Eine gekürzte Version in: Proc. Datenbanksysteme in Büro, Technik und Wissenschaft, BTW 1995, Dresden, März 1995

Strategien zum dynamischen Aufbau komplexer Objekte in der Anfrageverarbeitung

M. Gesmann

SFB 124
Forschungsbericht Nr. 24/94
Universität Kaiserslautern
Postfach 3049
67653 Kaiserslautern

Dezember 1994

Kurzfassung

Strukturell objektorientierte Datenbanksysteme bieten die Möglichkeit zur direkten Modellierung und Verarbeitung komplexer Objekte, die sich aus elementaren Objekten zusammensetzen. Mit der Entwicklung von deskriptiven, mengenorientierten Anfragesprachen stellt sich auch für diese Systeme, wie für die relationalen Systeme, die Frage nach der Unterstützung der Anfragebearbeitung durch angepaßte Verarbeitungsmodelle, Zugriffspfade und Optimierungsmaßnahmen.

Ein wesentlicher Kostenanteil bei der Bearbeitung komplexer Objekte besteht aus der Interpretation und dem Verfolgen von Referenzen, die Beziehungen zwischen elementaren bzw. komplexen Objekten modellieren. Eine grundlegende Basisoperation ist dabei der Aufbau einfach strukturierter Objekte, mit denen sich dann komplexere, z.B. rekursive oder netzwerkartige Objekte zusammensetzen lassen. Diese Basisoperation wird in der vorliegenden Arbeit genauer untersucht. Die aus relationalen Systemen bekannten Konzepte zur Anfrageverarbeitung können dazu nicht übernommen werden, da die Beziehungen zwischen Objekten dort durch Wertgleichheit von Attributen ausgedrückt und die Verknüpfungen von Relationen durch Verbund-Operationen realisiert werden. In der hier betrachteten Umgebung werden Beziehungen dagegen direkt modelliert und verarbeitet.

Am Beispiel des Non-Standard-Datenbanksystems PRIMA, in dem aufgrund seiner strukturellen Objektorientierung und seiner deskriptiven, mengenorientierten Anfragesprache das oben geschilderte Verarbeitungsproblem auftritt, werden Verfahren zum Aufbau einfach strukturierter Objekte entwickelt. Ausgehend von einem sehr einfachen, aber auch ineffizienten Verarbeitungsmodell wird ein erweitertes Modell aufgestellt, das sehr flexibel auf unterschiedliche Datenbankzustände und Anfrageformulierungen reagieren kann, um so eine optimale Ausführung zu gewährleisten. In dieses Modell werden unterschiedliche Aspekte der Zugriffspfadunterstützung, insbesondere die Clusterung komplexer Objekte, integriert. Darüber hinaus werden Probleme bei der parallelen Bearbeitung der Objekte untersucht.

Inhaltsverzeichnis

1. Einleitung 1
2. PRIMA 2
   2.1 Datenmodell 3
   2.2 Verarbeitungsmodell 3
   2.3 Zugriffspfadstrukturen 3
3. Der AEM-Operator 4
   3.1 Allgemeines 4
   3.2 Ein erstes (einfaches) Verarbeitungsmodell 5
4. Strategien zum Molekülaufbau 8
   4.1 Top-down-Strategie 9
   4.2 Bottom-up-Strategie 13
   4.3 Vollständiges Verarbeitungsmodell 16
   4.4 Verallgemeinertes Verarbeitungsmodell 16
   4.5 Zusammenfassung 17
5. Zugriffspfade 17
   5.1 Ausnutzung konventioneller Zugriffspfade 17
   5.2 Ausnutzung von Atomclustern 17
6. Parallelität 19
   6.1 Beurteilung paralleler AEM-Operator-Ausführungen 19
   6.2 Realisierungsformen 20
7. Vollständige Spezifikation des AEM-Operators 21
8. Zusammenfassung und Ausblick 22
9. Literatur 23

1. Einleitung

Strukturell objektorientierte Datenbanksysteme (SODBS) haben sich in den vergangenen Jahren als geeignete Datenverwaltungssysteme in den sogenannten Non-Standard-Anwendungen erwiesen. Im Vergleich zu relationalen Systemen, die vor allen Dingen die konventionellen Anwendungsklassen durch eine effiziente Verarbeitung großer homogener Datenmengen unterstützen, bieten SODBS die Möglichkeit zur direkten Modellierung und Verarbeitung komplexer Objekte, wie sie z.B. in Entwurfsanwendungen und der Wissensverarbeitung benötigt werden.

Während für relationale Systeme eine fundierte Theorie sowie langjährige Implementierungserfahrung mit Verarbeitungsalgorithmen existieren, sind diese für SODBS nur ansatzweise vorhanden. Dies zeigt sich vor allem bei der Anwendungsunterstützung durch adäquate Anfragesprachen. Für relationale Systeme gibt es mit SQL eine standardisierte, mengenorientierte und deskriptive Anfragesprache. Bemühungen, eine solche Sprache für (strukturell) objektorientierte Systeme zu entwickeln und zu standardisieren, sind noch nicht abgeschlossen und Gegenstand der Forschung [HS91, AWS92, Ki93]. Ihre Notwendigkeit ist aber unumstritten, und erste Sprachvorschläge sind bereits umgesetzt [Ca94, De90].

Einhergehend mit diesen Entwicklungen deskriptiver Sprachen treten Fragen nach angemessenen Verarbeitungsmodellen und geeigneten Umsetzungen der Anfragen in konkrete Verarbeitungspläne auf. Dabei sind Probleme der Optimierung und der Verarbeitungsunterstützung durch angemessene Speicherungsstrukturen sowie der Ausnutzung von Parallelität zu berücksichtigen. Aufgrund der direkten Modellierung von Beziehungen zwischen Objekten durch Referenzen ergeben sich im Vergleich zu relationalen Systemen andere Verarbeitungsmuster und Algorithmen.

In diesem Aufsatz werden verschiedene Strategien zum Aufbau komplexer Objekt aus elementaren Basisobjekten im SODBMS PRIMA [Sch93] vorgestellt und bewertet. PRIMA bietet mit MQL eine deskriptive, mengenorientierte Anfragesprache, die, vergleichbar zu SQL, in der FROM-Klausel die Spezifikation des Typs der aufzubauenden komplexen Objekte, in der WHERE-Klausel die Einschränkung der Ergebnismenge sowie in der SELECT-Klausel die Projektion einzelner Teile der Ergebnisobjekte ermöglicht. In dieser Arbeit konzentrieren wir uns auf solche Anfragen, die in der FROM-Klausel nur baumartig strukturierte Objekttypen beschreiben. Anfragen diesen Typs lassen sich in einer Basisoperation des verwendeten Verarbeitungsmodells, dem AEM-Operator (Aufbau Einfacher Moleküle), berechnen. Die damit aufgebauten Objekte können in speziellen Operationen, die in diesem Aufsatz aber nicht weiter betrachtet werden sollen, zu komplexeren Objekten, z.B rekursiver oder vernetzter Objekttypen zusammengesetzt werden.

Um einen notwendigen Begriffsapparat für die folgenden Diskussionen zur Verfügung zu stellen, wird im 2. Kapitel eine kurze Einführung in das PRIMA-System gegeben. Im 3. Kapitel werden der AEM-Operator, ein einfaches Verarbeitungsmodell sowie die Schwachpunkte dieses Modells diskutiert. In den folgenden Kapiteln werden diese Schwachpunkte durch ein Verarbeitungsmodell beseitigt, das durch eine an die gegebene Qualifikationsbedingung angepaßte Verarbeitung, durch Ausnutzung von Zugriffspfaden und parallele Verarbeitungsstrategien eine optimale Bearbeitung des AEM-Operators ermöglicht. Abschließend erfolgen eine Zusammenfassung der Ergebnisse dieser Arbeit und ein Ausblick auf weitere Untersuchungen.

Verwandte Arbeiten

Da die hier untersuchte Fragestellung schon in einer Reihe von Arbeiten betrachtet wurde, muß eine Einordnung und Abgrenzung erfolgen. Die Grundlage der vorliegenden Arbeit bilden die in [Sch94, Sch93] beschriebenen Probleme und deren Lösungsansätze bei der Anfrageverarbeitung in PRIMA. Von den dort erarbeiteten Strategien wird die Verarbeitung des AEM-Operators hier wesentlich verfeinert und konkretisiert. Zusätzlich berücksichtigen wir hier die parallele Abarbeitung eines AEM-Operators. Dazu

untersucht diese Arbeit, welche algorithmischen Varianten für die Bearbeitung eines AEM-Operators notwendig und sinnvoll sind. Dabei wird dargestellt, wodurch die Anwendbarkeit einzelner Varianten eingeschränkt werden. Darüberhinaus werden Heuristiken für eine vollständige Spezifikation des AEM-Operators und dabei existierende Freiheitsgrade vorgestellt. Es ist kein Ziel dieser Arbeit, ein Kostenmodell zu entwickeln, mit dem verschiedene Verarbeitungspläne bewertet werden können. Ein solches Modell wurde bereits in [Sch93] vorgestellt und müßte für die hier vorgestellten Erweiterungen angepaßt und erweitert werden.

Bei unseren Betrachtungen gehen wir von einem Datenmodell aus, das im Gegensatz zum NF^2-Datenmodell [PSS87] nicht nur die in diesem Aufsatz behandelten hierarchischen, sondern auch rekursive und vernetzte Strukturen direkt modelliert. Im Gegensatz zu DASDBS [SPS90] werden hierarchische Strukturen in PRIMA aber nicht automatisch auch in den zugrundeliegenden Speicherungsstrukturen zusammenhängend abgespeichert. Die in DASDBS realisierte Single-Scan Schnittstelle versucht hierarchische Objekte, die auch direkt in den Speicherungsstrukturen zusammenhängend abgespeichert werden, mit einem einzigen Scan über diesen Speicherungsstrukturen zu lesen. Diese Eigenschaft kann hier aufgrund der anderen Verarbeitungsalgorithmen und Speicherungsstrukturen nicht immer garantiert werden. Dies ist nur durch eine zusätzliche Speicherungsstruktur möglich, bei deren Nutzung sich für den AEM-Operator folglich eine Reihe von Ähnlichkeiten ergeben. Im Vergleich zu [MPP93] benötigen wir die dort beschriebenen Mechanismen zur dynamischen Definition von Beziehungen zwischen Objekten, die dort auf der Basis relationaler Technologie mittels der vorhandenen Verbund-Operationen ausgewertet werden, nicht. Dort ist eine direkte Modellierung der Beziehungen nicht möglich.

Eine Reihe von Arbeiten hat sich mit Speicherungsstrukturen und Zugriffspfaden für komplexe Objekte beschäftigt [KM90, Be94, KVC88, TRS93, JS90, HT94]. Im Gegensatz zu diesen Arbeitengeht es hier um die algorithmische Beschreibung von Verarbeitungsstrategien. Selbstverständlich müssen dabei die vorhandenen Speicherungseigenschaften berücksichtigt werden, wobei wir uns auf die von PRIMA angebotenen Strukturen konzentrieren. In dieser Arbeit betrachten wir wie auch [KGM91] und [KCB87] die Beschreibung der auf den zugrundeliegenden Speicherungsstrukturen aufbauenden Algorithmen. Im Gegensatz zu diesen Arbeiten beruht unser System allerdings auf einer weitgehend datenflußgesteuerten Architektur, die insbesondere den parallelen Aufbau komplexer Objekte ermöglicht. Die in [KKW88] vorgestellten Verfahren zur Anfrageverarbeitung in OODBMS entsprechend weitgehend unseren einfachen, später verbesserten, Verfahren. Allerdings kommen wir durch eine detailliertere Betrachtung der Randbedingungen zu anderen Heuristiken für die genaue Spezifikation. Die Umsetzung der Ausführung geschachtelter Anfragen oder Pfadausdrücke von Programmen mit geschachtelten Schleifen in eine Algebra, in der diese Konstrukte dann durch Verbund-Operationen aufgelöst werden können, wie z.B. in [SAB94] spielt hier keine Rolle. Ähnliche Arbeit wird in vorhergehenden Schritten der Anfrageübersetzung und -transformation durchgeführt, die außerhalb dieses Aufsatzes liegen. Die dabei definierten AEM-Operatoren werden hier als gegeben betrachtet. Im Vergleich zu [KD91] werden allgemeinere Anfragetypen betrachtet. Während dort besonders die Ausnutzung von Pfadindexen als besonderer Speicherungsstruktur zum Aufbau komplexer Objekte für Anfragen untersucht wird, wird hier ein anfrageunabhängiges, allgemeingültiges Modell erarbeitet, das die Spezifikation und Ableitung möglichst optimaler Abarbeitungspläne erlaubt.

2. PRIMA

In diesem Abschnitt wird ein kurzer Überblick über die für die folgende Diskussion wesentlichen Konzepte des PRIMA-Systems gegeben. Ausführlichere Informationen können z.B. [Mi88, Ge94, SS90] entnommen werden.

2.1 Datenmodell

PRIMA ist ein sogenanntes strukturell objektorientiertes Non-Standard-Datenbanksystem, das auf dem Molekül-Atom-Datenmodell (MAD) basiert [Mi88]. Es ermöglicht die direkte Verarbeitung von komplexen Objekten, den sogenannten **Molekülen**. Moleküle setzen sich aus den einfachen Basisobjekten des Modells, den **Atomen**, die ihrerseits durch **Attribute** beschrieben werden, zusammen. Dabei besitzt jedes Atom einen systemweit eindeutigen Objektidentifier (kurz: **Identifier**) und gehört eindeutig zu einem **Atomtyp**. Beziehungen zwischen Atomen eines oder verschiedener Atomtypen werden in sogenannten **Referenzattributen** mit Hilfe der Identifier direkt modelliert. Dabei werden Referenzen grundsätzlich symmetrisch definiert, d.h., zu einem Referenzattribut vom Atomtyp A zum Atomtyp B gibt es immer auch ein entsprechendes Referenzattribut vom Atomtyp B zum Atomtyp A, so daß, wenn es eine Referenz vom Atom a1 vom Atomtyp A zum Atom b1 vom Atomtyp B gibt, es auf jeden Fall auch eine Referenz von b1 zu a1 gibt. Diese Symmetrie der Referenzen wird von PRIMA garantiert. Durch einen **Molekültyp** (kurz: MT) wird der strukturelle Aufbau von Molekülen, bestehend aus den beteiligten Atomtypen und Referenzattributen beschrieben. Jedes Molekül besitzt ein eindeutiges **Wurzelatom** vom ebenso eindeutigen **Wurzelatomtyp** des zugehörigen Molekültyps; beide zeichnen sich dadurch aus, daß sie die einzigen Atome/der einzige Atomtyp sind, die nicht von anderen Atomen/Atomtypen referenziert werden.

