Optimierung von Ausdrücken der Ähnlichkeitsalgebra $\mathcal{SA}$

Thomas Herstel, Ingo Schmitt
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
PF 4120, 39016 Magdeburg
{herstel|schmitt}@iti.cs.uni-magdeburg.de

Abstract: Die kalkülbasierte Anfragesprache WS-QBE erlaubt eine deklarative Formulierung von ähnlichkeitsbasierten Multimedia-Anfragen. Für die Anfrageauswertung eignet sich hingegen durch ihren prozeduralen Charakter eher eine Algebra. Die Notwendigkeit der algebraischen Optimierung ergibt sich durch die Abbildung von WS-QBE-Anfragen auf die Ähnlichkeitsalgebra $\mathcal{SA}$. Optimierungsregeln der relationalen Algebra sind jedoch wegen ihrer Erweiterung um speziellen Konzepte nicht uneingeschränkt auf die $\mathcal{SA}$ übertragbar. In dieser Arbeit wird in die Optimierungsproblematik eingeführt.

1 Einleitung

Bei der Formulierung von Anfragen in Multimedia-Datenbanksystemen kommen neben den exakten Suchbedingungen sehr häufig Vergleiche bezüglich der Ähnlichkeit zu vorgegebenen Medien-Objekten zum Einsatz. Die daraus resultierenden Ähnlichkeitswerte geben dabei an, wie stark diese Kriterien insgesamt erfüllt wurden. Im Gegensatz zur relationalen Algebra wird durch den Ähnlichkeitsgrad implizit jedem Tupel ein Zugehörigkeitswert zur Ergebnismenge zugeordnet.

Für eine effiziente Anfrageausführung ist eine Anfrageoptimierung notwendig. Im Folgenden werden wir die Ähnlichkeitsalgebra $\mathcal{SA}$ vorstellen und danach in die Problematik der algebraischen Optimierung in dieser Algebra einführen. Wir präsentieren jedoch keine fertigen Optimierungsregeln, sondern diskutieren grundlegende Probleme im Vergleich zur klassischen, relationalen Optimierung.

2 Die Ähnlichkeitsalgebra $\mathcal{SA}$

In diesem Abschnitt soll die Ähnlichkeitsalgebra $\mathcal{SA}$ skizziert werden. Für eine ausführliche, formale Beschreibung verweisen wir auf die Arbeiten [SS04, Sch04].

Die Ähnlichkeitsalgebra $\mathcal{SA}$ erweitert die relationale Algebra um die Behandlung von aus resultierenden Ähnlichkeitsbedingungen Zugehörigkeitswerten aus dem Intervall $[0, 1]$. Aufgrund solcher Zugehörigkeitswerte müssen statt der booleschen Junktoren Konjunkti-

on und Disjunktion Fuzzy-Junktoren (T-Norm und T-Conorm [Zad88]) eingesetzt werden. Zusätzlich erlaubt die Algebra Gewichtungen, wobei das Gewichtungsschema von Fagin und Wimmers [FW00] übernommen und an unsere Algebra angepasst wurde [SS03].

Die Integration von Zugehörigkeitswerten in $\mathcal{SA}$ erfordert eine gesonderte Behandlung dieser Werte in den einzelnen Operationen. Die Attributposition 0 ist daher in jeder Relation für den speziellen Zugehörigkeitswert reserviert. Im folgenden werden einige Operationen der Ähnlichkeitsalgebra $\mathcal{SA}$ vorgestellt.

