

Variablen in Algebren? — Die CROQUE–Algebra

Denny Priebe, Joachim Kröger, Andreas Heuer

Universität Rostock
Fachbereich Informatik
Lehrstuhl Datenbank– und Informationssysteme
18051 Rostock

{priebe, jo, heuer}@informatik.uni-rostock.de
<http://www.db.informatik.uni-rostock.de/Forschung/CROQUE.html>

1 Das Projekt CROQUE

Im Rahmen des Projektes CROQUE¹ [KH99] wurden Optimierungsverfahren für objektorientierte Datenbankmanagementsysteme (OODBMS) entwickelt. Im Mittelpunkt der Arbeiten standen dabei Untersuchungen zu Verfahren des “Query–Rewriting” für algebraische und kalkülartige Anfragerepräsentationen, Auswertungsverfahren für objektorientierte Anfragen sowie die Optimierung des physischen Entwurfs von OODBMS bezüglich eines gegebenen Anwendungsprofils.

Ziel dieses Projektes waren Entwurf und Realisierung eines statischen, d.h. zur Übersetzungszeit einer Anfrage arbeitenden Anfrageoptimierers. Die geschaffenen prototypischen Teil–Lösungen, deren Integration und abschließende Evaluierung im Rahmen der restlichen Projektlaufzeit vorgesehen ist, realisieren insgesamt ein objektorientiertes Anfragebearbeitungssystem mit den folgenden drei Hauptmerkmalen:

- einer deskriptiven, objektorientierten Anfrageschnittstelle auf Basis von ODMG–OQL [Cat96],
- physischer Datenunabhängigkeit (d.h. Trennung der logischen Strukturen von den physischen Speicherstrukturen und Zugriffspfaden), die in den bisherigen objektorientierten Datenbanksystemen nicht gegeben ist, und
- einer regel– und kostenbasierten Optimierungskomponente zur Abbildung der deskriptiven Anfrage– und Änderungsoperationen der konzeptuellen Ebene auf einen Ausführungsplan in Abhängigkeit von der gewählten Speicherstruktur.

Für die Verwendung in CROQUE wurden das ODMG–Datenmodell und ODMG–OQL formalisiert und verfeinert [RS97]. Die deklarative, auf Collections aufbauende Anfragesprache CROQUE–OQL basiert auf dem CROQUE–Datenmodell. Zu den Eigenschaften dieses Datenmodells zählen die Realisierung einer Klassenhierarchie sowie die Anordnung aller Objekttypen in einem Typverband. Dadurch sind zum Beispiel objekterhaltende Projektionen auf beliebige Obertypen eines Objektes und Mengenoperationen auf Collections von Objekten immer möglich.

Um sowohl die Vorteile algebraischer als auch kalkülartiger Anfragerepräsentationen während des Optimierungsprozesses zu nutzen, wurde in CROQUE ein hybrider Ansatz der Optimierung [GKG⁺97] gewählt. Hierbei wird eine OQL–Anfrage für die anschließende Anfrageoptimierung

¹CROQUE (Cost– and Rule–based Optimization of object–oriented Queries) ist ein Kooperationsprojekt der Universitäten Rostock und Konstanz. CROQUE wurde bzw. wird von der DFG unter den Aktenzeichen He 1768/5–2 bzw. Scho 554/1–2 gefördert.

jeweils in einen Algebraausdruck und eine Kalküldarstellung [GS96] transformiert. Optimierte Anfragerepräsentationen werden nach C++ übersetzt und auf dem Speichersubsystem Object-Store ausgewertet.

In den folgenden Abschnitten 2 und 3 werden die Vor- und Nachteile variablenbehafteter bzw. variablenfreier Algebren am Beispiel der bisher in CROQUE verwendeten Algebren diskutiert. Anschließend wird in Abschnitt 4 aufgezeigt, welche positiven Auswirkungen die Wahl einer variablenfreien Algebra für die Verwendung in CROQUE auf die Verwaltung und Anwendung von Rewrite-Regeln über Algebra-Ausdrücken hat und welche Möglichkeiten sich daraus für den Optimierungsprozeß von CROQUE ergeben. Wir schließen mit einer kurzen Zusammenfassung.

