Strategien zum dynamischen Aufbau komplexer Objekte in der Anfrageverarbeitung

Michael Gesmann
Universität Kaiserslautern
email: gesmann@informatik.uni-kl.de

Kurzfassung

Strukturell objektorientierte Datenbanksysteme bieten die Möglichkeit zur direkten Modellierung und Verarbeitung komplexer Objekte, die sich aus elementaren Objekten zusammensetzen. Ein wesentlicher Kostenanteil bei der Bearbeitung komplexer Objekte besteht aus der Interpretation und dem Verfolgen von Referenzen, welche 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 am Beispiel des Non-Standard-Datenbanksystems PRIMA genauer untersucht. Ausgehend von sehr einfachen, aber auch ineffizienten Algorithmen wird ein erweitertes Verarbeitungsverfahren aufgestellt, das flexibel auf unterschiedliche Datenbankzustände und Anfrageformulierungen reagieren kann, um so eine optimale Ausführung zu gewährleisten. In dieses Verfahren werden unterschiedliche Aspekte der Zugriffspfadunterstützung, insbesondere die Clusterung komplexer Objekte, integriert. Darüber hinaus werden Probleme bei der parallelen Bearbeitung der Objekte untersucht.

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].

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 Algorithmen zum Aufbau komplexer Objekt aus elementaren Basisobjekten im SODBS 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 durch parallele Verarbeitungsstrategien eine optimale Bearbeitung des AEM-Operators ermöglicht. Abschließend erfolgen eine Zusammenfassung unserer Ergebnisse sowie 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 verfeinert und konkretisiert. Zusätzlich berücksichtigen wir hier die parallele Abarbeitung eines AEM-Operators. 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 an die hier vorgestellten Erweiterungen angepaßt werden.

Im Gegensatz zum DASDBS [SPS90] werden hierarchische Strukturen in PRIMA nicht automatisch auch in den zugrundeliegenden Speicherungsstrukturen zusammenhängend abgespeichert. Während DASDBS ittels einer Single-Scan Schnittstelle versucht, hierarchische Objekte, mit einem einzigen Scan über diesen Speicherungsstrukturen zu lesen, kann diese Eigenschaft in PRIMA 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.

Eine Reihe von Arbeiten hat sich mit Speicherungsstrukturen, Zugriffspfaden und deren Anwendung bei der Verarbeitung komplexer Objekte beschäftigt [KM90, Be94, KVC88, JS90, TRS93, HT94, KD91]. Im Gegensatz zu diesen Arbeiten geht 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. 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. Im Vergleich zu [KGM91] und [KCB87], die ebenfalls Strategien zum Aufbau komplexer Objekte beschreiben, beruht unser System auf einer weitgehend datenflußgesteuerten Architektur, die insbesondere den parallelen Aufbau komplexer Objekte ermöglicht.

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], entfällt hier. In der Anfrageübersetzung und -optimierung wird direkt aus der MQL-Anfrage ein Operatorgraph mit den AEM-Operatoren als Basisoperatoren erzeugt. In dieser Arbeit wird diskutiert, welche Möglichkeiten dabei für die Spezifikation des AEM-Operators bestehen sollen.

2. PRIMA

In diesem Abschnitt wird ein kurzer Überblick über die für die folgende Diskussion wesentlichen Konzepte des PRIMA-Systems gegeben.

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 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 und dessen Kanten dem resultierenden Datenfluß zwischen den Operatoren entsprechen [Sch93]. 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.

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 Identifiern kann die Ergebnismenge von Leseanforderungen eingeschränkt werden. Neben einer **Basisspeicherungsstruktur**, die alle Atome eines Atomtyps zusammenhängend verwaltet (siehe Abb. 1), werden auch Zugriffspfadstrukturen wie **B*-Bäume** und **Sortierordnungen** zum effizienteren Lesen von Atomen angeboten. Eine 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 im allgemeinen nicht möglich. Die dazu notwendige eindeutige Zuordnung eines Atoms zu einem Molekül ist in der Regel nicht gegeben ist, da ein Atom über verschiedene (oder gleiche) Referenzattribute zu unterschiedlichen Molekülen gehören kann.