2.2 Verarbeitungsmodell

Die Anfrageverarbeitung in PRIMA läßt sich durch ein zweistufiges Modell beschreiben. Das Datensystem führt die Umsetzung der deskriptiven, molekülmengenorientierten Anfragen auf die atommengenorientierte Schnittstelle des Zugriffssystems aus. Dazu wird eine Anfrage in einen **Operatorgraphen** übersetzt, dessen Knoten Operationen der Verarbeitungsalgebra [Sch93] und dessen Kanten dem resultierenden Datenfluß zwischen den Operatoren entsprechen. Dabei bildet der **AEM-Operator** den Basisoperator der Algebra, der vom Zugriffssystem Atommengen anfordern kann und daraus Moleküle einfacher, d.h. baumartig hierarchischer Molekültypen aufbauen kann. Sofern dies nicht explizit anders klargestellt wird, sind im folgenden mit dem Begriff Molekültyp nur solche einfachen Molekültypen gemeint. Der Aufbau vernetzer oder rekursiver Molekültypenwird durch spezielle Operatoren durchgeführt. Die Ausnutzung von parallelen Verarbeitungsstrategien wird dabei durch eine datenflußgesteuerte Abarbeitung der Operatoren ermöglicht [Ge94].

2.3 Zugriffspfadstrukturen

Das Zugriffssystem bietet eine atommengenorientierte Schnittstelle zum Lesen und Ändern von homogenen Atommengen. Durch auf diesen Atomen entscheidbare Qualifikationsbedingungen sowie durch eine Menge von Atomidentifiern kann die Ergebnismenge von Leseanforderungen eingeschränkt werden. Mit Hilfe verschiedener Zugriffspfadstrukturen (siehe Abb. 1) unterstützt das Zugriffssystem ein effizientes Lesen von Atomen. Neben einer **Basisspeicherungsstruktur**, die alle Atome eines Atomtyps zusammenhängend verwaltet, werden auch konventionelle Zugriffspfadstrukturen wie **B*-Bäume** und **Sortierordnungen** angeboten.

Darüber hinaus können **Atomcluster** definiert werden [SS89], die die redundante, zusammenhängende Abspeicherung aller Atome eines Moleküls von einem einfachen Molekültyp erlauben. Durch geeignete Retrievalfunktionen wird mit diesen Atomclustern der häufige Zugriff auf Moleküle des gleichen Molekültyps effizient unterstützt. Der **Atomcluster_Scan** ermöglicht das Lesen aller Atome einer angegebenen Menge von Atomtypen, die in einem konkreten Atomcluster vorhanden sind. Dabei können eine Projektionsliste für die Auswahl einzelner Attribute und eine Identifierliste zur Einschränkung der relevanten Atome angegeben werden. Zusätzlich ermöglicht der **Atomclustertyp_Scan** eine Auswertung

komplexerer Bedingungen auf einem zusammenhängend abgespeicherten Molekül, so daß nur solche Moleküle aus dem Cluster in der weiteren Verarbeitung betrachtet werden müssen, die diese Bedingung erfüllen. Ebenso wie bei der Single-Scan-Schnittstelle des DASDBS-Projektes können mit diesen Operationen ganze Moleküle durch einen einzigen Scan über diese Speicherungsstruktur gelesen werden. Allerdings handelt es sich hier in PRIMA um eine optionale, redundante Speicherungsstruktur, die nicht für alle (dynamisch) in Anfragen spezifizierten Komplexobjekttypen vorhanden ist. Die zur Aktualisierung dieser Speicherungsstrukturen benötigte Zeit wird durch geeignete parallele Verarbeitungsstrategien möglichst klein gehalten.

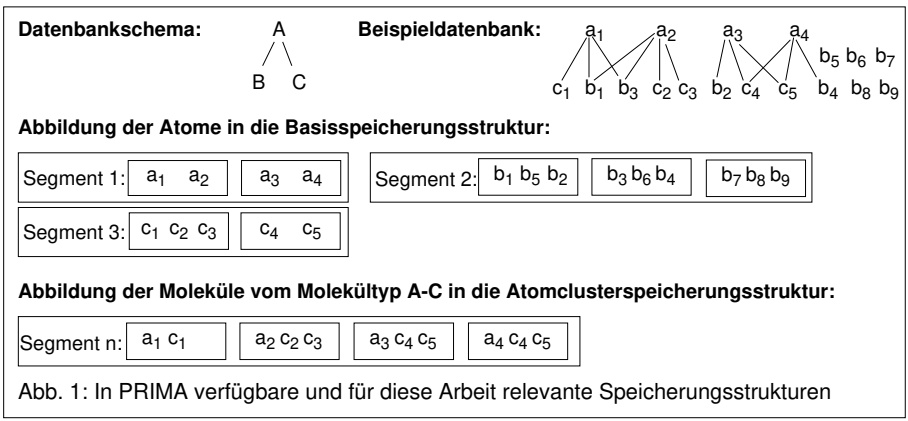


Abb. 1: In PRIMA verfügbare und für diese Arbeit relevante Speicherungsstrukturen

3. Der AEM-Operator

Nachdem nun die grundlegenden Begriffe erläutert wurden, kann der bereits erwähnte AEM-Operator im Detail vorgestellt werden. Dazu werden in Abschnitt 3.1 seine Funktionalität sowie Anforderungen an seine Realisierung beschrieben. Anschließend wird anhand eines einfachen Beispiels ein erstes allgemeines Verarbeitungsmodell vorgestellt, das eine Reihe von konzeptionellen Schwächen aufweist, die in den folgenden Kapiteln behoben werden.

3.1 Allgemeines

Der bereits im 2. Kapitel kurz vorgestellte AEM-Operator hat die Aufgabe, Moleküle aufzubauen, die einem einfachen Molekültyp (**AEM-MT**) angehören und angegebene Bedingungen erfüllen. Dies entspricht der Bearbeitung von Anfragen vom Typ “SELECT S(M) FROM M WHERE Q(M)”, wobei M der AEM-MT, Q(M) eine auf diesem Molekültyp entscheidbare Bedingung und S(M) die zu erfolgende Projektion sind. Beim Aufbau der Moleküle bedient sich der Operator der vom Zugriffssystem angebotenen Funktionen und Zugriffspfade, um die Atome aus den Speicherungsstrukturen zu lesen, die für die Bedingungsauswertung bzw. für den Aufbau der sich qualifizierenden Ergebnismoleküle benötigt werden.

Im Rahmen der Anfrageverarbeitung sind folgende Ziele bei der Entwicklung von Verarbeitungsstrategien für diesen Operator wesentlich:

(1) die gesamte Ergebnismenge soll möglichst schnell vollständig bestimmt werden und

(2) die einzelnen Ergebnisse sollen möglichst schnell aufgebaut werden.

Während sich der erste Punkt von selber erklärt, muß der zweite motiviert werden. Wie bereits erwähnt, stellt der AEM-Operator eine Grundfunktion in der Anfrage-Verarbeitung dar. Die dort erzeugten Mo-

leküle werden in komplexeren Anfragen, die das obige Muster sehr einfacher SELECT-Anfragen verlassen, durch weitere Operatoren verarbeitet. Eine datenflußgetriebene Architektur, die mittels Pipelining Parallelität zwischen den Operatoren ausnutzt, soll das Gesamtergebnis der Anfrage möglichst schnell berechnen [Sch93]. Dieses Pipelining kann aber nur dann funktionieren, wenn entsprechend geeignete, d.h. gleichmäßige Eingabeströme erzeugt werden, die von den AEM-Operatoren (Blattoperatoren, siehe 2.2) ausgehen. Deshalb muß der AEM-Operator die einzelnen Ergebnismoleküle möglichst schnell und auch gleichmäßig produzieren.

3.2 Ein erstes (einfaches) Verarbeitungsmodell

Die bisher erläuterten Begriffe und die Problemstellung der Aufgabe werden nun anhand eines einfachen Beispiels kurz erläutert. Mit Hilfe dieses Beispiels wird ein erstes, sehr einfaches Verarbeitungsmodell vorgestellt, das als Ausgangsmodell für weitere Verfeinerungen dient.

Für unser Beispiel sei ein stark vereinfachter Ausschnitt aus einer Entwurfsdatenbank für den VLSI-Chip-Entwurf gegeben (siehe Abb. 2). Die Funktionalität einer Zelle Z wird durch ihre Spezifikation Sp formal beschrieben. Eine Dokumentation D enthält eine Beschreibung des Entwurfsprozesses. Sowohl die Spezifikation als auch die Dokumentation werden verantwortlich von (möglicherweise verschiedenen) Angestellten A erstellt. Die letztendlich gefertigte Zelle wird durch eine Realisierung R beschrieben, deren zellenunabhängige Fertigungsparameter, wie z.B. die Technologie T, explizit modelliert werden. Schließlich werden hierarchische Beziehungen zwischen Zellen bzw. zwischen verschiedenen Realisierungen mit reflexiven Beziehungen auf den Atomtypen Z (enthält, geht_ein_in) und R (besteht_aus, ist_bestandteil_von) modelliert. Auf dieser Datenbank werde eine einfache Anfrage gestellt, die alle Zellen, die einen 2-Bit Addierer realisieren, zusammen mit ihrer Dokumentation, ihrer Spezifikation und ihrer Realisierung sowie der dazu verwendeten Technologie extrahieren, sofern als Technologie MOSFETs benutzt werden.

Für den Aufbau der zu erzeugenden Moleküle sind eine Vielzahl von Algorithmen denkbar. Die einfachen Algorithmen lassen sich nach folgenden Kriterien klassifizieren:

- Anforderungsgranulat
  Atomorientierte Leseaufträge fordern vom Zugriffssystem einzelne Atome an, die als Bestandteile des aufzubauenden Moleküls gelesen werden müssen. Dagegen berücksichtigen atommengenorientierte Algorithmen immer die Gesamtheit oder eine Teilmenge der zu lesenden Atome eines Atomtyps.
- Bearbeitungsgranulat
  Ebenso wie die Atomanforderungen für Moleküle atom- oder atommengenorientiert erfolgen können, kann die Erzeugung der Moleküle selber molekül- oder molekülmengenorientiert erfolgen. Molekülorientierte Algorithmen bearbeiten die aufzubauenden Moleküle isoliert voneinander, während molekülmengenorientierte Verfahren gleichzeitig mehrere oder sogar alle Moleküle betrachten.

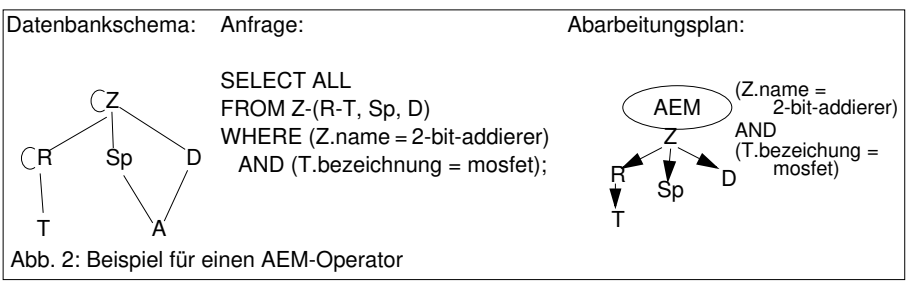


Abb. 2: Beispiel für einen AEM-Operator

- Verarbeitungsreihenfolge
  Aufgrund der baumartigen Molekültypstrukturen können Depth-first- und Breadth-first-Strategien unterschieden werden. Depth-first-Strategien versuchen, einzelne Teilmoleküle möglichst schnell zu konstruieren. Breadth-first-Strategien bauen die Moleküle dagegen atomtypweise auf, indem sie in der entsprechenden Reihenfolge alle Atome zu einem Atomtyp im betroffenen Molekül lesen.

Die vorgestellten Kriterien sind zueinander orthogonal und allgemeingültig, d.h., sie können für jede konkrete Operatorspezifikation unabhängig voneinander betrachtet werden und sind für alle möglichen AEM-MT anwendbar. Dabei entsprechen die molekülorientierten Verfahren der nested-loop Methode und die molekülmengenorientierten der sort-domain Methode aus [KKW88]. Die beiden übrigen Kriterien werden dort nicht berücksichtigt. In Abb. 3 werden beispielhaft drei Algorithmen skizziert, ohne im letzten Detail das Anforderungsgranulat für die Atomanforderungen festzulegen. Während die beiden ersten Varianten ((a) und (b)) einzelne Moleküle in Depth- bzw. Breadth-first-Ordnung aufbauen, arbeitet der dritte Algorithmus (c) molekülmengenorientiert in einer Breadth-first-Strategie. Eine mengenorientierte Depth-first-Variante, vergleichbar zu (c), ist natürlich auch möglich.