2 Variablenbehaftete Algebren

Variablenbehaftete Algebren stellen Zusammenhänge zwischen Teilausdrücken über das Binden und Referenzieren von Variablen dar. Der Kontext einer geschachtelten Anfrage wird dabei durch die aktuelle Belegung von in übergeordneten Ausdrücken gebundenen Variablen bestimmt. Diese Möglichkeit, Beziehungen zwischen Teilanfragen auszudrücken, wird im Allgemeinen auch von Anfragesprachen, wie etwa OQL, genutzt. Es ergeben sich daher als positive Eigenschaften variablenbehafteter Algebren, daß Ausdrücke der Anfragesprache im Allgemeinen leicht in eine entsprechende Algebradarstellung übersetzbar und derartige Algebraausdrücke gut lesbar sind. Abbildung 1 zeigt die beispielhafte Übersetzung einer OQL-Anfrage in die bisher in CROQUE verwendete Algebra. Die dichte Beziehung zwischen OQL- und Algebra-Ausdrücken — die Ur-

<pre>select sum(x.Gewicht) from (Schafe union Kühe) as x</pre>	<pre>SUM(LOOP x (U (PATH "Schafe", PATH "Kühe"), PATH "x.Gewicht", LITERAL "true"))</pre>
--	---

Abbildung 1: Übersetzung einer OQL-Anfrage in eine variablenbehaftete Algebra

sache der oben aufgeführten Vorteile — ist aus der Abbildung ersichtlich: die SFW-Klausel wird durch die LOOP-Operation realisiert, Variablen in der OQL-Anfrage können unmittelbar in die Algebra-Darstellung übernommen werden.

Werden jedoch Rewrite-Regeln über Ausdrücken variablenbehafteter Algebren notiert, wirkt sich die Existenz von Variablen nachteilig auf die Möglichkeit der deklarativen Spezifikation dieser Regeln aus. Abbildung 2 zeigt anhand eines Beispiels, daß sowohl explizite Vorbedingungen²

Bedingung:	$Q_1.type = BAG(T) \vee Q_2.type = BAG(T)$
Regel:	$LOOP\ x\ (U(Q_1, Q_2), PATH\ "x", Q_3) \rightarrow$ $U(LOOP\ x\ (Q_1, PATH\ "x", Q_3),$ $LOOP\ y\ (Q_2, PATH\ "y", Q_3))$
Aktion:	Variablenumbenennung im letzten $Q_3 : x \rightarrow y$

Abbildung 2: Verschieben der Selektion

als auch Anweisungen zur Anwendung einer Rewrite-Regel notwendig sein können. Vorbedingungen können etwa die Anwendbarkeit einer Regel aufgrund von Typ-Constraints entscheiden,

²die aufgeführte Vorbedingung resultiert aus der Definition der Operation LOOP und dient der Erhaltung der Kardinalität der Collections

während Anweisungen zur Regelanwendung etwa Variablenumbenennungen bei Einführung neuer Teilausdrücke realisieren.

Ein weiterer Nachteil ist die Eigenschaft, daß im Allgemeinen das Erkennen der linken Seite einer Regel allein durch Pattern-Matching nicht möglich ist. Soll beispielsweise ein Ausdruck unter Ausnutzung der Idempotenz der mengenwertigen Vereinigung vereinfacht werden, kann die Regel nicht mittels $\cup(Q, Q) \rightarrow Q$ formuliert werden, sondern muß wie in Abbildung 3 dargestellt umschrieben werden.

Bedingung: $Q_1.type = SET(T) \wedge Q_2.type = SET(T) \wedge Q_1 \cong Q_2$
 Regel: $\cup(Q_1, Q_2) \rightarrow Q_1$
 Aktion: —

Abbildung 3: Idempotenz der mengenwertigen Vereinigung

Die einfache Übersetzbarkeit und die gute Lesbarkeit variablenbehafteter Algebren stehen daher der Notwendigkeit gegenüber, explizit Vorbedingungen und Anweisungen zur Transformation angeben und den Gültigkeitsbereich von allen Variablen bestimmen zu müssen. Durch die Notwendigkeit der Angabe von Code-Fragmenten sind Rewrite-Regeln für variablenbehaftete Algebren nicht deklarativ.