Deshalb können für den effizienten Zugriff auf ganze Moleküle einfacher Molekültypen als zusätzliche Zugriffspfadstruktur **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. Vergleichbar mit der Single-Scan-Schnittstelle des DASDBS-Projektes können mit diesen Operationen ganze Moleküle durch einen einzigen Scan über diese Speicherungsstruktur gelesen werden.

3. Der AEM-Operator

In diesem Abschnitt wird zunächst die Funktionalität des AEM-Operators vorgestellt. Anschließend wird ein erstes allgemeines Verarbeitungsmodell für denAEM-Operator eingeführt. Die konzeptionellen Schwächen dieses Modells werden in den folgenden Kapiteln behoben.

Der bereits im 2. Kapitel kurz vorgestellte AEM-Operator hat die Aufgabe, Moleküle aufzubauen, die durch Anfragen vom Typ “SELECT S(M) FROM M WHERE Q(M)” beschreiben werden. Dabei beschreiben M einen einfachen Molekültyp (**AEM-MT**), Q(M) eine auf Molekülen diesen Molekültyps entscheidbare Bedingung und S(M) die zu erfolgende Projektion. 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

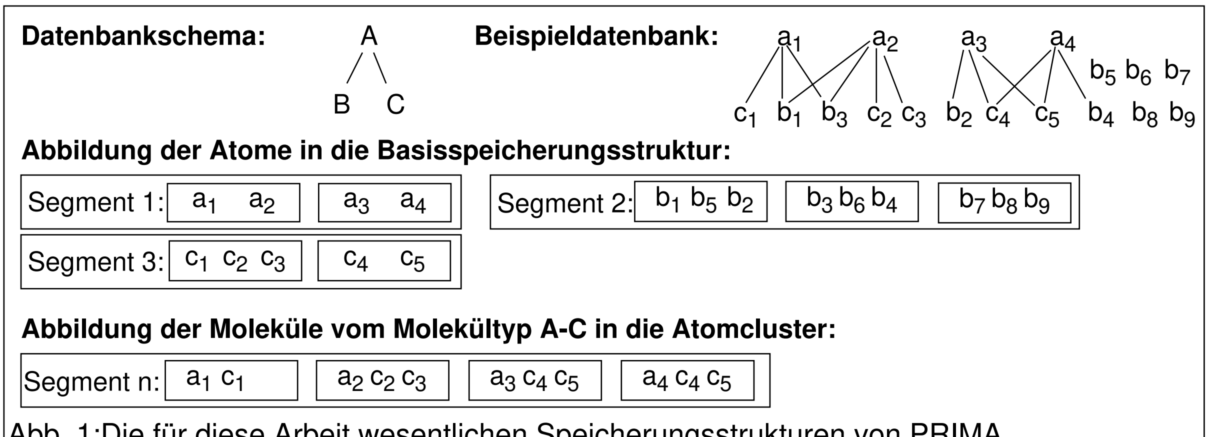


Abb. 1:Die für diese Arbeit wesentlichen Speicherungsstrukturen von PRIMA

die Bedingungsauswertung bzw. für den Aufbau der sich qualifizierenden Ergebnismoleküle benötigt werden.

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

- Die gesamte Molekülmenge soll möglichst schnell vollständig bestimmt werden, um eine möglichst kurze Antwortzeit für Anfragen zu erhalten.
- Die einzelnen Moleküle sollen möglichst schnell aufgebaut werden, damit sie in komplexeren Operatorgraphen durch ein Pipelining in den folgenden Operatoren möglichst früh zur weiteren Bearbeitung verfügbar werden [Sch93]. Für diese Art der Parallelität in der Anfrageverarbeitung ist es wichtig daß die AEM-Operatoren gleichmäßige Eingabeströme für die folgenden Operatoren erzeugen.

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.