Die Operation $\pi_{\#_{p_1},\#_{p_2},\ldots,\#_{p_n}}(E)$ führt eine **Projektion** des Algebraausdrucks E auf die Attribute mit den Positionen $\#_{p_1}, \#_{p_2}, \ldots, \#_{p_n}$ durch. Bei der Duplikateliminierung werden die jeweiligen Duplikattupel mit ihren Zugehörigkeitswerten disjunktiv verknüpft. Die **Selektion** wird mit $\sigma_{y_i \delta y_j}(E)$ notiert. Das Selektionsprädikat kann dabei eine Ähnlichkeitsbedingung beinhalten. Der berechnete Ähnlichkeitswert wird konjunktiv mit dem Zugehörigkeitswert des jeweiligen Tupels aus E verbunden. Für die **Mengenvereinigungen** gibt es drei verschiedene Varianten $(E_1 \cup E_2)$, $(E_1 \overrightarrow{\cup}_\theta E_2)$ und $(E_1 \overleftarrow{\cup}_\theta E_2)$, wobei die Algebraausdrücke E_1 und E_2 vereinigungsverträglich sein müssen. Die korrespondierenden Tupel werden disjunktiv (T-Conorm) miteinander verknüpft. Bei der stärker rechts $(E_1 \overrightarrow{\cup} E_2)$ bzw. links $(E_1 \overleftarrow{\cup} E_2)$ gewichteten Variante der Vereinigung wird eine gewichtete Disjunktion entsprechend dem Gewichtungsschema aus [FW00] eingesetzt. Die drei **Schnittoperatoren** $(E_1 \cap E_2)$, $(E_1 \overrightarrow{\cap}_\theta E_2)$ und $(E_1 \overleftarrow{\cap}_\theta E_2)$ sind analog zu den Vereinigungen definiert sind. Statt der Disjunktion wird jedoch die Konjunktion eingesetzt. Der **Universal-Operator** wird mit $(E_1 \uplus^z E_2)$ bzw. $(E_1 \overrightarrow{\uplus}^z_\theta E_2)$ und $(E_1 \overleftarrow{\uplus}^z_\theta E_2)$ bezeichnet. Wenn der Schnitt zu restriktiv und eine Vereinigung zu wenig restriktiv ist, kann mit dem Universal-Operator durch Variieren des Wertes $z \in [0,1]$ die Semantik zwischen diesen beiden Extremen eingestellt werden. Die konkrete Semantik ergibt sich aus der Linearkombination der zugrunde liegenden Konjunktion und Disjunktion. Die **Differenz** $(E_1 \setminus E_2)$ wird mittels einer Konjunktion und eine Negation aus den entsprechenden Zugehörigkeitswerten berechnet. Das **kartesische Produkt** $(E_1 \times E_2)$ und der **Verbund** $(E_1 \bowtie_{\#_{i_1}=\#_{j_1},\ldots,\#_{i_m}=\#_{j_m}} E_2)$ sind analog zu den entsprechenden relationalen Operationen definiert, wobei jedoch die Zugehörigkeitswerte konjunktiv verknüpft werden.

Auf den Abriss weiterer Operationen, wie dem Schwellwertoperator oder der fuzzifizierten Projektion muss an dieser Stelle aus Platzgründen verzichtet werden.

Die Ähnlichkeitsalgebra $\mathcal{SA}$ ist keine nutzerfreundliche Sprache. Eine bessere Nutzerunterstützung bei der Anfrageformulierung ist die QBE-Sprache WS-QBE [SSH04, Sch04]. WS-QBE-Anfragen werden für eine effiziente Ergebnisberechnung in $\mathcal{SA}$-Anfragen nach dem in [SS04] beschriebenen Verfahren umgewandelt. Unsere Ähnlichkeitsalgebra ist damit eine Zielsprache für die Abbildung ausgehend von WS-QBE-Anfragen und eine Sprache, in der die Optimierung statt findet.

Nachfolgend ist eine WS-QBE-Anfrage gegeben, die alle zu einem Vorgabebild ähnlichen Ölgemälde von einem holländischen Maler bestimmt. Das Ergebnis der Transformation in die Ähnlichkeitsalgebra ist in Abbildung 1 links dargestellt.