3 Variablenfreie Algebren

Während variablenbehaftete Algebren Zusammenhänge zwischen geschachtelten Teilausdrücken auf intuitive Art durch die Verwendung von Variablen darstellen, sind variablenfreie Algebren darauf beschränkt, diese Zusammenhänge etwa durch geeignete Komposition von Ausdrücken umzusetzen. Mit der Beseitigung des Konzepts der Variable gehen auch die damit verbundenen Vorteile variablenbehafteter Algebren verloren. Im Vergleich mit variablenbehafteten Algebren sind die Übersetzung von der Anfragesprache in eine variablenfreie Algebra komplexer und die entstehenden Algebraausdrücke schlechter lesbar, da der Bezug auf die Umgebung einer Teilanfrage nicht implizit durch Variablen, sondern explizit durch Operationen der Algebra erfolgen muß. Abbildung 4 zeigt die Repräsentation der OQL-Anfrage aus Abbildung 1 in der variablenfreien CROQUE-Algebra.

```
(sum) o
  ((((((iterate) ((K true)))
    ((Gewicht:'object(Gewicht:int) → int') o (snd))) o
    (((loop) ((K true))) (id))) o
    ((id,(union) o ((K Schafe),(K Kühe))))))
```

Abbildung 4: Ausdruck der variablenfreien CROQUE-Algebra

Durch den Verzicht auf Variablen werden jedoch auch die damit verbundenen Nachteile variablenbehafteter Algebren beseitigt: variablenfreie Darstellungsformen erlauben die deklarative Spezifikation von Rewrite-Regeln, da die Notwendigkeit der Angabe von Code zur Durchführung einer Transformation aufgrund der Beseitigung von Variablen nicht mehr gegeben ist. Auch Vorbedingungen, die in Abbildung 2 noch explizit formuliert wurden, können deklarativ spezifiziert werden. Abbildung 5 zeigt die dort dargestellte Transformation für die variablenfreie

CROQUE–Algebra. Auffällig ist hier die erhöhte Anzahl der zur Durchführung der Transforma-

$$\begin{aligned}
 & (\text{iterate } p \ q) \circ \text{union} \circ \langle f: \text{any} \rightarrow \text{bag}(\text{any}), g \rangle \\
 & \rightarrow \text{union} \circ \langle (\text{iterate } p \ q) \circ f, (\text{iterate } p \ q) \circ g \rangle \\
 \\
 & (\text{iterate } p \ q) \circ \text{union} \circ \langle f, g: \langle \text{any} \rightarrow \text{bag}(\text{any}) \rangle \rangle \\
 & \rightarrow \text{union} \circ \langle (\text{iterate } p \ q) \circ f, (\text{iterate } p \ q) \circ g \rangle
 \end{aligned}$$

Abbildung 5: Regel über einer variablenfreien Algebra

mation notwendigen Regeln. In diesem Beispiel vergrößerte sich die Anzahl der Regeln durch das Aufspalten der disjunktiven Vorbedingung; im Allgemeinen sind jedoch die strukturellen Eigenschaften variablenfreier Algebren Ursache der Vergrößerung der Regelmenge.

Zusammenfassend läßt sich aufführen, daß die komplexe Übersetzung und die schlechte Lesbarkeit variablenfreier Algebren der Möglichkeit der Spezifikation deklarativer Regeln gegenüberstehen. Deklarative Regeln können interpretiert und daher getrennt vom Optimiererkern verwaltet werden. Die Vergrößerung der Regelmenge durch den Einsatz variablenfreier Algebren kann durch geeignete Optimierungskonzepte (wie sie im Rahmen von CROQUE bereits entwickelt wurden [KPH98]) zumindest teilweise kompensiert werden.

4 Algebra und Regelverwaltung in CROQUE

Aufgrund der Vorteile variablenfreier Algebren wurde die bisher in CROQUE verwendete variablenbehaftete logische Algebra in [Pri98] durch eine variablenfreie Algebra abgelöst, wodurch die Spezifikation deklarativer Regeln für den Optimierungsprozeß ermöglicht wurde. Dies war für die variablenbehaftete Algebra zuvor nicht möglich.

Die variablenfreie CROQUE–Algebra, deren Vollständigkeit bezüglich der formalisierten Anfragesprache CROQUE–OQL durch Angabe einer Abbildungsfunktion von der Anfragesprache in die Algebra nachgewiesen wurde, basiert in ihrer Grundkonzeption auf KOLA [CZ96]. Dennoch war eine einfache Adaption von KOLA aufgrund der durch das CROQUE–Datenmodell gegebenen Anforderungen nicht möglich.