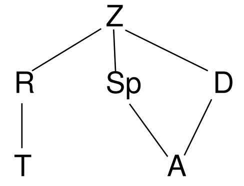
Es sei ein stark vereinfachter Ausschnitt aus einer Entwurfsdatenbank für den VLSI-Chip-Entwurf gegeben (siehe Abb. 2). Die Funktionalität einer Zelle Z wird durch ihre Spezifikation Sp formal beschrieben. Eine Dokumentation D enthält eine Beschreibung des Entwurfsprozesses. Sowohl die Spezifikation 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. Auf dieser Datenbank werde eine einfache Anfrage gestellt, die alle 2-Bit Addierer zusammen mit ihrer Dokumentation, Spezifikation und 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
  Einerseits können benötigte Atome vom Zugriffssystem einzeln, d.h. atomorientiert, angefordert werden. Andererseits berücksichtigen atommengenorientierte Algorithmen immer die Gesamtheit oder eine Teilmenge der zu lesenden Atome eines Atomtyps in einer Anforderung.
- Bearbeitungsgranulat
  Molekülorientierte Algorithmen bearbeiten aufzubauende Moleküle isoliert voneinander. Molekülmengenorientierte Verfahren betrachten gleichzeitig mehrere oder sogar alle Moleküle.
- Verarbeitungsreihenfolge
  Aufgrund der baumartigen Molekültypstrukturen können die üblichen Depth-first- und Breadth-first-Strategien unterschieden werden.

Die vorgestellten Kriterien sind zueinander orthogonal und allgemeingültig, d.h., sie können für jede konkrete Operatorspezifikation unabhängig voneinander betrachtet werden und sind für

Datenbankschema:



Anfrage:

```
SELECT ALL
FROM Z-(R-T, Sp, D)
WHERE (Z.name = 2-bit-add)
      AND
      (T.bezeichnung = mosfet);
```

Abarbeitungsplan:

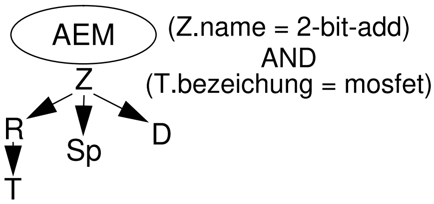


Abb. 2: Beispiel für einen AEM-Operator

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.

Allein nach diesen Kriterien entworfene Verarbeitungsstrategien genügen allerdings nicht aus, um ein gutes Leistungsverhalten zu erzielen. Dazu sind folgende Erweiterungen notwendig:

Die in der WHERE-Klausel spezifizierten Restriktionen, die bisher erst nach einem vollständigen Molekülaufbau ausgewertet werden, müssen besser berücksichtigt werden. Um den Molekülaufbau auf die unbedingt notwendigen Schritte einschränken zu können, sollte es möglich sein, den für die Bedingungsauswertung benötigten Ausschnitt der Moleküle beschreiben zu können, so daß bis zur endgültigen Qualifikation eines Moleküls nur dieser Ausschnitt berücksichtigt werden muß. Bei komplexeren Bedingungen kann es darüber hinaus sinnvoll sein, eine Reihenfolge festzulegen, in der Teilmoleküle aufgebaut werden. So erreicht man eine möglichst frühzeitige Disqualifikation des aktuell bearbeiteten Moleküls. Eine detaillierte Untersuchung hierzu erfolgt in Kapitel 4.

Selbstverständlich sollte es möglich sein, die Nutzung konventioneller Zugriffspfade wie auch der Atomcluster spezifizieren zu können. Dies wird in Kapitel 5 näher untersucht.

Um ein gutes Antwortzeitverhalten erreichen zu können, ist es notwendig, parallele Verarbeitungsstrategien zu nutzen. Kapitel 6 betrachtet diesen Problemkreis genauer.