(a) Molekülorientiert, Depth-first:	(b) Molekülorientiert, Breadth-first:	(c) Molekülmenge, Breadth-first:

WHILE noch ein Wurzelatom da
WHILE noch referenzierte Atome zu lesen
baue für sie rekursiv die betroffenen Teilmoleküle auf
END;
baue Molekül auf und gebe es aus, falls es die Bedingung erfüllt;
END;
	WHILE noch ein Wurzelatom da
	REPEAT
	bestimme alle noch zu lesenden Atome, die von den bisher gelesenen referenziert werden, und lese sie;
	UNTIL alle Atome gelesen
	baue Molekül auf und gebe es aus, falls es die Bedingung erfüllt;
	END;
		lese alle Wurzelatome;
		REPEAT
		bestimme alle noch zu lesenden Atome, die von den bisher gelesenen referenziert werden, und lese sie alle;
		UNTIL alle Atome gelesen
		baue Moleküle auf und gebe sie aus, falls sie die Bedingung erfüllen;

Abb. 3: Beispielhafte Algorithmen für den AEM-Operator

Aus Gründen, die im folgenden genauer beschrieben werden, können allein nach diesen Kriterien entworfene Verarbeitungsstrategien allerdings kein gutes Leistungsverhalten aufweisen. Dabei lassen sich die einzelnen Gründe in drei Problemklassen einteilen.

3.2.1 Fehlende Anpassung der Strategien an die gegebenen Qualifikationsbedingungen

(1) Die bisher beschriebenen Verfahren berücksichtigen in der WHERE-Klausel spezifizierte Restriktionen, die erst nach einem vollständigen Molekülaufbau ausgewertet werden, überhaupt nicht. Dies ist bei sich nicht qualifizierenden Molekülen natürlich unsinnig, wenn die Bedingung bereits zu einem früheren Zeitpunkt auswertbar ist. In vielen Fällen ist es aufrund des gegebenen AEM-MT und der gegebenen Bedingung überhaupt nicht notwendig, das gesamte Molekül zu betrachten, da oft ein Ausschnitt aus dem AEM-MT genügt, nach dessen Aufbau die Bedingung in jedem Fall entscheidbar ist (im Beispiel Z, R und T). Die übrigen Teile (Sp, D) tragen nichts zur Bedingungsauswertung bei. Deshalb sollte im Abarbeitungsplan genauer als bei den allgemeinen Strategien festgelegt werden können, wie die Moleküle aufgebaut werden und wann Bedingungen ausgewertet werden sollen.

(2) Die oben dargestellte Problematik kann auf einzelne Teilbedingungen erweitert werden, wenn diese bereits zur endgültigen Auswertung führen können. In diesem Fall kann es sinnvoll sein, zunächst nur die zur Auswertung einer Teilbedingung benötigten Teilmoleküle aufzubauen. So sollten z.B. in einer mit AND verknüpften Bedingung diejenigen Molekülteile zuerst aufgebaut werden, deren zugehörige

Teilbedingung schneller zu einer Disqualifikation des Moleküls führt. So können aufgrund einer frühzeitigen Disqualifikation unnötige Leseoperationen für andere Atome eingespart werden.

3.2.2 Fehlende Ausnutzung von Zugriffspfaden

(3) Die bisherige Beschreibung der Operatorverarbeitung berücksichtigt noch keine Zugriffspfade. Konventionelle Zugriffspfade (z.B. B*-Baum, Grid-File), die das Lesen von einzelnen Atomen unterstützen, können einfach integriert werden, indem ihre Nutzung beim Aufruf an das Zugriffssystem spezifiziert wird. Anders verhält es sich bei zusammenhängender Abspeicherung der Atome einzelner Moleküle in Atomclustern, deren gezielte Nutzung im AEM-Operator spezifizierbar sein muß. Sie unterstützen in besonderer Weise die molekülorientierten Aufbauverfahren. Eine entsprechende Clusterbildung der Atome nach ihrer Zugehörigkeit zu Molekülen in der Basisspeicherungsstruktur, d.h., zu einem Molekül gehörende Atome eines Atomtyps werden auch dort zusammenhängend abgespeichert, ist normalerweise nicht möglich, so daß der Molekülaufbau hierdurch nicht unterstützt werden kann. Erstens ist die notwendige eindeutige Zuordnung eines Atoms zu einem Molekül in der Regel nicht gegeben, da ein Atom über verschiedene (oder gleiche) Referenzattribute zu unterschiedlichen Molekülen gehören kann. Im Beispiel können z.B. mehrere Realisierungen zu unterschiedlichen Zellen trotzdem die gleiche Technologie verwenden. Zweitens kann sich die zusammenhängende Abspeicherung nur an einem einzigen Referenzattribut orientieren. Man beachte die Analogie zum Netzwerkmodell, in dem die Abspeicherung von Member-Records in einem Set als Cluster auch nur für einen Settyp möglich ist. Allerdings kann dort ein Member-Record nur in einer Set-Ausprägung eines Settyps enthalten sein, während ein Atom in unserer Umgebung durch die direkte Modellierung der n:m-Beziehungen über ein Referenzattribut durchaus in Beziehung zu mehreren Atomen stehen kann.

3.2.3 Fehlende Ausnutzung von Parallelität

(4) Atomorientierte Leseoperationen erzeugen gegenüber atommengenorientierten Anforderungen aufgrund der höheren Anzahl an Leseaufträgen wesentlich mehr Zusatzaufwand. Zusätzlich erschweren sie dem Zugriffssystem eine sinnvolle Ausnutzung von Parallelität, weil alle Leseaufträge dort unabhängig voneinander ankommen und so auch nur kontextfrei bearbeitet werden können. Dies kann zwar parallel geschehen, aber aufgrund der unkontrollierten Ausführung können inhaltliche Zusammenhänge, z.B. Anforderungen verschiedener Atome, die in derselben Seite liegen, nicht genutzt werden. Somit müssen Seiten nach einer zwischenzeitlichen Verdrängung aus dem Puffer ggf. wiederholt angefordert werden. Bei atommengenorientierten Anforderungen kann das Zugriffssystem dagegen parameter- und datenabhängig entscheiden, wie die Anforderung bearbeitet werden soll und z.B. **Prefetching** und parallele Verarbeitungsstrategien einsetzen. Dabei ist es möglich, auch inhaltliche Zusammenhänge bei der Anforderung mehrerer Atome zu berücksichtigen.

(5) Das für die Atomanforderungen dargestellte Problem läßt sich auf den molekül- bzw. molekülmengenorientierten Molekülaufbau übertragen. Die molekülorientierten Algorithmen können sehr leicht die Lokalität der Seitenzugriffe im Zugriffssystem zerstören, da eine Seite im ungünstigsten Fall bei jedem Molekül einmal gelesen wird, wenn jedes Atom in der Seite zu einem anderen Molekül gehört. Dies führt in der Regel zu schlechten Antwortzeiten bei Leseanforderungen an das Zugriffssystem, die sich auch auf die Antwortzeit für die gesamte Anfrage auswirken. Bei einer molekülmengenorientierten Verarbeitung, wie z.B. Variante (c) in Abb. 3, werden dagegen alle in einer Seite liegenden Atome der betrachteten Molekülmenge zusammen angefordert, so daß diese Seite nicht wiederholt eingelesen werden muß.

(6) Das soeben geschilderte Problem tritt bei molekülmengenorientierten Verfahren nicht auf. Aufgrund folgender Probleme stellen sie aber auch keine allgemeingültige Lösung dar. Zunächst kommen sie unserer Forderung nach möglichst gleichmäßiger Produktion der Ergebnisse nicht nach (siehe 3.1), weil

alle Ergebnismoleküle erst gegen Ende der Operatorausführung fertiggestellt werden, so daß das angestrebte Pipelining nicht mehr sinnvoll ausgenutzt werden kann. Nur im Falle sehr kurzer Pipelines, in denen die auf den AEM-Operator folgenden Operatoren weniger Zeit zur Verarbeitung der Moleküle benötigen, ist die Antwortzeit für die Bearbeitung des AEM-Operators das alleinige Optimierungskriterium. Desweiteren können sie Speicherplatzprobleme auslösen, wenn erst große Mengen von Teilmolekülen aufgebaut werden, die dann aufgrund von sehr selektiven Bedingungen nicht mehr benötigt werden. In diesem Fall führt eine externe Zwischenspeicherung der Teilmoleküle zu hohen Ein-/Ausgabekosten.

(7) In den vorgestellten Verfahren werden die für ein Molekül benötigten Atome unbedingt gelesen. Aufgrund der vernetzten Strukturen innerhalb desselben und zwischen verschiedenen Molekülen, die durch die direkte Modellierung von n:m-Beziehungen über die Referenzen entstehen, kann es aber sein, daß Atome bereits beim Aufbau des aktuellen bzw. anderer Moleküle gelesen wurden. In diesem Fall ist ein mehrfaches Lesen desselben Atoms zu vermeiden, sofern der Aufwand für ein entsprechendes Vorgehen kleiner ist als der Aufwand für das Mehrfachlesen.

(8) Wenn schließlich die Untersuchungen nicht mehr auf einen einzelnen AEM-Operator beschränkt werden, sondern auf die Menge aller AEM-Operatoren in einem Operatorgraphen erweitert wird, soll auch dabei mehrfaches Lesen eines Atoms aus der Datenbank effektiv verhindert werden. Dies ist dann notwendig, wenn der gleiche Atomtyp mit unterschiedlicher Bedeutung an verschiedenen Stellen im Ergebnismolekültyp der Anfrage vorkommt, z.B., wenn er von verschiedenen Atomtypen referenziert, und in verschiedenen AEM-Operatoren behandelt wird. Der Zusatzaufwand für dieses Vorgehen sollte aber wiederum eingespart werden, wenn aufgrund der Datencharakteristik klar ist, daß keine oder nur sehr wenige Atome mehrfach benutzt werden.

Wie die hier skizzierten Probleme gelöst werden können, wird in der folgenden Diskussion genauer untersucht. Dabei werden unabhängige Teilprobleme getrennt voneinander betrachtet. Wir beginnen mit einer Untersuchung verschiedener Typen von Qualifikationsbedingungen und deren Behandlung im AEM-Operator (Probleme 1 und 2, Kapitel 4). Anschließend wird die Nutzung verschiedener Zugriffspfade, insbesondere der Atomcluster, erörtert (3, 5). Danach werden Fragen der Parallelität in der AEM-Operator-Ausführung betrachtet (4-8, 6). Die in diesen Abschnitten erzielten Ergebnisse führen zu einem allgemeinen Verarbeitungsmodell für den AEM-Operator.

4. Strategien zum Molekülaufbau

Wie im vorangegangenen Abschnitt bereits dargestellt wurde, ist es sinnvoll, die in der WHERE-Klausel spezifizierbaren Bedingungen näher zu untersuchen. An diese Bedingungen angepaßte Verarbeitungsalgorithmen erlauben eine Entscheidung der Bedingung, ohne die für diese Entscheidung überflüssigen Atome lesen zu müssen.

Grundsätzlich entsteht bei der beschriebenen Verfahrensweise allerdings folgender Zielkonflikt. Einerseits soll möglichst wenig unnötige Arbeit geleistet werden, wenn sich Moleküle nicht qualifizieren, d.h. Teile, die nicht zur Entscheidung der Bedingung benötigt werden, sollen auch nicht bearbeitet werden. Andererseits sollen die Moleküle möglichst schnell aufgebaut werden, d.h., bei sich qualifizierenden Molekülen sollen unabhängige Teilmoleküle parallel aufgebaut werden. Um dieses Problem zu lösen, ist datenabhängig von Fall zu Fall zu unterscheiden, welche Strategien angewandt werden sollen. Als Grundlage für diese Entscheidung dienen dem Optimierer Statistiken über Werteverteilungen der Attribute und die Struktur der vorliegenden Bedingung.

Für eine weitere Betrachtung klassifizieren wir zunächst mögliche (Teil-) Bedingungen, die im folgenden Ausdrücke genannt werden, wenn nicht unterschieden wird, ob es sich um die ganze Bedingung oder um Teilbedingungen handelt:

- Einatomtyp-Ausdrücke
  Hierbei handelt es sich um Ausdrücke, die auf einem einzelnen Atomtyp im AEM-MT entschieden werden können, z.B. (WHERE Atomtyp1.Attribut1 > 3) oder (WHERE EXISTS Atomtyp1).
- Mehratomtyp-Ausdrücke
  Diese Ausdrücke sind nicht auf Atomen eines einzelnen Atomtyps entscheidbar, sondern sie betreffen mehrere Atomtypen, z.B. (WHERE Atomtyp1.Attribut1 > Atomtyp2.Attribut1).
- quantifizierte Ausdrücke
  Sofern dies nicht ausdrücklich vom Anwender in der WHERE-Klausel geschieht, werden noch nicht quantifizierte Atomtypen in PRIMA existentiell quantifiziert. Neben dem existentiellen Quantor (EXISTS) sind auch noch der universelle Quantor (FOR_ALL) sowie einige Spezialquantoren (EXISTS_AT_LEAST n, EXISTS_AT_MOST n und EXISTS_EXACTLY n) verfügbar.

Für alle Bedingungen gilt notwendigerweise, daß sie auf dem AEM-MT auswertbar sind. Der kleinste Ausschnitt aus einem Molekültyp, der alle in einem Ausdruck Q referenzierten Atomtypen umfaßt, wird im folgenden Ausdrucksmolekültyp (**AMT**) genannt. Der kleinste AMT, der die Wurzel des AEM-MT enthält, wird wurzelbasierter Ausdrucksmolekültyp (**wAMT**) genannt. Die Erweiterung eines AMTs zu einem wAMT besteht einfach aus der Hinzunahme aller Atomtypen und Referenzattribute auf dem Pfad (d.h. den Atomtypen und den Referenzattributen) von der Wurzel des AEM-MTs zur Wurzel des AMTs. Offensichtlich besteht der AMT bei Einatomtyp-Ausdrücken nur aus diesem Atomtyp, bei einem Einatomtyp-Ausdruck auf der Wurzel besteht auch der wAMT nur aus dem Wurzelatomtyp. Im Beispiel von Abb. 2 besteht der wAMT zum Ausdruck (T.bezeichnung = mosfet) aus den Atomtypen Z, R und T und den benutzten Referenzattributen.

