriktion (siehe Kapitel 2.1), jedoch sind z.B. Projektionen innerhalb einer Anfrage *beliebig* anwendbar.

Ähnlichkeitsprädikate als Eintrittswahrscheinlichkeiten von Datenrelationen In [6] wurden Ähnlichkeitsprädikate als probabilistische Relationen modelliert, welche *während* des Auswertungsprozess eingebunden werden. Im Gegensatz dazu schlagen Dalvi und Suciu in [4] vor, die Wahrscheinlichkeiten für die verwendeten Ähnlichkeitsprädikate *vor* der eigentlichen Anfrageauswertung zu ermitteln. Die Ergebnisse dieser Vorberechnungen werden dann den Datenrelationen, auf welche sich die jeweilige Ähnlichkeitsprädikate beziehen direkt als Eintrittswahrscheinlichkeiten zu gewiesen. Zur Verdeutlichung sollen die bereits eingeführten Tabellen *VArt* und *VBeob* dienen, wobei die Tabelle *VBeob* hier ohne die Spalte *Pr* betrachtet wird (notiert als *VBeob*$'$). Somit besitzen beide Relationen keine individuellen Eintrittswahrscheinlichkeiten.

Es soll folgende Beispielanfrage betrachtet werden: *Bestimme alle Vogelarten, welche in der Nähe von Berlin beobachtet worden sind und möglichst ähnlich einem Vorgabebild sind.* Als Algebraausdruck kann die Anfrage wie folgt formuliert werden: $\pi_{Art}(\sigma_{(Bild \approx_{BV} \text{VBild} \wedge Ort \approx_{OV} \text{Berlin})}(VArt \bowtie \text{VBeob}'))$. Bevor dieser Algebraausdruck ausgewertet wird, werden die Ähnlichkeitsprädikate $Bild \approx_{BV}$ VBild bezüglich der Tupel in *VArt* und das Ähnlichkeitsprädikat $Ort \approx$ Berlin bezüglich der Tupel in *VBeob*$'$ berechnet. Die Ergebnisse werden als Eintrittswahrscheinlichkeiten in die Tabellen *VArt* und *VBeob*$'$ kodiert. Die Tabellen *VArt* und *VBeob*$'$ werden somit zu den probabilistischen Relationen *VArt*^{p} und *VBeob*^{p}. Der auszuwertendende Ausdruck ergibt sich dann zu $\pi_{Art}(VArt^p \bowtie VBeob^p)$.

Da in einer Verbundoperation die Wahrscheinlichkeiten für die zu verbindenden Tupel beider Relationen konjunktiv verknüpft werden [4], ergibt sich in der Ergebnisrelation die erwartete Wahrscheinlichkeit für die Konjunktion $Bild \approx_{BV} VBild \wedge Ort \approx_{OV} Berlin$.

Dieser Mechanismus funktioniert jedoch lediglich bei Anfragen mit konjunktiv verknüpften Ähnlichkeitsprädikaten. Bereits bei einer einfachen Disjunktion von Ähnlichkeitsprädikaten, welche sich jeweils auf verschiedene Relationen beziehen, ist es nicht mehr möglich die Auswertung der disjunktiven Ähnlichkeitsbedingung aufzuteilen und in die jeweiligen Relationen zu verschieben. Beispielhaft soll folgende Anfrage betrachtet werden: *Bestimme alle Vogelarten, welche in der Nähe von Berlin beobachtet worden sind* oder *möglichst ähnlich einem Vorgabebild sind.* Der entsprechende Algebraausdruck ist nun gegeben durch: $\pi_{Art}(\sigma_{(Bild \approx_{BV} \text{VBild} \vee Ort \approx_{OV} \text{Berlin})}(VArt \bowtie VBeob'))$.

Ein Verschieben der Ähnlichkeitsprädikate in ihre jeweiligen Relationen steht der Widerspruch zwischen ihrer disjunkten Verknüpfung in der Selektion und der konjunktiven Verknüpfung von Wahrscheinlichkeiten innerhalb der Verbundoperation entgegen.

In weiteren Ansätzen (z.B. [21] und [8]) können Eintrittswahrscheinlichkeiten auch auf Attributebene modelliert werden. Somit besteht hier die Option die Auswertung der Ähnlichkeitsprädikate direkt in den abgefragten Attributen zu kodieren, bevor die eigentliche Anfrageauswertung gestartet wird. Dies funktioniert jedoch wiederum nur bei konjunktiv verknüpften Ähnlichkeitsprädikaten, da die Wahrscheinlichkeit für ein Tupel konjunktiv aus den einzelnen Eintrittswahrscheinlichkeiten seiner Attributwerte gebildet wird.

Zusammenfassend kann festgestellt werden, dass im Gegensatz zu QSQL^p in den diskutierten Ansätzen [6], [4], [8] und [21] eine Integration *beliebiger logikbasierter Ähnlichkeitsbedingungen* nicht gegeben ist.

6.2 Fuzzy Datenbanken

Fuzzy Datenbanken (z.B. [7]) können ebenfalls mit unsicheren Anfragen auf unsicheren Daten umgehen. Es handelt sich hier jedoch nicht um ein probabilistisches Anfragemodell. Vielmehr werden die hier verwendeten Tupel-Zugehörigkeitswerte, ähnlich wie bei extensionalen probabilistischen Systemen, aggregiert ohne die eigentliche Semantik der kombinierten Teilbedingungen zu berücksichtigen. Das Konzept einer semantischen Normalisierung ist unbekannt.