4. Strategien zum Molekülaufbau

Dieser Abschnitt zeigt, wie an Bedingungen angepaßte Verarbeitungsalgorithmen einen effizienten Molekülaufbau ermöglichen. Grundsätzlich ist dabei folgender Zielkonflikt zu lösen. Einerseits soll möglichst wenig unnötige Arbeit geleistet werden, wenn sich Moleküle nicht qualifizieren, d.h. Teile, die nicht notwendig 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. Diese werden im folgenden Ausdrücke genannt, wenn nicht unterschieden wird, ob es sich um die ganze Bedingung oder um eine Teilbedingungen handelt. **Einatomtyp-Ausdrücke** sind auf einem einzelnen Atomtyp im AEM-MT entscheidbar, z.B. (Atomtyp1.Attribut1 > 3) oder (EXISTS Atomtyp1). **Mehratomtyp-Ausdrücke** sind nicht auf Atomen eines einzelnen Atomtyps entscheidbar, sondern betreffen mehrere Atomtypen, z.B. (Atomtyp1.Attribut1 > Atomtyp2.Attribut1). Für alle Bedingungen gilt notwendigerweise, daß sie auf dem AEM-MT auswertbar sind. Sie werden existentiell abgeschlossen, d.h. vom Anwender nicht quantifizierte Atomtypen werden 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. 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. 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

Die einfachen Strategien zum Molekülaufbau können das unnötige Lesen für die Bedingungsauswertung nicht benötigter Atome in Abhängigkeit von der Bedingung nicht vermeiden. Aus diesem Grund wird nun ein 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. Aufgrund des vom Wurzelatomtyp zu den Blättern gerichteten Molekülaufbaus wird dieses Verfahren Top-down-Strategie genannt.

Im Gegensatz zu den einfachen Strategien wird die Bearbeitungsreihenfolge für die einzelnen Atomtypen jetzt aber explizit durch eine Reihenfolgenummer im Molekültypgraphen festgelegt. Atomtypen mit der gleichen Nummer können gleichzeitig angefordert werden. Somit kennzeichnen die Nummern verschiedene Phasen des Molekülaufbaus. Eine Phase mit der Phasennummer (i+1) darf erst dann beginnen, wenn die Phase i beendet ist. Dabei ist eine Phase i dann beendet, wenn keine Atome von Atomtypen mit der Phasennummer i 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 qualifizierten Ergebnismoleküle, bedingungsunabhängig fertiggestellt werden.

Ausgehend von der Wurzel, die die Phasennummer 1 bekommt, wird so eine Reihenfolge festgelegt, in der die Atomtypen bearbeitet werden. Dabei darf die Phasennummer eines referenzierenden Atomtyps im Molekültyp nie größer sein, als die irgendeines seiner referenzierten Atomtypen. Zur Illustration enthält Abb. 3 für den einfachen AEM-MT aus unserem Beispiel verschiedene Verarbeitungsreihenfolgen, 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-add) erfüllen. In der anschließenden 2. Phase werden nur noch für diese Wurzelatome die Teilmoleküle R-T aufgebaut und der zweite 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.

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 Konjunktionen bzw. Disjunktionen von Ausdrükken Q_i besteht. Zu jedem 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. Deshalb spielt eine Normalisierung der Bedingungen in eine 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.