4.1 Top-down-Strategie

Wenn, wie gefordert, in Abhängigkeit von der Bedingung das unnötige Lesen für die Bedingungsauswertung nicht benötigter Atome vermieden werden soll, genügen die allgemeinen Breadth-first- und Depth-first-Strategien nicht mehr. Aus diesem Grund wird nun ein etwas erweitertes Modell vorgestellt, das diese Forderung besser erfüllt. Dabei werden wie bisher, ausgehend von der Wurzel, Molekülteile aufgebaut, die zur Auswertung der Bedingung benötigt werden.

Im Gegensatz zu den einfachen bereits beschriebenen Strategien, die die Bearbeitungsreihenfolge implizit enthalten, wird die Reihenfolge jetzt im Molekültypgraphen explizit durch eine Reihenfolgenummer für jeden Atomtyp festgelegt. Atomtypen mit der gleichen Nummer können auch gleichzeitig angefordert werden. Somit kennzeichnen die Nummern verschiedene Phasen des Molekülaufbaus, wobei eine Phase mit der Phasennummer (i+1) erst dann beginnen darf, wenn die Phase i beendet ist. Dabei ist eine Phase i dann beendet, wenn keine Atome mit der entsprechenden Phasennummer mehr zu lesen sind und die gegebene Bedingung ausgewertet wurde. Falls die Bedingung noch nicht entschieden ist, wird die Bearbeitung mit der nächsten Phase fortgesetzt. Ansonsten können die Ergebnismoleküle, für die die Bedingung zu TRUE ausgewertet wurde, bedingungsunabhängig fertiggestellt werden.

Ausgehend von der Wurzel, die die Phasennummer 1 bekommt, wird so eine Reihenfolge festgelegt, in der die Atome der Atomtypen bearbeitet werden. Dabei darf die Phasennummer eines referenzierenden Atomtyps im Molekültyp nie größer sein, als die irgendeines seiner referenzierten Atomtypen. Wie eine korrekte Phaseneinteilung definiert werden kann, wird im folgenden Abschnitt beschrieben. Zur Illustration enthält Abb. 4 für den einfachen AEM-MT aus unserem Beispiel verschiedene Verarbei-

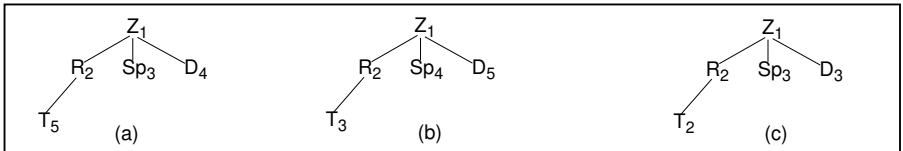


Abb. 4: Beispiele für die Top-down-Phaseneinteilungen

tungsreihenfolgen, wobei a) die Breadth-first-Strategie, b) die Depth-first-Strategie und c) eine an die Bedingung angepaßte Reihenfolge festlegt. In c) werden in der ersten Phase die Z-Atome gelesen, die den ersten Teil der Bedingung (Z.name = 2-bit-addierer) erfüllen. In der anschließenden 2. Phase werden nur noch für diese Wurzelatome die Teilmoleküle R-T aufgebaut. Auf den so aufgebauten Teilmolekülen wird dann auch der zweite und letzte Teil der Bedingung ausgewertet, so daß nur noch für die qualifizierten Moleküle in der abschließenden 3. Phase die noch fehlenden zum Molekül gehörenden Atome gelesen werden müssen. Aufgrund des vom Wurzelatomtyp zu den Blättern gerichteten Molekülaufbaus wird dieses Verfahren Top-down-Strategie genannt.

4.1.1 Phaseneinteilung

Um für konkrete AEM-Operatoren gezielt Phaseneinteilung spezifizieren zu können, benötigen wir allgemeine Regeln, die im folgenden hergeleitet werden sollen. Dazu betrachten wir eine Qualifikationsbedingung Q, die aus mehreren mit UND bzw. ODER verknüpften Ausdrücken Q_i (Konjunktionen/Disjunktionen) besteht. Zu jeder Q_i gehört ein $wAMT_i$, der für die Auswertung aufgebaut werden muß. Die hier vorgenommene Untersuchung der Bedingung Q kann rekursiv auf die einzelnen Q_i übertragen werden, so daß das folgende Verfahren hierarchisch von der ganzen Bedingung bis zu nicht weiter zerlegbaren Ausdrücken angewandt werden kann. Aus diesem Grund spielt eine Normalisierung der Bedingungen in eine konjunktive oder disjunktive Normalform für die weiteren Untersuchungen keine Rolle, da beide Verknüpfungsoperatoren betrachtet werden müssen. Aufgrund der unterschiedlichen Verknüpfungssemantik muß aber zwischen Konjunktionen und Disjunktionen unterschieden werden.

Bei einer Konjunktion von Ausdrücken ist die Bedingung dann nicht erfüllt und ihre Auswertung kann abgebrochen werden, wenn ein Ausdruck nicht erfüllt ist. Um also unnötige Arbeit zu vermeiden, ist es sinnvoll, die einzelnen Ausdrücke sequentiell auszuwerten, d.h. die einzelnen AMT_i sequentiell aufzubauen und anschließend den Ausdruck Q_i auszuwerten. Zur Festlegung der Reihenfolge, in der die einzelnen Q_i bearbeitet werden, muß der Optimierer für jede Q_i seine Selektivität S_{Qi}, also die Wahrscheinlichkeit, mit der Q_i zu FALSE ausgewertet wird, und die Kosten K_{AMTi} für den Aufbau der Teilmoleküle vom AMT_i (z.B. die Anzahl der zu lesenden Seiten) betrachten. Dazu werden statistische Informationen aus den Metadaten der Datenbank genutzt, die hier aber nicht näher betrachtet werden sollen. Eine alleinige Berücksichtigung der Selektivitäten genügt nicht, da für die einzelnen Teilbedingungen in der Regel unterschiedlich große AMT_i aufgebaut werden müssen. Dies kann dazu führen, daß es günstiger ist eine Teilbedingung mit geringerer Selektivität, die einfach und kostengünstig ausgewertet werden kann, einer anderen Teilbedingung mit höherer Selektivität aber sehr hohem Berechnungsaufwand vorzuziehen. Da es gilt, den Gesamtaufwand im Falle der Disqualifikation des Moleküls zu minimieren, wird ein Ausdruck Q_i vor einem Ausdruck Q_j ausgewertet, wenn $S_{Qi}/K_{AMTi} > S_{Qj}/K_{AMTj}$, d.h., je höher die Selektivität und je geringer die Kosten zum Molekülaufbau für einen Ausdruck Q_i, umso früher wird dieser Ausdruck betrachtet. Die Bestimmung der S_{Qi} und K_{AMTi} soll hier nicht weiter beschrieben werden, sie ist in [Sch93] enthalten.

Bei Disjunktionen ist die Behandlung wesentlich einfacher. Da eine Bedingung erst dann zu FALSE entschieden wird, wenn alle Ausdrücke zu FALSE ausgewertet wurden, müssen in jedem Fall alle AMT_i aufgebaut werden. Aus diesem Grund können alle Ausdrücke und AMT_i parallel betrachtet werden.

Selbstverständlich kann eine weitere Bedingungsauswertung entfallen, sobald der Ausdruck zu TRUE ausgewertet wurde. Falls alle zur Bedingungsauswertung benötigten Teilmoleküle auch zur Ausgabe der Anfrage gehören, wird mit diesem Verfahren keine unnötige Arbeit durchgeführt. Anders verhält es sich, bei nicht auszugebenden Molekülteilen. In diesem Fall ist ein zur konjunktiven Verknüpfung ähnliches Vorgehen notwendig, mit dem Unterschied, daß die Ausdrücke möglichst früh zu TRUE ausgewertet werden sollen. Sobald der Ausdruck zu TRUE ausgewertet wurde, brauchen die Molekülteile, die nicht ausgegeben werden, auch nicht mehr aufgebaut werden.

Eine nach diesen Kriterien erstellte Phaseneinteilung beschreibt eine Abarbeitungsreihenfolge, die nur in geringem Maße zu unnötiger Arbeit beim Aufbau von Molekülen führt, da die Disqualifikation und die Qualifikation möglichst früh entschieden werden, so daß nach der Entscheidung der Bedingung nur noch die auszugebenden Molekülteile aufgebaut werden müssen. Allerdings erweist sich die strikte Trennung der Phasen, also die Ausführung von Phase i+1 erst nach vollständiger Beendigung der Phase i, zu restriktiv. Aufgrund von Zusammenhängen in den Bedingungen, die bei der Phaseneinteilung verlorengehen, können Teile der Phase i+1 schon bearbeitet werden, nachdem bestimmte Teile der Phase i bearbeitet worden sind, aber die gesamte Phase i noch nicht beendet ist. Es sei beispielsweise eine Bedingung der Form ((Q_1 AND Q_2) OR Q_3) gegeben. Q_1 habe eine höhere Selektivität als Q_2 und Q_3 habe einen deutlich höheren Berechnungsaufwand als die beiden anderen Ausdrücke. Schließlich sollen alle für die Bedingungsauswertung benötigten Teilmoleküle auch projiziert werden. Mit diesen Eigenschaften sowie den bisher vorgestellten Kriterien zur Phaseneinteilung werden Q_1 und Q_3 in der ersten Phase und Q_2 in der zweiten Phase behandelt ((Q_1 AND Q_2) parallel mit Q_3, Q_1 vor Q_2). Mit der Bearbeitung von Q_2 wird gewartet, bis auch Q_3 mit ihrem hohen Berechnungsaufwand beendet ist. Dies ist aber nicht unbedingt notwendig, da Q_2 schon nach Beendigung von Q_1 bearbeitet werden kann, was i.a. zu einer Reduzierung der Antwortzeit führt. Um dieses zu ermöglichen, werden diese Präzedenzbeziehungen (im Beispiel zwischen Q_1 und Q_2) ebenfalls in die Strategiebeschreibung aufgenommen und die bisherige strikte Phaseneinteilung aufgehoben. Eine Phase i+1 kann nun bereits begonnen werden, wenn in der Phase i ein gemäß einer Präzedenzbeziehung festgelegter Atomtyp bereits vollständig bearbeitet wurde.

Insgesamt ergibt sich somit der in Abb. 5 dargestellte rekursive Algorithmus für die Phaseneinteilung in einem AEM-Operator. Eingabe sind der aktuell betrachtete Ausdruck Q sowie die Phasennummer, mit

```
PROCEDURE Definiere_Phaseneinteilung (      Q                : Bedingung;
                                            phn              : Phasennummer;
                                      VAR maximale_phn       : Phasennummer);
BEGIN
   IF (Q nicht weiter zerlegbar) THEN
      alle Atomtypen des zu Q gehörenden AEM-MT werden in Phase phn bearbeitet, sofern sie nicht
           duch eine andere Teilbedingung bereits eine Phasennummer haben, die kleiner als phn ist.
      IF maximale_phn < phn THEN maximale_phn := phn END;
   ELSIF (Q zerlegbar in ODER-verknüpfte Qi) THEN
      Für alle Qi: Definiere_Phaseneinteilung(Qi, phn, maximale_phn);
   ELSE
      sortiere alle Qi nach (SQi/KAEM-MTi);
      in sortierter Reihenfolge für alle Qi:
         Definiere_Phaseneinteilung(Qi, phn, maximale_phn);
         kennzeichne Reihenfolge mit vorhergehendem AEM-MTi-1;
         phn := maximale_phn + 1;
   END;
END Definiere_Phaseneinteilung;
```

Abb. 5: Algorithmus zur Phaseneinteilung für den Top-down-Molekülaufbau

der dieser Ausdruck begonnen wird. Der Algorithmus vermerkt bei jedem Atomtyp im AMT dessen Phasennummer sowie die oben vorgestellten Präzedenzbeziehungen. Nach der vollständigen Ausführung des Algorithmus bekommen alle noch nicht behandelten Atomtypen die niedrigste, noch nicht vergebene Phasennummer; dabei handelt es sich um die Atomtypen, die nicht in der Bedingung vorkommen und nicht auf den Pfaden von der Wurzel zu solchen Atomtypen liegen. Diese abschließende Phase entspricht dem Vervollständigen derjenigen Moleküle, die sich gemäß der gegebenen Bedingung qualifiziert haben, um die noch fehlenden Atome. Zu beachten sind mögliche Überlappungen bei solchen Atomtypen, die in mehreren AMTen auftreten. In diesen Fällen werden die Atome in der Phase mit der niedrigsten auftretenden Phasennummer gelesen und zur Bedingungsauswertung benutzt (siehe Abb. 6 Beispiel 2).

Das hier vorgestellte Verfahren zur Einteilung des Molekülaufbaus in verschiedenen Phasen kann noch an einigen Stellen weiter verfeinert werden. Beispielsweise werden bisher einzelne Konjunktionen völlig isoliert voneinander betrachtet. Um die Antwortzeit zu reduzieren, können sie aber auch sinnvoll parallel berechnet werden, wenn die Ausdrücke nur eine geringe Selektivität aufweisen, so daß die Gesamtbedingung mit hoher Wahrscheinlichkeit zu TRUE ausgewertet wird.

4.1.2 Phasenabarbeitung

Die bisherige Phaseneinteilung orientiert sich ausschließlich an den von Bedingungen betroffenen Atomtypen und betrachtet noch nicht die Ausführung der einzelnen Phasen, sondern legt nur eine "Reihenfolge" fest, in der einzelne Teile des Molekültyps aufgebaut werden sollen. Gerade bei molekülorientierten Aufbaustrategien, die die Lokalität der Seitenreferenzen im Zugriffssystem stark verringern können, ist es deshalb wichtig, die Anzahl der Leseoperationen je Atomtyp auf ein notwendiges Minimum zu reduzieren. Falls also mehrere Atome referenziert werden, sollen nur die unbedingt benötigten gelesen werden. Dazu betrachten wir die Quantoren auf den betroffenen Atomtypen in einer Q_i.