Da die variablenfreie Algebra lediglich der internen Anfragerepräsentation dient, überwiegen die gewonnenen Vorteile der einfacheren Handhabbarkeit bzw. Modifizierbarkeit von Regeln sowie die Beseitigung der Notwendigkeit, Gültigkeitsbereiche von Variablen zu untersuchen, die Nachteile der schlechteren Lesbarkeit und der komplexeren Übersetzung, die sich aus den strukturellen Differenzen zwischen Anfragesprache und Algebra ergeben. Die CROQUE–Algebra ist zunächst minimal in der Menge ihrer Operatoren. Eine Einführung weiterer, redundanter Operatoren ist möglich und würde das zudem Optimierungspotential weiter erhöhen.

Die für die bisherige Algebra aufgestellten Regeln wurden bei der Umstellung auf die neue Algebra vom eigentlichen Optimierer entkoppelt und können nun auch mittels einer Regelverwaltung in den Optimierer eingebracht werden, so daß eine Veränderung der Regelmenge keine Modifikationen des Optimierers mehr erfordert.

Eine Anwendung findet die externe Verwaltung von Regeln bei der Integration physischer Substitutionen in den Prozeß der Anfrageauswertung. Physische Substitutionen werden von der flexiblen internen Ebene von CROQUE produziert und beschreiben die tatsächlich vorliegenden internen Speicherstrukturen. Diese Rekonstruktionsvorschriften können in Form von OQL–Anfragen und damit als Algebra–Ausdrücke formuliert werden, so daß sich die Berücksichtigung

der auf der internen Ebene vorhandenen Speicherstrukturen nahtlos in den Optimierungsprozeß einfügt.

5 Zusammenfassung

Die im Projekt CROQUE vorgenommene Einführung einer variablenfreien Algebra gestattet die deklarative Spezifikation von Rewrite-Regeln und ermöglicht damit die Interpretation und externe Verwaltung dieser Regeln. Dadurch ist es möglich, dem Optimierer erst zur Optimierungszeit Regeln bekanntzugeben, wie dies etwa durch die Integration physischer Substitutionen in den Prozeß der Anfrageauswertung gefordert wird.

Es wurde ein Prototyp realisiert, der Ausdrücke der CROQUE-Anfragesprache in die variablenfreie Algebra abbildet, extern gespeicherte Regeln anwendet und die Ergebnisse dieser Transformationen auswertet.

Literatur

- [Cat96] Cattell, R. (Hrsg.): *The Object Database Standard: ODMG-93, Release 1.2*. Morgan-Kaufmann, San Mateo, CA, 1996.
- [CZ96] Cherniack, M.; Zdonik, S.: Rule Languages and Internal Algebras for Rule-Based Optimizers. In: *Proc. of the ACM SIGMOD Int. Conference on Management of Data, Montreal, Quebec*, Juni 1996.
- [GKG⁺97] Grust, T.; Kröger, J.; Gluche, D.; Heuer, A.; Scholl, M. H.: Query Evaluation in CROQUE — Calculus and Algebra Coincide. In: *Proc. of the 15th British National Conference on Databases (BNCOD 15), London, UK, LNCS 1271, Springer*, S. 84–100, Juli 1997.
- [GS96] Grust, T.; Scholl, M. H.: Translating OQL into Monoid Comprehensions — Stuck with Nested Loops? Technischer Bericht 3/1996, Department of Mathematics and Computer Science, Database Research Group, University of Konstanz, März 1996.
- [KH99] Kröger, J.; Heuer, A.: Optimierung objektorientierter Anfragen: Das Projekt CROQUE. Technischer Bericht, Fachbereich Informatik, Universität Rostock, März 1999. Eingereicht zur Veröffentlichung.
- [KPH98] Kröger, J.; Paul, S.; Heuer, A.: On the Ordering of Rewrite Rules (Extended Abstract). In: *Proc. of the Second East-European Symposium on Advances in Databases and Information Systems (ADBIS'98), Poznan, Poland, LNCS 1475, Springer*, S. 157–159, September 1998.
- [Pri98] Priebe, D.: Konzeption einer variablenfreien CROQUE-Algebra und Implementierung einer geeigneten Regelverwaltung. Diplomarbeit, Universität Rostock, Fachbereich Informatik, November 1998.
- [RS97] Riedel, H.; Scholl, M. H.: A Formalization of ODMG Queries. In: *Proc. of the 7th Int. Conference on Database Semantics (DS-7), Leysin, Switzerland*, Oktober 1997.