Gemälde	Id	Photo	Maler	Titel	Technik
	P.	~ 	_kid		Öl

Künstler	kid	Name	Land
	_kid		Holland

Um die Ergebnistupel wie üblich nach ihren Zugehörigkeitswerten sortieren zu können, gehen wir von einem entsprechenden Sortieroperator ω aus, der auf dem Ergebnis einer Ähnlichkeitsanfrage angewendet wird. Zur Beschränkung grosser Ergebnismengen ist eine `getNext`-Semantik wichtig, bei der die Tupel durch den Nutzer jeweils nacheinander explizit angefordert werden.

3 Ausgewählte Optimierungsaspekte

Das Prinzip der algebraischen Optimierung von Ausdrücken ist allgemein, einen Ausdruck in einen algebraisch äquivalenten umzuformen. Da jeder Ausdruck eine prozedurale Berechnungsvorschrift darstellt, ist es das Ziel, einen Ausdruck zu finden, der mit weniger Berechnungsaufwand zum selben Ergebnis führt. Dazu werden Transformationsregeln angegeben, die einen gegebenen Ausdruck in einen garantiert äquivalenten Ausdruck umwandeln. Da es zu einem Ausdruck unendlich viele äquivalente Ausdrücke gibt, sollen Heuristiken die Transformationsregeln und die Reihenfolge ihrer Anwendung steuern.

Die algebraische Äquivalenz muss bei der $\mathcal{SA}$ zusätzlich zu den Tupeln auch die Übereinstimmung ihrer Zugehörigkeitswerte garantieren. Die besondere Problematik der algebraischen Optimierung zeigen die folgenden Unterscheidungen:

Adaptierte Operatoren: Die Operatoren π, σ, $\times$, $\bowtie$, $\setminus$, $\cap$, $\cup$ unterscheiden sich von den relationalen Operatoren durch die besondere Behandlung der Zugehörigkeitswerte. Diese bedingen den Einsatz einer T-Norm und T-Conorm.

Nutzerpräferenzen: Zur Abbildung von Nutzerpräferenzen werden in der $\mathcal{SA}$ unter anderem gewichtete Operatoren ($\overrightarrow{\cup}_\theta$, $\overleftarrow{\cup}_\theta$, $\overrightarrow{\cap}_\theta$, $\overleftarrow{\cap}_\theta$, $\overrightarrow{\uplus}^z_\theta$, $\overleftarrow{\uplus}^z_\theta$) eingesetzt. Allgemein wird der resultierende Zugehörigkeitswert pro Tupel dabei nicht allein durch T-Norm bzw. T-Conorm bestimmt, sondern zusätzlich durch das Gewicht θ beeinflusst. Bei $\overrightarrow{\cap}$ und $\overleftarrow{\cap}$ können zudem aufgrund der Gewichtungssemantik auch Tupel in der Ergebnismenge auftreten, die aus nur einer Relation stammen.

Die adaptierten Operationen verwenden entweder eine T-Norm `T` (σ, $\times$, $\bowtie$, $\setminus$, $\cap$) oder eine T-Conorm `S` (π, $\cup$). Bei der Berechnung der Zugehörigkeitswerte für die Ergebnistupel gehen Eigenschaften von T-Norm und T-Conorm auf die Algebra-Operatoren über. Bspw. kann allgemein in der relationalen Algebra die Verbundreihenfolge vertauscht werden: $((R_1 \bowtie R_2) \bowtie R_3) \equiv (R_1 \bowtie (R_2 \bowtie R_3))$. In der $\mathcal{SA}$ erfolgt der Verbund analog, wobei der Zugehörigkeitswert μ_r eines Ergebnistupels sich aus den Zugehörigkeitswerten μ_1, μ_2 und μ_3 der Tupel aus R_1, R_2 und R_3 berechnet. Die Äquivalenz der Berechnung $T(T(\mu_1, \mu_2), \mu_3) = T(\mu_1, T(\mu_2, \mu_3))$ folgt aus Assoziativität und Kommutativität der T-Norm.