Bei existentiell quantifizierten Ausdrücken kann jedes einzelne Teilmolekül des betroffenen AMT zur Qualifikation des Ausdrucks führen. Deshalb müssen ggf. alle Teilmoleküle aufgebaut werden. Dies kann folglich wieder parallel geschehen, ohne daß viel unnötige Arbeit geleistet wird, wenn alle Teilmoleküle auch ausgegeben werden. Zuviel geleistete Arbeit im Falle einer späteren Disqualifikation (wenn dieser Ausdruck mit anderen Ausdrücken konjugiert wird, die in späteren Phasen bearbeitet werden) ist hier vernachlässigbar, da zunächst die Ausdrücke bearbeitet werden, die wahrscheinlicher zu FALSE ausgewertet werden. (Die speziellen EXISTS-Quantoren sollen hier nicht weiter betrachtet werden, da sie nur eine marginale Veränderung durch evtl. früher möglichen Abbruch des Molekülaufbaus bringen). Falls die Teilmoleküle nicht Bestandteil der Ergebnismoleküle sind, werden die Auswertung und der weitere Aufbau solcher Teilmoleküle unmittelbar nach der Qualifikation des gegebenen Ausdrucks beendet.

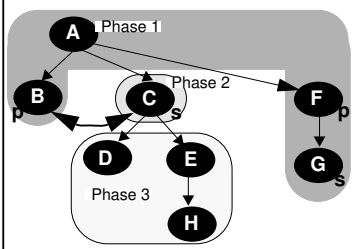
Im Gegensatz zu den existentiellen Quantoren kann bei universell abgeschlossenen Bedingungen jedes einzelne Molekül des AMT zur Disqualifikation des Moleküls führen. Deshalb kann hier der parallele Aufbau aller Moleküle erhebliche unnötige Arbeit verursachen. Aus diesem Grund wird in diesem Fall zunächst eine sequentielle Verarbeitung vorausgesetzt. Andererseits verlangsamen dieser streng sequentielle Aufbau und die Bedingungsauswertung nach jedem einzelnen Molekül des AMT die AEM-Operatorausführung. Um diesem, hier erneut auftretenden Zielkonflikt (schnelle Antwort gegen geringe Ressourcenbelastung) gerecht zu werden, muß hier ein Mittelweg zwischen möglicherweise überflüssigem Aufwand und dem Parallelitätsgrad beim Aufbau der Moleküle des AMT spezifizierbar sein. Dieser hängt von der Wahrscheinlichkeit ab, mit der ein Molekül des AMT zur Disqualifikation führt. Je höher diese Wahrscheinlichkeit für die Disqualifikation ist, desto geringer ist der anzuwendende Parallelitätsgrad.

4.1.3 Beispiele

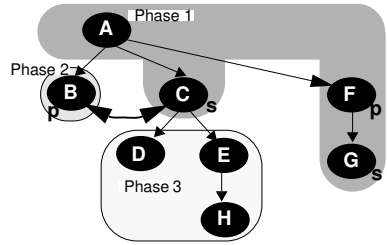
Abschließend werden in Abb. 6 einige Beispiele für die Spezifikation von Top-down-Strategien für den Molekülaufbau im AEM-Operator in Abhängigkeit von den angegebenen Bedingungen gezeigt. Die Indizes s bzw. p an den Atomtypen kennzeichnen, ob die betroffenen Teilmoleküle für die Bedingungsauswertung sequentiell bzw. parallel aufgebaut werden.

Beispiel 1:

SELECT ALL
FROM A-(B, C-(D, E-H), F-G)
WHERE (EXISTS B: B.att1 = 7 **AND**
FOR_ALL C: C.att1 > 1) **OR**



ODER

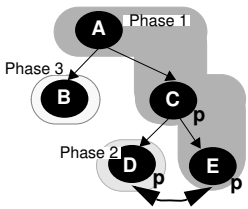


Selektivität von EXISTS B: B.att1 = 7 höher, als die von FOR_ALL C: C.att1 > 1

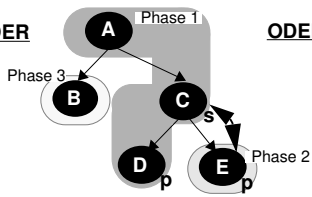
Selektivität von FOR_ALL C:C.att1 > 1 höher, als die von EXISTS B:B.att1 = 7

Beispiel 2:

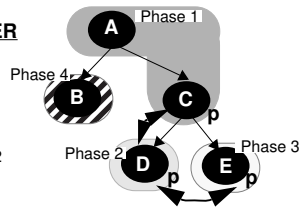
SELECT ALL
FROM A-(B, C-(D, E))
WHERE (EXISTS E: E.att1 = 7 **AND**
FOR_ALL C: **EXISTS** D: D.att1 = 8 **AND**



ODER



ODER



Selektivität von EXISTS E: E.att1=7 am höchsten, dadurch wird der AMT für EXISTS C:C.att1=6 bereits mit aufgebaut, so daß dieser Teil nicht mehr behandelt werden muß.

Selektivität von FOR_ALL C: EXISTS D: D.att1 = 8 am höchsten, dadurch wird der AMT für EXISTS C:C.att1=6 bereits mit aufgebaut, so daß dieser Teil nicht mehr behandelt werden muß.

Selektivität von EXISTS C:C.att1=6 am höchsten, dann FOR_ALL C: EXISTS D: D.att1 = 8 und mit der geringsten Selektivität EXISTS E: E.att1=7.

Abb. 6: Beispiele für Phaseneinteilungen bei Top-down-Strategien

4.2 Bottom-up-Strategie

Eine inhärente Schwäche aller Top-down-Strategien ist der notwendige Molekülaufbau aller Moleküle, bis die gegebene Bedingung entschieden ist, auch wenn sich die Moleküle nicht qualifizieren. Dabei sind immer die Pfade von der Wurzel des Moleküls bis zu den Atomen aufzubauen, die für die Auswertung benötigt werden. Je länger diese Pfade sind und je weniger Atome auf diesen Pfaden für die Auswertung

der Bedingung tatsächlich benötigt werden, umso größer ist der Zusatzaufwand, der für disqualifizierte Moleküle geleistet werden muß. Deshalb soll nun versucht werden, diesen Zusatzaufwand zu minimieren, indem die Menge der möglichen Ergebnismoleküle bereits vorher eingeschränkt wird.

Eine solche Einschränkung ist möglich, wenn man bei den Atomen mit dem Molekülaufbau anfängt, auf denen ein Ausdruck entschieden wird. Von diesen Atomen ausgehend werden die Pfade zu den Wurzelatomen verfolgt. Die so bestimmte Wurzelatommenge schränkt den Suchraum für den weiteren Molekülaufbau ein. Wesentlich für dieses sogenannte Bottom-up-Verfahren ist, daß es mindestens alle Wurzelatome der Ergebnismolekülmenge liefert. Wenn diese Eigenschaft nicht zugesichert werden kann, ist es nicht anwendbar, da dann das Top-down-Verfahren trotz des Bottom-up-Verfahrens noch vollständig ausgeführt werden müßte, um die fehlenden Ergebnismoleküle zu bestimmen. In diesem Abschnitt werden das Verfahren sowie notwendige Voraussetzungen dazu im Detail erläutert.

Die grundlegende Idee ist, zunächst die Atome zu lesen, die eine Einatomtyp-(Teil-)Bedingung erfüllen. Aufgrund der Symmetrie der Referenzen ist es dann möglich, von diesen Atomen ausgehend die Pfade zur Wurzel des gegebenen AEM-MT zu verfolgen. Das Ergebnis ist eine Wurzelatommenge, die eine Teilmenge der existierenden Atome des Wurzelatomtyps ist. Für die Atome dieser Wurzelatommenge müssen anschließend im Top-down-Verfahren die Moleküle um noch nicht gelesene Atome vervollständigt werden. Dabei sind weiter, noch nicht berücksichtigte Bedingungen auszuwerten.

In Abb. 7 wird dieses Verfahren anhand eines sehr einfachen Beispiels illustriert. Unter Ausnutzung eines Zugriffspfades für C.attr1 wird zunächst nur c3 gelesen. In der Bottom-up-Strategie werden über b3 und b4 die Pfade zu a2 und a3 verfolgt, um anschließend die Moleküle mit diesen Wurzeln zu vervollständigen. Ohne das Bottom-up-Verfahren müßten zur Auswertung der Bedingung alle Moleküle vom Molekültyp A-B-C aufgebaut werden, um dann die Bedingung entscheiden zu können.

Die möglichen Vorteile dieses Verfahrens werden durch das Beispiel deutlich. Eine genauere Betrachtung dieses Verfahrens zeigt allerdings, daß es aufgrund einiger Einschränkungen nicht immer sinnvoll bzw. überhaupt nicht anwendbar ist. Die notwendigen Voraussetzungen und die Grenzen des Verfahrens sollen im folgenden erläutert werden. Dazu werden mit **Startatomtyp/Startatomen** der Atomtyp/die Atome bezeichnet, von dem/denen das Bottom-up-Verfahren ausgeht, im Beispiel also C/(c1, .., c4).

- Zunächst hängt die sinnvolle Anwendbarkeit der Bottom-up-Strategie von der Datencharakteristik der zugrundeliegenden Datenbank ab. Falls die Anzahl der Atome vom Startatomtyp sehr hoch ist, ohne durch Zugriffspfade sinnvoll eingeschränkt werden zu können, kann ein Scan über diesen Atomtyp teurer sein, als der Top-down-Aufbau der Moleküle. Dies wäre in Abb. 7 dann der Fall, wenn beispielsweise noch 10 weitere Atome vom Atomtyp C (kurz: C-Atome) existieren würden. Aber auch ein Zugriffspfad würde das Problem nicht lösen, wenn alle Atome im Attribut “attr1” den Wert “wert1” hätten. Auch dann müßten noch alle diese Atome gelesen werden, und ggf. weitere referen-

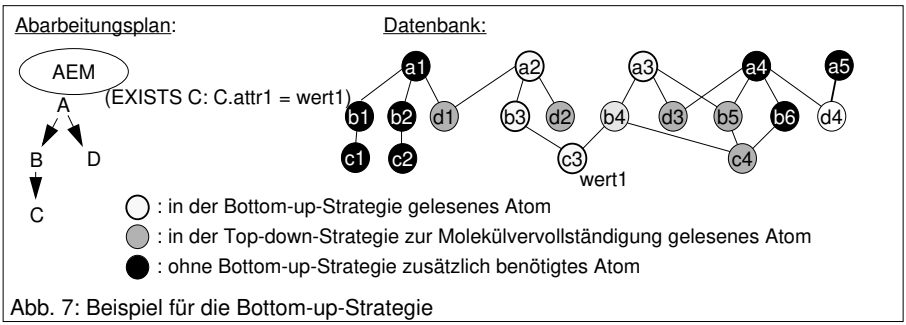


Abb. 7: Beispiel für die Bottom-up-Strategie

zierte B-Atome, selbst wenn es für diese dann keine Referenz mehr zu A-Atomen gibt. In diesem Fall würden folglich wieder zu viele Atome unnötig gelesen.

Um über einen Einsatz der Bottom-up-Strategie zu entscheiden, muß also abgeschätzt werden, wieviel Aufwand mit der Bottom-up-Strategie im Vergleich zum Aufwand für die Top-down-Strategie verbunden ist. Dies ist von den Kardinalitäten der beteiligten Atomtypen, der Anzahl der Referenzen in den beteiligten Referenzattributen und den Werteverteilungen auf den zur Bedingungsauswertung benötigten Attributen abhängig [Sch93].

- Im Beispiel wurde bewußt ein existentieller Abschluß auf C gewählt, weil das der einfachste Fall für das Bottom-up-Verfahren ist. Für den universellen Abschluß ist das Verfahren mit einer leichten Modifikation auch anwendbar. Nachdem alle C-Atome, die die Bedingung erfüllen, gelesen wurden (im Beispiel von Abb. 7 bei Ersetzung des EXISTS durch FOR_ALL: c3), werden alle referenzierten B-Atome (b3, b4) gelesen, aber nur noch diejenigen weiter berücksichtigt, die nur C-Atome aus der bereits gelesenen Menge referenzieren (b3). Dieses Verfahren kann über mehrere Stufen bis zu den Wurzelatomen angewandt werden, so daß nur solche Wurzelatome (vom Typ A) bestimmt werden, von denen keine Startatome erreicht werden, die die Bedingung nicht erfüllen. Dies entspricht der Umformung der Bedingung von (FOR_ALL x: Q(x)) in (NOT EXISTS x: NOT Q(x)).