Des Weiteren stellt die zu Grunde liegende Fuzzy Logik [22] im Allgemeinen keine Boolesche Algebra dar. Bekannte logische Äquivalenzen und Transformationsregeln (z.B. Idempotenz und Distributivität) sind somit nicht gültig. Ein detaillierter Vergleich zwischen Fuzzy Logik und Quantenlogik wird in [19] präsentiert.

7 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde die etablierte Many-World-Sematik für probabilistische Datenbanken um das Konzept der Relevanzwahrscheinlichkeiten erweitert. Diese werden in Form von logikbasierten Ähnlichkeitsanfragen auf einer unsicheren Datengrundlage formuliert. Neben der konzeptionellen Kombination beider Anfrageparadigmen wurde mit den Ähnlichkeitsanfragesprachen CQQL^p, QA^p und QSQL^p eine praktische Umsetzung diskutiert. Des Weiteren wurde aufgezeigt, dass bisherige Ansätze beliebige logikbasierte Anfragen nicht ausreichend unterstützen. Als zukünftiges Forschungsvorhaben ist die Erweiterung des hier entwickelten probabilistischen Anfragemodells um disjunktive Eintrittswahrscheinlichkeiten auf Tupel- und Attributebene zu nennen.

Literatur

1. Serge Abiteboul, Paris C. Kanellakis, and Gösta Grahne. On the Representation and Querying of Sets of Possible Worlds. In Umeshwar Dayal and Irving L. Traiger, editors, *SIGMOD Conference*, pages 34–48. ACM Press, 1987.
2. Daniel Barbará, Hector Garcia-Molina, and Daryl Porter. The management of probabilistic data. *IEEE Trans. Knowl. Data Eng.*, 4(5):487–502, 1992.

3. Roger Cavallo and Michael Pittarelli. The theory of probabilistic databases. In Peter M. Stocker, William Kent, and Peter Hammersley, editors, *VLDB*, pages 71–81. Morgan Kaufmann, 1987.
4. Nilesh Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. *The VLDB Journal The International Journal on Very Large Data Bases*, 16(4):523–544, October 2007.
5. Debabrata Dey and Sumit Sarkar. A probabilistic relational model and algebra. *ACM Trans. Database Syst.*, 21(3):339–369, 1996.
6. Norbert Fuhr and Thomas Roelleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. Inf. Syst.*, 15(1):32–66, 1997.
7. Jose Galindo, Angelica Urrutia, and Mario Piattini. *Fuzzy Databases: Modeling, Design and Implementation*. Idea Group Publishing, Hershey, USA, 2006.
8. Christoph Koch. MayBMS: A System for Managing Large Uncertain and Probabilistic Databases. In *Managing and Mining Uncertain Data, chapter 6*. Springer-Verlag, 2008.
9. Sebastian Lehrack. The Probabilistic Similarity Calculus CQQL^p. Technical report, Brandenburgische Technische Universität Cottbus, Institut für Informatik, Cottbus, Germany, 2010.
10. Sebastian Lehrack. The Retrieval Model Behind CQQL. Technical report, Brandenburgische Technische Universität Cottbus, Institut für Informatik, 2010.
11. Sebastian Lehrack. A Unifying Probability Measure for Logic-Based Similarity Conditions on Uncertain Relational Data. Technical report, Brandenburgische Technische Universität Cottbus, Institut für Informatik, Cottbus, Germany, 2011.
12. Sebastian Lehrack and Sascha Saretz. The Definition of QA^p. Technical report, Brandenburgische Technische Universität Cottbus, Institut für Informatik, Cottbus, Germany, 2010.
13. Sebastian Lehrack and Ingo Schmitt. QSQL: Incorporating Logic-Based Retrieval Conditions into SQL. In Hiroyuki Kitagawa, Yoshiharu Ishikawa, Qing Li, and Chiemi Watanabe, editors, *DASFAA (1)*, volume 5981 of *Lecture Notes in Computer Science*, pages 429–443. Springer, 2010.
14. Sebastian Lehrack, Ingo Schmitt, and Sascha Saretz. CQQL: A Quantum Logic-Based Extension of the Relation Domain Calculus. In *Proceedings of the International Workshop Logic in Databases (LID '09)*, October 2009.
15. Dan Olteanu, Jiewen Huang, and Christoph Koch. Approximate confidence computation in probabilistic databases. In *ICDE*, pages 145–156, 2010.
16. J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
17. Christopher Re, Nilesh N. Dalvi, and Dan Suciu. Efficient top-k query evaluation on probabilistic data. In *ICDE*, pages 886–895, 2007.
18. Ingo Schmitt. QQL: A DB&IR Query Language. *The VLDB Journal*, 17(1):39–56, 2008.
19. Ingo Schmitt, Andreas Nürnberger, and Sebastian Lehrack. On the Relation between Fuzzy and Quantum Logic. In *Views on Fuzzy Sets and Systems from Different Perspectives, chapter 5*. Springer-Verlag, 2009.
20. C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.
21. J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. In *Proceedings of the Second Biennial Conference on Innovative Data Systems Research (CIDR '05)*, January 2005.
22. Lotfi Asker Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, June 1965.