Eine Besonderheit gibt es bei einer Kombination von Operationen, bei denen Zugehörigkeitswerte durch Verschachtelung von T-Norm und T-Conorm berechnet werden. Sind R_1 und R_2 vereinigungsverträglich, so ist etwa die Äquivalenz $\sigma_{\#1\delta_1 C_1}(R_1 \cup R_2) \equiv$

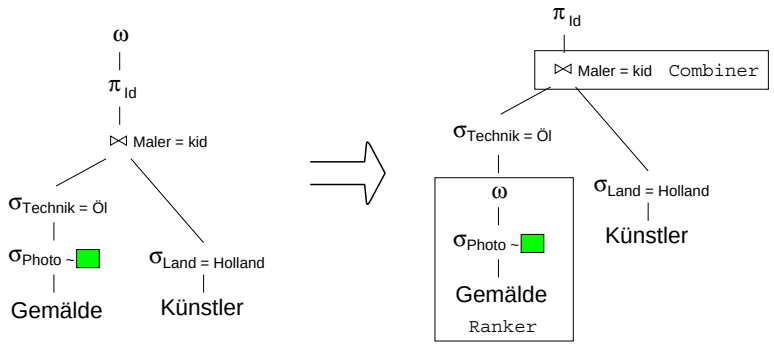

Abbildung 1: Algebrabaum vor (links) und nach der Transformation (rechts)

$\sigma_{\#1\delta_1 C_1}(R_1) \cup \sigma_{\#1\delta_1 C_1}(R_2)$ im relationalen Fall gegeben. In der $\mathcal{SA}$ wird der Gesamtzugehörigkeitswert μ_r aus den Zugehörigkeitswerten μ_1 und μ_2 der Tupel und dem Ähnlichkeitswert μ_s bezüglich der Selektionsbedingung berechnet. Daher ergibt sich $\mu_{r_1} = T(\mu_s, S(\mu_1, \mu_2))$ falls die Selektion nach der Vereinigung ausgeführt wird bzw. umgekehrt $\mu_{r_2} = S(T(\mu_s, \mu_1), T(\mu_s, \mu_2))$. Die Gleichheit von μ_{r_1} und μ_{r_2} ist nicht durch die (allgemeinen) Eigenschaften von T-Norm und T-Conorm garantiert. Die hier notwendige Distributivität verlangt Idempotenz der Funktionen ist ausschließlich für die Funktionen `min` und `max` gegeben. Für die Optimierung ist daher die Beschränkung auf diese Funktionen empfehlenswert.

Eine Heuristik für die Plangenerierung ist bspw., den Sortieroperator so weit wie möglich nach innen zu schieben. Erfolgt die Sortierung unmittelbar nach einer Ähnlichkeitsselektion, können Ranker-Algorithmen [Hen94, HS95] zur effizienten Sortierung eingesetzt werden. Zum einen kann so eine geforderte `getNext`-Semantik unter Umständen als Pipeline direkt bis hin zum Ranking-Algorithmus realisiert werden, wodurch Zwischenergebnisse minimiert werden. Zum anderen bringt eine vorhandene Sortierung mitunter bei der Kombination zweier Ausdrücke durch $\bowtie$, $\cap$, $\times$ oder $\oplus^z$ Effizienzgewinne, insbesondere durch Einsatz von Combiner-Algorithmen [Fag02, FLN03, GBK00, NR99, GBK01].

Das Bewegen des Sortierungsoperators durch den Algebrabaum unseres Beispiels wird in der Abbildung 1 demonstriert. Da nach der Umformung Ausgangsrelation, Ähnlichkeitsvergleich und Sortierung eine direkte Kette bilden, können diese drei Operatoren durch einen geeigneten Ranker-Algorithmus zusammengefasst werden. Für den Verbund liegen die Tupel von links kommend sortiert vor. An der rechten Seite hingegen erscheinen nur Tupel mit dem Zugehörigkeitswert 1. Daher kann der Verbund durch einen Combiner-Algorithmus realisiert werden.