  Da ein universeller Abschluß aber zur Erfüllung der Bedingung die Existenz entsprechender Atome nicht fordert, werden solche Bedingungen offensichtlich auch von Molekülen erfüllt, die überhaupt keine Startatome enthalten (z.B. a5-d4). Um also eine vollständige Wurzelatommenge durch das Bottom-up-Verfahren in dieser Situation erzeugen zu können, darf in einem Molekül keine Referenz auf dem Pfad vom Wurzelatomtyp zum Startatomtyp leer sein, wie dies bei d4 der Fall ist. D.h., zu jedem Wurzelatom gibt es auch mindestens ein Startatom. Diese Eigenschaft des Pfades wird "sicherer Pfad" genannt. Wäre diese Bedingung nicht erfüllt, dann würde das Bottom-up-Verfahren nicht sicher eine vollständige Wurzelatommenge liefern und wäre somit nicht anwendbar. Die entsprechende Informationen ist in den Metadaten der Datenbank enthalten (Kardinalitätsrestriktionen in der Schemadefinition der Referenzattribute).
- Unabhängig von den gegebenen Quantoren muß in PRIMA gefordert werden, daß die Referenzattribute auf dem Pfad vom Wurzelatomtyp zum AMT keine undefinierten Werte aufweisen dürfen. Diese Eigenschaft wird "definierter Pfad" genannt. Sie ist notwendig, weil PRIMA zwischen einer sicheren Ergebnismenge und einer unsicheren Ergebnismenge unterscheidet [Sch91]. Die sichere Ergebnismenge enthält die Ergebnismoleküle, die die gegebene Bedingung erfüllen, während die Auswertung dieser Bedingung bei den Molekülen der unsicheren Menge nicht entschieden werden konnte ("undefinierter Wert"). Für die Elemente der unsicheren Menge muß der Anwender selber entscheiden, ob sie weiter bearbeitet werden sollen oder nicht. Ohne die Eigenschaft des "definierten Pfades" würden im Bottom-up-Verfahren aber Wurzelatome der unsicheren Menge nicht gefunden (vgl. sicherer Pfad bei universellem Abschluß). Auch diese Informationen ist in den Metadaten enthalten (NOT_NULL-Klausel in der Schemadefinition der Attribute).
- Während der Algorithmus mit leichten Modifikationen auch die Quantoren EXISTS_AT_LEAST und EXISTS_EXACTLY behandeln kann, ist das Bottom-up-Verfahren für EXISTS_AT_MOST Quantoren überhaupt nicht anwendbar. Selbst wenn der Pfad sicher und definiert ist, kann es sein, daß vom Wurzelatom nur solche Startatome referenziert werden, die die Bedingung nicht erfüllen, im Beispiel z.B. das Molekül mit der Wurzel a1. Da solche Wurzelatome durch das Bottom-up-Verfahren aber nicht gefunden werden können, ist das Verfahren hier nicht anwendbar.

Die bisher gezeigten Einschränkungen gelten für einfache, nicht negierte Ausdrücke auf einzelnen Atomtypen. Eine Erweiterung zur Behandlung negierter Ausdrücke ist leicht möglich, da sie nach ein-

fachen Regeln in nicht-negierte Ausdrücke umgeformt werden können (z.B. NOT EXISTS_AT_MOST n zu EXISTS_AT_LEAST n+1).

Neben den bisher betrachteten Ausdrücken können auch komplexere, d.h. konjunktiv oder disjunktiv verknüpften Bedingungen behandelt werden. In diesen Fällen wird die Bottom-up-Strategie für die einzelnen Ausdrücke getrennt angewandt, um so Wurzelatommengen zu bestimmen. Bei Disjunktionen wird dann eine Vereinigung, bei Konjunktionen eine Durchschnittsbildung dieser Wurzelatommengen durchgeführt. Dabei wird für die einzelnen Mengenoperationen die gleiche Reihenfolge eingehalten wie sie bei den entsprechenden logischen Operationen definiert ist. Auf den resultierenden Mengen werden dann die weiteren Abschnitte des Molekülaufbaus im Top-down-Verfahren durchgeführt. Um eine vollständige Wurzelatommenge zu bestimmen, ist bei Disjunktionen deswegen notwendige Voraussetzung, daß alle disjunktiv verknüpften Ausdrücke bottom-up bearbeitet werden.

Weitere Optimierungen sind möglich; sie sollen hier aus Platzgründen aber nur kurz skizziert werden. Zunächst einmal muß die Durchschnittsbildung bei konjunktiv verknüpften Ausdrücken nicht notwendigerweise erst auf der Wurzelatommenge erfolgen. In Abhängigkeit von den gegebenen Ausdrücken kann diese auch bereits auf anderen gemeinsamen Atomtypen auf dem Pfad zum Wurzelatomtyp erfolgen. Bei existentiell quantifizierten Ausdrücken kann das Verfahren zusätzlich dadurch beschleunigt werden, daß die Wurzelatommenge bestimmt werden kann, ohne alle Atome auf dem Pfad dorthin lesen zu müssen. Dabei lassen sich die Symmetrie der Referenzen und ihre Modellierung mittels der Identifier ausnutzen. Diese ermöglichen es, ggf. nur Atome jedes zweiten Atomtyps lesen zu müssen. Sei beispielsweise auf dem Molekültyp A-B-C eine existentielle Bedingung auf C gegeben, seien ferner die betroffenen C-Atome gelesen und somit die referenzierten Identifier für die betroffenen B-Atome bestimmt. Dann müssen die B-Atome nicht unbedingt gelesen werden. Vielmehr können direkt die A-Atome gelesen werden, die Referenzen auf die identifizierten B-Atome besitzen. Dieses Verfahren ist insbesondere dann sinnvoll, wenn die B-Atome nicht mit ausgegeben werden oder wenn in einer anschließenden Top-down-Verarbeitung auch noch Ausdrücke ausgewertet werden müssen, so daß die B-Atome möglicherweise überhaupt nicht benötigt werden. Weitere Details und eine ausführlichere Diskussion dieser Optimierungen finden sich in [St94].

4.3 Vollständiges Verarbeitungsmodell

Die beiden bisher beschriebenen Teilmodelle zum Molekülaufbau werden jetzt in ein allgemeines Ausführungsmodell integriert. Jede Bearbeitung eines AEM-Operators beginnt, sofern vorhanden, mit einer Bottom-up-Strategie, um die Menge der aufzubauenden Moleküle einzuschränken. Daher ist dieses Verfahren immer molekülmengenorientiert, d.h., es werden immer alle Atome eines Atomtyps zusammen angefordert und bearbeitet. Anschließend an das Bottom-up-Verfahren werden in einer in verschiedene Phasen zerlegten Top-down-Strategie die noch notwendigen Schritte zur endgültigen Auswertung der Bedingung ausgeführt. Dieses kann in Abhängigkeit von der durch den Optimierer erfolgten Spezifikation sowohl molekül- als auch molekülmengenorientiert geschehen. Dabei wird ein Molekül aus der Menge der zu bearbeitenden Moleküle entfernt, sobald seine Disqualifikation feststeht. Umgekehrt wird zum phasenunabhängigen parallelen Molekülaufbau übergegangen, sobald die Qualifikation des Moleküls feststeht. Abschließend wird das fertig konstruierte Molekül ausgegeben.

4.4 Verallgemeinertes Verarbeitungsmodell

Die Bottom-up-Strategie im vollständigen Verarbeitungsmodell erweist sich als besonders einfach, wenn als Ausgangspunkte nur einzelne Atomtypen betrachtet werden, auf denen Einatomtyp-Ausdrücke gegeben sind. Eine Verallgemeinerung dieses Verarbeitungsmodells läßt auch komplexere Ausdrücke auf Teilmolekülen zu. Dazu werden diese Teilmoleküle rekursiv unter Anwendung des verallgemeiner-

ten Verarbeitungsmodells aufgebaut, und von den Wurzeln der so erhaltenen Teilmoleküle, die die gegebene Bedingung erfüllen, wieder die Bottom-up-Strategie angewandt. In diesem Fall übernehmen die Wurzelatome dieser Teilmoleküle die Rolle der Startatome für das Bottom-up-Verfahren. Allerdings entfällt bei den Teilmolekülen deren Vervollständigung, da diese erst ausgeführt werden soll, wenn sich das Molekül endgültig qualifiziert hat.

4.5 Zusammenfassung

Die wesentlichen Punkte für einen an die Qualifikationsbedingungen angepaßten Molekülaufbau sollen hier noch einmal zusammenfassend dargestellt werden:

In einem vorgeschalteten Bottom-up-Verfahren kann die mögliche Ergebnismenge eingeschränkt werden. Dieses Verfahren ist nur bei sicheren und definierten Pfaden und nicht bei EXISTS_AT_MOST-Quantoren anwendbar. Darüberhinaus kann es bei entsprechenden Datencharakteristika ungeeignet sein. Das Verfahren wurde zunächst zur Behandlung von zusammengesetzten Einatomtyp-Ausdrücken eingeführt und anschließend zum verallgemeinerten Verarbeitungsmodell für Mehratomtyp-Ausdrücke erweitert. Für einen Top-down-Molekülaufbau wurde ein Phasenmodell entwickelt, das eine bedingungsangepaßte AEM-Operator-Bearbeitung ermöglicht. Dadurch kann unnötige Arbeit bei sich nicht qualifizierenden Molekülen bzw. auf Teilen, die nicht zur Ausgabe gehören, bei sich qualifizierenden Molekülen vermieden werden. Desweiteren können die Moleküle durch Ausnutzung paralleler Verarbeitungsstrategien schnell aufgebaut werden. Hier muß der Optimierer von Fall zu Fall entscheiden, welche Strategie angewandt werden soll. Neben der atomtypspezifischen Festlegung von Phasennummern wurden weitere Ergänzungen (Phasenabhängigkeiten, paralleler vs. sequentieller Teilmolekülaufbau in einer Phase) gemacht, die einen schnelleren an die Bedingung angepaßten Molekülaufbau ermöglichen.

5. Zugriffspfade

Nachdem wir nun in Abhängigkeit von der gegebenen Bedingung eine Strategie für den Molekülaufbau festlegen können, mit der keine unvermeidbare Arbeit geleistet wird, wenden wir uns nun der Frage zu, wie die Ausnutzung von Zugriffspfaden in diese Bearbeitung integriert werden kann. Wie eingangs erwähnt, bietet PRIMA dazu konventionelle Zugriffspfade für den wertabhängigen Zugriff auf einzelne Atome und Atomcluster zum effizienten Lesen ganzer komplexer Objekte an.

5.1 Ausnutzung konventioneller Zugriffspfade

Aus der bisherigen Darstellung des Molekülaufbaus geht hervor, daß der zielgerichtete, wertabhängige Zugriff auf einzelne Atome zu Beginn der Verarbeitung notwendig ist, um so die Menge der aufzubauenden Moleküle effizient einschränken zu können. Dies gilt für Bedingungen auf dem Wurzelatomtyp beim Top-down- und für Bedingungen auf den Startatomtypen beim Bottom-up-Verfahren. Darüber hinaus werden diese Zugriffspfade im Molekülaufbau nicht mehr benötigt, da der weitere Aufbau nur noch aus der Verfolgung von Referenzen über verschiedenen Atomtypen besteht. Das entspricht einem vom Identifier abhängigen wertbasierten Zugriff, der aufgrund seiner Häufigkeit standardmäßig über Zugriffspfade (in PRIMA über Hashverfahren) unterstützt wird.

5.2 Ausnutzung von Atomclustern

Für häufig benötigte Molekültypen wird der Molekülaufbau durch die in Kapitel 2 vorgestellten Atomcluster unterstützt. Die in den Clustern materialisierten Molekültypen entsprechen genau solchen, wie sie im AEM-Operator aufgebaut werden. Durch die zusammenhängende Abspeicherung der Atome eines Moleküls wird die molekülorientierte Bearbeitung des AEM-Operators besonders unterstützt.

Aufgrund des hohen Speicheraufwandes durch die redundante Speicherung der Daten verbietet es sich, für jeden möglichen Molekültyp einen eigenen Clustertyp anzulegen. Deshalb ist genauer zu untersuchen, wie auch Teile von Atomclustertypen für Anfragen genutzt werden können, deren Molekültypen nicht vollständig durch einen Atomcluster materialisiert werden. Dazu betrachten wir die Überdeckungen zwischen dem AEM-MT und dem im Atomcluster materialisierten Clustermolekültyp (**Cl-MT**), d.h., inwieweit sich Teile des AEM-MT im Cl-MT wiederfinden. Dabei kann der Cl-MT natürlich auch noch Teilmolekültypen umfassen, die mit dem AEM-MT nichts zu tun haben. Eine Diskussion aller möglichen Situationen würde hier zuweit führen. Aufgrund folgender Eigenschaften, die der Optimierer bei der Spezifikation des AEM-Operators in einem Abarbeitungsplan zu berücksichtigen hat, läßt sich jedoch allgemein feststellen, daß je genauer der AEM-MT im Cl-MT repräsentiert wird, umso besser kann der Cl-MT ausgenutzt werden (siehe Abb. 8):

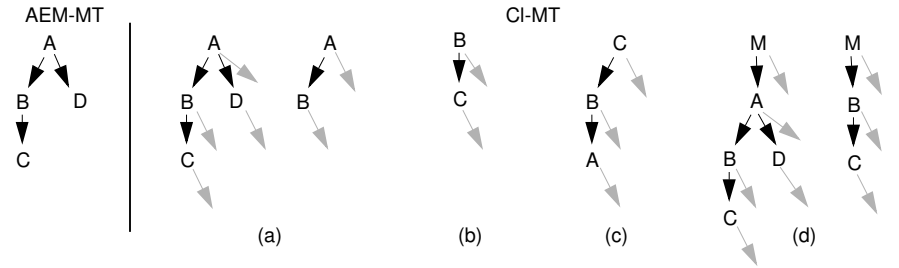


Abb. 8: Beispiele für Überlappungen von AEM-MT mit Cl-MT

- Falls der Cl-MT einen (Teil-) Molekültyp des AEM-MT enthält, wobei der Wurzelatomtyp des AEM-MT dem Wurzelatomtyp des Cl-MT entspricht (Abb. 8a), können die dort enthaltenen (Teil-) Moleküle direkt gelesen werden. Dies ist der denkbar günstigste Fall.
- Falls der Cl-MT einen Teilmolekültyp des AEM-MT ohne dessen Wurzelatomtyp enthält, aber der Wurzelatomtyp des betrachteten Teilmolekültypen dem Wurzelatomtyp des Cl-MT entspricht (Abb. 8b), so können entsprechende Teilmoleküle direkt aus dem Cluster aufgebaut werden. Auch dies ist noch ein sehr günstiger Fall, da er keine weiteren Probleme macht.
- Falls der Cl-MT Teile des AEM-MT enthält, wobei aber die Referenzen in zum AEM-MT umgekehrter Richtung modelliert werden (Abb. 8c), so ist eine Clusternutzung für den Molekülaufbau nur noch bedingt möglich, weil die Atome der aufzubauenden Moleküle nicht mehr vollständig in den Clustern enthalten sein müssen. Allerdings eignen sich diese Strukturen in besonderer Weise für die Bottom-up-Strategie.
- Falls die Wurzel des Cl-MT nicht auch gleichzeitig die Wurzel des (Teil-) Molekültypen des AEM-MT ist, der im Cluster bearbeitet werden soll (Abb. 8d), so enthalten die Cluster nicht mehr alle möglichen (Teil-) Moleküle, sondern nur noch solche, die von der Wurzel des Cl-MT über den Pfad zur Wurzel des (Teil-) Molekültyps erreichbar sind. D.h., die Atomclusternutzung verliert an Bedeutung, weil Teile des AEM-MT außerhalb der Atomcluster gelesen werden müssen.
- Ein Cl-MT, der auch Teilmolekültypen enthält, die nicht im konkreten AEM-Operator benötigt werden (Abb. 8a-d), läßt sich umso besser nutzen, je kleiner die nicht benötigten Teilmoleküle sind. Je mehr unnötige Teilmoleküle ein Cluster enthält, umso mehr unnötige Seitenanforderungen werden während der Leseoperationen auf dem Cluster notwendig.

Bisher wurde die Eignung von Atomclustern nur zur Unterstützung von Lesevorgängen betrachtet. Zusätzlich zeigen sich weitere Auswirkungen auf das bisher vorgestellte Verarbeitungsmodell, so daß die

Spezifikation des Phasenmodells und der Atomclusternutzung nicht unabhängig voneinander erfolgen können. Darauf soll im folgenden kurz eingegangen werden.

Atomclustertypen, die Teile eines oder den ganzen bottom-up zu verfolgenden Pfad materialisieren, ermöglichen ein schnelleres Bottom-up-Verfahren, weil über einen Scan nur noch die Atome am Endpunkt dieses dort materialisierten (Teil-) Pfades gelesen werden müssen. Von diesen ausgehend kann dann das Bottom-up-Verfahren fortgesetzt werden. Die übrigen, auf dem (Teil-) Pfad liegenden Atome müssen nicht mehr gelesen werden.

Darüberhinaus bietet die erweiterte Funktionalität mit der atomclusterinterne Auswertung von Bedingungen auf Molekülen eine wesentliche Erweiterung gegenüber der bisher angebotenen atomspezifischen Bedingungsauswertung im Zugriffssystem. Diese Erweiterung ermöglicht vor allen Dingen eine effiziente Verarbeitung auch von komplexeren Bedingungen zur Bestimmung von Startatomen für das Bottom-up-Verfahren.

Schließlich wirkt sich die Atomclusternutzung auch auf das vorgestellte Phasenmodell aus. Die dort ermittelte Phaseneinteilung erfolgte unter ausschließlicher Berücksichtigung der Struktur der angegebenen Qualifikationsbedingung. Dabei wurde implizit die Annahme gemacht, daß das Lesen von Atomen verschiedener Atomtypen unabhängig von der Phaseneinteilung immer mit dem gleichen Aufwand geschieht. Diese Annahme gilt bei der Ausnutzung von Atomclustern aber nicht mehr. Hier kann es sinnvoll sein, Atome, die bei einer nach den bisherigen Kriterien erstellten Phaseneinteilung nicht direkt in der gleichen Phase benötigt werden, trotzdem in dieser Phase zu lesen, weil die entsprechenden Seiten bei der Clusterverarbeitung bereits vorliegen. Dieses Vorgehen widerspricht zwar der Forderung aus 3.2, nur solche Atome zu lesen, die für den Molekülaufbau und die Bedingungsauswertung benötigt werden, bedeutet aber ggf. wesentlich weniger Zusatzaufwand als das wiederholte Einlesen der Seiten mit diesen Atomen.

6. Parallelität

Die bisherigen Betrachtungen zur Ausführung von AEM-Operatoren sind von Aspekten ihrer parallelen Verarbeitung unabhängig, da über die zeitliche Abfolge der Molekülbearbeitung nichts ausgesagt wurde. Alle Diskussionen zum Aspekt der Parallelität bezogen sich auf Anforderungen an das Zugriffssystem. Deshalb soll dieser Aspekt der parallelen Bearbeitung eines einzelnen AEM-Operators näher betrachtet werden. Dazu gehen wir zunächst auf die Vor- und Nachteile der Parallelität ein, ehe konkrete Realisierungsformen vorgestellt werden.

6.1 Beurteilung paralleler AEM-Operator-Ausführungen

Da AEM-Operatoren häufig sehr umfangreiche Molekülmengen bearbeiten, sind parallele Verarbeitungsstrategien sinnvoll, um kurze Antwortzeiten des Systems zu garantieren. Zwar ist die Belastung im Zugriffssystem deutlich größer als die der AEM-Operator-Verarbeitung, die nur auf Hauptspeicherinformationen zurückgreift, aber es kann davon ausgegangen werden, daß im Zugriffssystem durch parallele Verarbeitungsstrategien mehr Atome produziert werden, als sie von einer einzelnen Ausführungseinheit für den AEM-Operator verarbeitet werden können. Dies liegt unter anderem auch daran, daß der CPU-Bedarf zur Bearbeitung komplexer Objekte, wie sie im AEM-Operator anfällt, deutlich höher ist, als der bei der Bearbeitung von Tupeln in relationalen Systemen (vgl.[SPS90]). Deshalb ist ein paralleler Aufbau der Moleküle oft sinnvoll.

Die parallele Ausführung eines AEM-Operators bedeutet in unserer Umgebung, die aufzubauende Molekülmenge in verschiedenen Ausführungseinheiten des AEM-Operators zu konstruieren. Dabei ist die aufzubauende Molekülmenge eine (in den hier betrachteten Anwendungsklassen typischerweise sehr große) Obermenge der auszugebenden Molekülmenge, weil auch die Moleküle, die sich gemäß der ge-

gebenen Qualifikationsbedingung nicht qualifizieren, zu einem bestimmten Teil aufgebaut werden müssen. Eine parallele Bearbeitung disjunkter Teilmengen dieser aufzubauenden Molekülmenge kann die Antwortzeit für diesen Operator deutlich reduzieren. Allerdings zeigen sich bei der Untersuchung dieses Problems neben den immer auftretenden zusätzlichen Verwaltungskosten weitere Abhängigkeiten zu anderen Parametern der Verarbeitung, z.B. zu den benutzten Speicherungsstrukturen. Diese Abhängigkeiten sollen im folgenden skizziert werden.

Bei der unabhängigen Bearbeitung verschiedener Molekülteilmengen in verschiedenen Ablaufeinheiten eines AEM-Operators entstehen als Folge dieser verschiedenen Ablaufeinheiten unabhängige Aufträge an das Zugriffssystem. Wenn die benutzten Speicherungsstrukturen dann nicht an die Molekülverteilung auf die einzelnen AEM-Operatorausführungen angepaßt sind, kann dies, wie bereits bei der molekülorientierten Verarbeitung beschrieben, zu wiederholten Seitenanforderungen im Zugriffssystem führen, wenn verschiedene, unabhängige Leseoperationen unterschiedliche Atome in derselben Seite anfordern.

Weiterer Zusatzaufwand bei der Ausnutzung von Parallelität kann durch mögliche Überlappungen bei von verschiedenen Molekülen gemeinsam genutzten Teilmolekülen entstehen. Je größer diese Überlappungen sind, umso größer ist auch der Zusatzaufwand durch wiederholten Aufbau der überlappenden Teilmoleküle. Deshalb muß dieses und damit das mehrfache Lesen von Atomen vermieden werden. Da mit Ausnahme der Wurzelatome die Anforderungen über die Referenzen (Identifier) erfolgen, ist ein Erkennen dieser Überlappungen und das Feststellen, ob ein Atom bereits gelesen oder angefordert wurde, relativ einfach. Im Falle einer parallelen Ausführung eines AEM-Operators in mehreren Ablaufeinheiten muß ein Protokoll für die Verwaltung entsprechender Informationsstrukturen eingehalten werden, das natürlich etwas komplizierter als bei der sequentiellen Verarbeitung ist. Dieser Zusatzaufwand entfällt, wenn aufgrund der Datencharakteristik klar ist, daß diese Überlappungen nur in sehr geringem Maße existieren, so daß dieses Protokoll mehr Aufwand verursachen würde als die Mehrarbeit für den wiederholten Teilmolekülaufbau.

6.2 Realisierungsformen

Im folgenden werde bei der Untersuchung, wie Parallelität beim Aufbau eines oder mehrerer Moleküle ausgenutzt werden kann, davon ausgegangen, daß der AEM-Operator während seiner Bearbeitung nur einmal aufgerufen wird und eine a priori unbekannte oder sehr große Molekülmenge aufbaut. Unter diesen Voraussetzungen sind zwei Verfahren zur Ausnutzung von Parallelität denkbar.

Eine naheliegende, wertabhängige Partitionierung der Ergebnismolekülmenge, wobei jede Partition in einem eigenen Operatoraufruf aufgebaut wird, ist dabei nicht sinnvoll, da sie sich wiederum, um die Lokalität der Seitenzugriffe nicht zu zerstören, an den Speicherungsstrukturen orientieren müßte. Um diese Lokalität nicht zu vermindern, müßten die Atome der Moleküle auch in der Basisspeicherungsstruktur nach ihrer Molekülzugehörigkeit zusammenhängend abgespeichert werden, was aufgrund der Zugehörigkeit von Teilmolekülen zu mehreren Ergebnismolekülen nicht möglich ist (siehe 3.2.2). Aus diesem Grund wird die folgende Strategie gewählt:

Nicht die konkrete Ergebnismenge wird partitioniert, sondern der zu bearbeitende Molekültyp wird in verschiedene Teilmolekültypen zerlegt, von denen jeder in einem eigenen Operator realisiert wird. Die einzelnen Ergebnismoleküle werden abschließend zu den Endergebnissen zusammengesetzt (2. in Abb. 9). Dies entspricht einer Zerlegung des AEM-Operators in mehrere AEM- und Verbindungsoperatoren. Dabei erledigten die Verbindungsoperatoren eine Teilaufgabe des AEM-Operators, nämlich Verbindungen zwischen Teilmolekülen herzustellen. So können die üblichen, in [Ge94] beschriebenen Mechanismen zur Ausnutzung von Parallelität genutzt werden. Dies ist zunächst die unabhängige Ausführung der Operatoren (Fall 1 in Abb. 9), die anwendbar ist, wenn fast alle Submoleküle (C-Atome) des abgespaltenen AEM-Operators mit nicht leeren Referenzattributen zum Wurzelmolekültyp tatsächlich benötigt

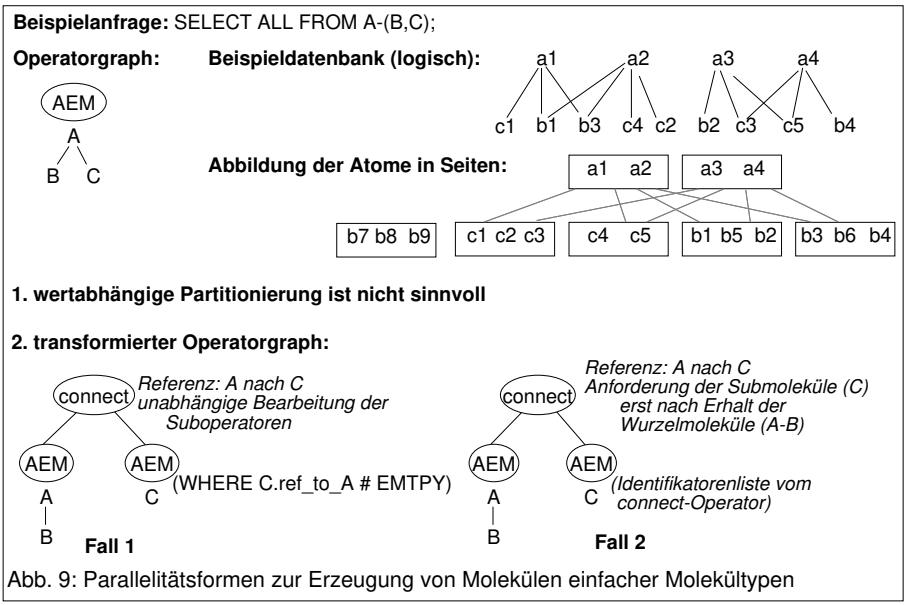


Abb. 9: Parallelitätsformen zur Erzeugung von Molekülen einfacher Molekültypen

werden. Im gegebenen Beispiel gilt diese Voraussetzung. Wenn aber auf dem Molekültyp A-B eine sehr selektive Bedingung angegeben wäre, würden auch nur noch wenige C-Atom benötigt. In diesem Fall bietet sich die zweite Methode zur Ausnutzung von Parallelität, das Pipelining an. Dort werden die Submoleküle (C-Atome) erst nach Erhalt des Wurzelmoleküls im Verbindungsoperator angefordert (Fall 2 in Abb. 9) und während der erste AEM-Operator weitere Wurzelmoleküle erzeugt, produziert der zweite AEM-Operator die anzuhängenden Submoleküle für die bereits hergestellten Wurzelmoleküle.

Bei dieser Zerlegung für den AEM-Operator sind allerdings zwei Einschränkungen zu machen, da die Zerstörung der Molekültypstruktur erheblichen Einfluß auf die Anfrageverarbeitung hat. Zur Bedingungsauswertung benötigte Atomtypen müssen in den AEM-Operatoren auch zusammenhängend bleiben, da sonst die im Phasenmodell beschriebenen Strategien zur frühzeitigen Disqualifikation nicht mehr anwendbar sind. Außerdem müssen Molekültypstrukturen, die in Atomclustern ausgenutzt werden, bestehen bleiben, damit die Nutzung dieser Speicherungsstruktur überhaupt möglich ist.

7. Vollständige Spezifikation des AEM-Operators

Aus dieser Diskussion der für den Aufbau von Molekülen baumartig hierarchischer Molekültypen wesentlichen Faktoren ergibt sich folgendes Modell zur Bestimmung eines geeigneten Verarbeitungsmodells.