Literatur

[Fag02] R. Fagin. Combining Fuzzy Information: an Overview. In *SIGMOD'02, Proc. of the 2002 ACM SIGMOD Int. Conf. on Management of Data, Madison, Wisconsin, June 3-6, 2002*, Jgg. 31 of *ACM SIGMOD Record*, Seiten 109–118. ACM Press, Juni 2002.

[FLN03] R. Fagin, A. Lotem und M. Naor. Optimal aggregation algorithms for middleware. *Journal of Computer and System Sciences*, 66(4):614–656, 2003.

[FW00] R. Fagin und E. L. Wimmers. A Formula for Incorporating Weights into Scoring Rules. *Special Issue of Theoretical Computer Science*, 2000.

[GBK00] Ulrich Güntzer, Wolf-Tilo Balke und Werner Kießling. Optimizing Multi-Feature Queries for Image Databases. In Amr El Abbadi, Michael L. Brodie, Sharma Chakravarthy, Umeshwar Dayal, Nabil Kamel, Gunter Schlageter und Kyu-Young Whang, Hrsg., *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt*, Seiten 419–428. Morgan Kaufmann, 2000.

[GBK01] Ulrich Güntzer, Wolf-Tilo Balke und Werner Kießling. Towards Efficient Multi-Feature Queries in Heterogeneous Environments. In *International Symposium on Information Technology (ITCC 2001), 2-4 April 2001, Las Vegas, NV, USA*, Seiten 622–628. IEEE Computer Society, 2001.

[Hen94] A. Henrich. A distance-scan algorithm for spatial access structures. In *Proc. of the 2nd ACM Workshop on Advances in Geographic Information Systems, Gaithersburg, Maryland*, Seiten 136–143, dec 1994.

[HS95] G. R. Hjaltason und H. Samet. Ranking in Spatial Databases. In Max J. Egenhofer und John R. Herring, Hrsg., *Advances in Spatial Databases, 4th International Symposium, SSD'95*, Jgg. 951 of *Lecture Notes in Computer Science*, Seiten 83–95, 1995.

[NR99] S. Nepal und M. V. Ramakrishna. Query Processing Issues in Image(multimedia) Databases. In M. Kitsuregawa, Hrsg., *Proc. of the 15th IEEE Int. Conf. on Data Engineering, ICDE'99, Sydney, Australia, March 1999*, Seiten 22–29, Los Alamitos, CA, 1999. IEEE Computer Society Press.

[Sch04] N. Schulz. *Formulierung von Nutzerpräferenzen in Multimedia-Retrieval-Systemen.* Dissertation, Otto-von-Guericke-Universität Magdeburg, Fakultät für Informatik, 2004.

[SS03] N. Schulz und I. Schmitt. Relevanzwichtung in komplexen Ähnlichkeitsanfragen. In G. Weikum, H. Schöning und E. Rahm, Hrsg., *Datenbanksysteme in Business, Technologie und Web, BTW'03, 10. GI-Fachtagung, Leipzig, Februar 2003*, Lecture Notes in Informatics (LNI) Volume P-26, Seiten 187–196, Bonn, 2003. Gesellschaft für Informatik.

[SS04] I. Schmitt und N. Schulz. Similarity Relational Calculus and its Reduction to a Similarity Algebra. In Dietmar Seipel und J. M. Turull-Torres, Hrsg., *Third Intern. Symposium on Foundations of Information and Knowledge Systems (FoIKS'04), Austria, February 17-20*, Jgg. 2942 of *lncs*, Seiten 252–272. Springer-Verlag Berlin Heidelberg, 2004.

[SSH04] I. Schmitt, N. Schulz und T. Herstel. WS-QBE: Eine QBE-Anfragesprache für komplexe Ähnlichkeitsanfragen. *KI - Künstliche Intelligenz: Special Issue Adaptive Multimedia Retrieval*, 2004. erscheint.

[Zad88] Lofti A. Zadeh. Fuzzy Logic. *IEEE Computer*, 21(4):83–93, April 1988.