Zunächst wird die gegebene WHERE-Bedingung näher untersucht und in Abhängigkeit dieser Bedingung ein Phasenmodell erstellt. Dieses Phasenmodell spezifiziert zunächst unter Berücksichtigung der verfügbaren konventionellen Zugriffspfade und Atomcluster den Aufbau der für die Bottom-up-Verfahren benötigten Teilmolekültypen, die durchzuführenden Bottom-up-Verfahren selbst sowie die Verknüpfung der einzelnen für den Wurzelatomtyp erhaltenen Identifiermengen. Anschließend werden die weiteren, im Top-down-Verfahren notwendigen Abschnitte festgelegt. Dies umfaßt die Beschreibung der Reihenfolge, in der einzelne Atomtypen des AEM-MT behandelt werden, die dabei bestehenden Ab-

hängigkeiten sowie die Festlegung, ob eine molekül- oder eine molekülmengenorientierte Verarbeitung der einzelnen Atomtypen erfolgen soll.

Nachdem dieses Phasenmodell aufgestellt wurde, wird die Ausnutzung von Atomclustern für das Top-down-Verfahren definiert. Dazu werden zunächst die benutzbaren Atomcluster und die relevanten Teile des aufzubauenden Molekültyps identifiziert, ehe deren tatsächliche Nutzung festgelegt und die Phaseneinteilung des Top-down-Verfahrens überarbeitet wird.

Abschließend wird untersucht, ob durch eine Zerlegung des AEM-Operators ein äquivalenter kostengünstigerer Verarbeitungsplan erzeugt werden kann, in dem auch ein paralleler Aufbau verschiedener Molekülteile möglich ist.

8. Zusammenfassung und Ausblick

In der hier präsentierten Arbeit wurden Strategien zum Aufbau komplexer Objekte, wie sie für SODBS benötigt werden, diskutiert. Dabei wurden in der Datenbank vielfältige und komplexe Strukturen wie netzwerkartig und rekursiv verbundene Objekte angenommen, die sich nicht redundanzfrei in hierarchische Strukturen abbilden lassen. Von diesen Annahmen ausgehend, galt es, ein Verarbeitungsmodell zu finden, das es erlaubt, deskriptive, mengenorientierte Anfragen an das Datenbanksystem in optimale Abarbeitungspläne umzuformen. Aufgrund der Komplexität des Problems konzentrierte sich die Arbeit hier auf den Aufbau einfacher, d.h. baumartig hierarchischer Objektstrukturen. Dabei wurde das an der Universität Kaiserslautern entwickelte Non-Standard-Datenbanksystem PRIMA als Beispiel für die in der Diskussion angesprochenen Probleme genutzt. Dort übernimmt der AEM-Operator, als ein spezieller Operator des Verarbeitungsmodells in PRIMA, die vorgestellte Aufgabe.

Bei der Entwicklung des Verfahrens wurde darauf geachtet, daß es allgemein verwendbar ist, also nicht nur auf bestimmte Anfragetypen zugeschnitten optimale Abarbeitungspläne erzeugt. Von sehr einfachen Algorithmen wie z.B. Depth-first-Strategien ausgehend wurden deren Schwächen identifiziert und anschließend durch verbesserte Strategien schrittweise behoben. Grundlegendes Ziel bei der Entwicklung war dabei zunächst, die Verarbeitungsschritte auf ein notwendiges Minimum zu reduzieren. Dazu wurden mögliche, auf den aufzubauenden komplexen Objekten entscheidbare Bedingungen untersucht und eine flexible Top-down-Strategie entwickelt, die an solche Bedingungen angepaßt werden kann. Dieses Verfahren spezifiziert einen zielgerichteten Molekülaufbau, so daß die gegebene Bedingung möglichst früh entschieden werden kann. Eine Schwäche dieses Verfahrens ist der notwendige Aufbau aller Moleküle, also auch solcher, die sich nicht für die Ausgabe qualifizieren. Ein dem Top-down-Verfahren vorgeschaltetes Bottom-up-Verfahren, in dem die resultierende Ergebnismenge eingeschränkt wird, kann dieses Problem lösen. Wie gezeigt wurde, hängt die Anwendbarkeit dieses Verfahrens allerdings von dem gegebenen Schema, der gegebenen Qualifikationsbedingung und der Datencharakteristik der zugrundeliegenden Datenbank ab.

Im Anschluß an die Darstellung dieses flexiblen Verarbeitungsmodells wurden weitere Leistungsgesichtspunkte untersucht. Zunächst ging es um die Integration von Zugriffspfadstrukturen. Es wurde gezeigt, daß neben den konventionellen Zugriffspfaden auch die Atomcluster sowohl in der Top-down- als auch in der Bottom-up-Phase einsetzbar sind, wobei ihre Nutzung allerdings Einfluß auf die Phaseneinteilung hat. Deshalb muß die in der Anfrageoptimierung zunächst festgelegte Phaseneinteilung bei einer Atomclusternutzung evtl. noch modifiziert werden.

Abschließend wurde untersucht, wie die parallele Verarbeitung eines AEM-Operators zu bewerten ist. Hier zeigte sich, daß eine parallele Verarbeitung dieses Operators grundsätzlich möglich, aber nicht immer tatsächlich sinnvoll einsetzbar ist. Eine wertbasierte Partitionierung der Ergebnismenge für deren parallele Bearbeitung im AEM-Operator ist nicht vorteilhaft, weil dazu nicht nur die Wurzelatommenge,

sondern die ganzen Moleküle mit ihren Überlappungen angemessen auf die Speicherungsstrukturen abgebildet werden müßten. Daher erscheint es sinnvoller, den Molekültyp zu zerlegen, einen daran angepaßten Operatorgraphen zu erzeugen und bei dessen Verarbeitung die in [Ge94] vorgestellten Mechanismen zur Ausnutzung von Parallelität in Anspruch zu nehmen.

Die Ergebnisse dieser Arbeit, die hier am Beispiel von PRIMA ermittelt wurden, können auf andere objektorientierte DBMS übertragen werden. Zur einfacheren Systembenutzung benötigen auch diese Systeme deskriptive und mengenorientierte ad-hoc Anfragesprachen, die vor ihrer Ausführung optimiert werden müssen. Dabei ist es unerheblich, ob diese Sprachen eine SQL-ähnliche Notation mit SELECT, FROM und WHERE-Klauseln zur Verfügung stellen [De90, Ca94], oder ob eine andere Notation gewählt wird [Ba89]. Für alle diese Anfragesprachen gilt, daß sie eine Ausdrucksmächtigkeit haben, die die vom AEM-Operator realisierte umfaßt. Ein ausführlicher Vergleich der verschiedenen Sprachen würde hier zuweit führen. Deshalb sei hier nur ein sehr einfaches, selbsterkärendes Beispiel angegeben: Die MQL-Formulierung laute “SELECT Student(name) FROM Student.besucht_kurse-Kurs.gehalten_von-Professor WHERE Professor.fachgebiet = dbs;”. Eine in ihrer Funktionalität ähnliche Formulierung, beispielsweise in OQL von ODMG [Ca94], sieht dann folgendermaßen aus: “select student.name from x in Studenten, y in x.besucht_kurse, z in y.gehalten_von where z.fachgebiet = dbs”. Die für die Bearbeitung der MQL-Anweisung hier vorgestellten Verfahren können somit bei einer entsprechenden Umsetzung der OQL-Anfrage in einen dem AEM-Operator ähnlichen Algebra-Operator dort auch angewandt werden.

In weiterführenden Arbeiten gilt es nun, zu dem hier vorgestellten allgemeinen Verarbeitungsmodell ein Kostenmodell zu erarbeiten, das in Abhängigkeit der verschiedenen Parameter (Phasenzerlegung, Zugriffspfadnutzung, Parallelitätsnutzung) unterschiedliche Verarbeitungspläne für eine Anfrage bewerten kann. Dazu soll das in [Sch93] beschriebene Modell, das sich auf eine einfachere Realisierung des AEM-Operators bezieht, um die Aspekte der Phasenzerlegung und der Parallelität erweitert werden. Mit der Implementierung des hier vorgestellten Verarbeitungsmodells wird dann das Leistungsverhalten verschiedener, semantisch äquivalenter Operatorgraphen gemessen und mit den durch das Kostenmodell erfolgten Abschätzungen verglichen.

9. Literatur

AWS92 Ahmed, S., et al.: Object-oriented database management systems for engineering: A comparison, Journal of Object-Oriented Programming, pp. 27-44, 1992

Ba89 Bancilhon, F.: Query Languages for Object-Oriented Database Systems: Analysis and a Proposal, Proc. Datenbanksysteme in Büro, Technik und Wissenschaft, BTW 1989, Informatik-Fachberichte 204, Springer-Verlag, pp. 1-18, 1989

Be94 Bertino, E.: A Survey of Indexing Techniques for Object-Oriented Database Management System, in: Freytag, J.C., Maier, D., Vossen, G.: Query Processing For Advanced Database System, Morgan Kaufmann Publishers, Inc., pp. 383-418, 1994

Ca94 Cattell, R.G.G. (ed): Object Database Standard: ODMG-93, Morgan Kaufmann Publishers, 1994

De90 Deux, O., et al.: The Story of O2, IEEE Transactions on Knowledge and Data Engineering, Vol. 2, No. 1, pp. 91-108, 1990

Ge94 Gesmann, M.: Parallel Query Execution in Hierarchically Layered Dataflow-Driven Complex Object DBMS, in Vorbereitung

HS91 Heuer, A., Scholl, M.: Principles of Object-Oriented Query Languages, Proc. Datenbanksysteme in Büro, Technik und Wissenschaft, BTW 1991, Informatik-Fachberichte 270, Springer-Verlag, pp. 178-197, 1991

HT94 Hua, K., Tripathy, C.: Object Skeletons: An Efficient Navigation Structure for Object-Oriented Database System, Proc. of the 10th Conf. on Data Engineering, pp. 508-517, 1994

JS90 Jhingran, A., Stonebraker, M.: Alternatives in Complex Object representation : A Performance Perspective, Proc. of the 6th Conf. on Data Engineering, pp. 94-102, 1990

Ki93 Kim, W.: Object-Oriented Database Systems: Promises, Reality and Future, Proc. of the 19th VLDB Conference, pp. 676-687, 1993

KCB87 Khoshafian, S., Valduriez, P., Copeland, G.: Parallel Query Processing for Complex Objects, Proc. of the 4th Conf. on Data Engineering, pp. 202-208, 1988

KD91 Keßler, U., Dadam, P.: Auswertung komplexer Anfragen an hierarchisch strukturierte Objekte mit Pfadindexen, Proc. Datenbanksysteme in Büro, Technik und Wissenschaft, BTW 1991, Informatik-Fachberichte 270, Springer-Verlag, pp. 218-237, 1991

KGM91 Keller, T., Graefe, G., Maier, D.: Efficient Assembly of Complex Objects, Research Report, University of Coloardo at Boulder, CU-CS-502-90, 1991

KKW88 Kim, K.-C., et al.: Acyclic Query Processing in Object-Oriented Databases, Proc. 7th Int. Conf. on the Entity/Relationship Approach, Roma, Italy, pp. 193-210, 1988

KM90 Kemper, A., Moerkotte, G.: Advanced Query Processing in Object Bases Using Access Support Relations, Proc. of the 16th VLDB Conference, pp. 290-301, 1990

KVC88 Kim, W., Chou, H.-T., Banerjee, J.: Operations and Implementation of Complex Objects, Proc. of the 3rd Conf. on Data Engineering, pp. 626-633, 1987

Mi88 Mitschang, B.: Ein Molekül-Atom-Datenmodell für Non-Standard-Anwendungen, Informatik-Fachberichte 195, Springer-Verlag, 1988

MPP93 Mitschang, B., et al.: SQL/XNF - Processing Composite Objects as Abstractions over Relational Data, Proc. of the 9th Conf. on Data Engineering, pp. 272-282, 1993

PSS87 Paul, H.-B., et al.: Architecture and Implementation of the Darmstadt Database Kernel System, Proc. of the ACM/SIGMOD Conf., pp 196-207, 1987

Sch91 Schöning, H.: Praktische Behandlung von Nullwerten - Realisierung im Molekül-Atom-Datenmodell, Proc. Datenbanksysteme in Büro, Technik und Wissenschaft, BTW 1991, Informatik-Fachberichte 270, Springer-Verlag, pp. 502-507, 1991

Sch93 Schöning, H.: Anfrageverarbeitung in Komplexobjekt-Datenbanksystemen, Deutscher Universitäts-Verlag, 1993

Sch94 Schöning, H.: Optimization of Complex-Object Queries in PRIMA - Statement of Problems, in: Freytag, J.C., Maier, D., Vossen, G.: Query Processing For Advanced Database System, Morgan Kaufmann Publishers, Inc., pp. 99-120, 1994

St94 Stratmann, O.: Realisierung verschiedener Strategien zum Aufbau einfacher Moleküle in PHOENIX, Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 1994

SAB94 Steenhagen, H., et al.: From Nested-Loop to Join Queries in OODB, Proc. of the 20th VLDB Conference, pp. 618-629, 1994

SPS90 Schek, H.J., et al.: The DASDBS Project: Objectives, Experiences, and Future Prospects, IEEE Trans. on Knowledge and Data Engineering, Vol. 2, No. 1, pp. 25-43, 1990

SS89 Schöning, H., Sikeler, A.: Cluster Mechanisms Supporting the Dynamic Construction of Complex Objects, Proc. 3rd Int. Conf. on Foundations of Data Organization and Algorithms, FODO ‘89, pp. 41-46, 1989

SS90 Schöning, H., Sikeler, A.: Design of Storage Schemes for Enhanced Database Management Systems, SFB-Bericht SFB 124-25/90, Universität Kaiserslautern, 1990

TRS93 Teeuw, W. et al.: An Evaluation of Physical Disk I/Os for Complex Object Processing, Proc. of the 9th Conf. on Data Engineering, pp. 363-371, 1993