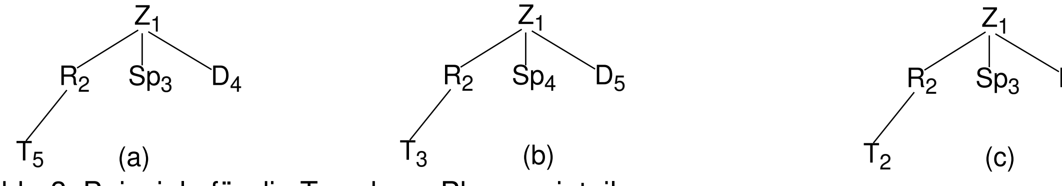


Abb. 3: Beispiele für die Top-down-Phaseneinteilungen

Bei einer Konjunktion von Ausdrücken kann die Auswertung abgebrochen werden, wenn einer der Ausdrücke nicht erfüllt ist. Es ist also 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 sollen, muß der Optimierer für jeden Q_i dessen Selektivität S_{Qi}, also die Wahrscheinlichkeit, mit der Q_i zu FALSE ausgewertet wird, und die Kosten $K_{wAMTi'}$ für den Aufbau der Teilmoleküle vom $wAMT_i'$ (z.B. die Anzahl der zu lesenden Seiten) betrachten. Dabei ist zu berücksichtigen, daß in vorherigen Phasen bereits Teile des $wAMT_i$ aufgebaut worden sein können. Eine alleinige Berücksichtigung der Selektivitäten genügt nicht, da für die einzelnen Teilbedingungen in der Regel unterschiedlich große $wAMT_i$ aufgebaut werden müssen. Dies kann dazu führen, daß es günstiger ist, eine Teilbedingung mit geringerer Selektivität, die 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 $(S_{Qi}/K_{wAMTi'}) > (S_{Qj}/K_{wAMTj'})$. Die Bestimmung der S_{Qi} und $K_{wAMTi'}$ soll hier nicht weiter beschrieben werden, sie ist in [Sch93] enthalten.

Bei Disjunktionen ist diese Behandlung wesentlich einfacher. Da eine Bedingung erst dann zu FALSE entschieden wird, wenn alle Ausdrücke zu FALSE ausgewertet wurden, müssen dazu alle $wAMT_i$ aufgebaut werden. Aus diesem Grund können alle Ausdrücke und AMT_i parallel betrachtet werden. Selbstverständlich kann eine weitere Bedingungsauswertung entfallen, sobald ein Ausdruck zu TRUE ausgewertet wurde. Bei diesem Vorgehen kann allerdings noch unnötige Arbeit verrichtet werden, wenn in der Bedingungsauswertung vorkommende Molekülteile nicht projiziert werden. 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 bzw. Qualifikation früh entschieden werden und danach nur noch die auszugebenden Molekülteile aufzubauen sind. Allerdings erweist sich die strikte Trennung der Phasen, also die Ausführung von Phase i+1 erst nach vollständiger Beendigung der Phase i, als 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. Beispielsweise sei 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 höheren Berechnungsaufwand als Q_1 und Q_2 zusammen. Schließlich sollen alle für die Bedingungsauswertung benötigten Teilmoleküle auch projiziert werden. Mit den bisher vorgestellten Kriterien 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 also gewartet, bis auch Q_3 beendet ist. Dies ist aber nicht notwendig, da Q_2 schon nach Beendigung von Q_1 bearbeitet werden kann. Um dieses zu ermöglichen, werden diese Präzedenzbeziehungen (im Beispiel Q_2 nach Q_1) 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 der in Abb. 4 dargestellte rekursive Algorithmus für die Phaseneinteilung in einem AEM-Operator. Eingabe sind der aktuell betrachtete Ausdruck Q sowie die Phasennummer, mit der dieser Ausdruck begonnen wird. Der Algorithmus vermerkt bei jedem Atomtyp im AMT dessen Phasennummer sowie die 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 im wAMT der vollständigen Bedingung vorkommen. Diese abschließende Phase entspricht dem Vervollständigen der qualifizierten Moleküle um die noch fehlenden Atome. Zu beachten sind mögliche Überlappungen bei solchen Atomtypen, die in mehreren $wAMT_i$ auftreten. In diesen Fällen werden die Atome in der Phase mit der niedrigsten auftretenden Phasennummer gelesen und zur Bedingungsauswertung benutzt (siehe Abb. 5 Beispiel 2).

```
PROCEDURE Definiere_Phaseneinteilung (      Q              : Bedingung;
                                            phn            : Phasennummer;
                                        VAR maximale_phn   : Phasennummer);
BEGIN
   IF (Q nicht weiter zerlegbar) THEN
      alle Atomtypen des zu Q gehörenden wAMT werden in Phase phn bearbeitet, sofern sie nicht
         durch 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 (* Q zerlegbar in UND-verknüpfte Qi *)
      für alle Qi:
         wähle Qj aus noch nicht behandelten Qi, so daß (SQj/KwAMTj') maximal;
         Definiere_Phaseneinteilung(Qj, phn, maximale_phn);
         kennzeichne Reihenfolge mit vorhergehendem wAMTj';
         phn := maximale_phn + 1;
   END;
END Definiere_Phaseneinteilung;
```

Abb. 4: Algorithmus zur Phaseneinteilung für den Top-down-Molekülaufbau

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. Sie können 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, wenn Seiten mit Atomen für verschiedene Moleküle mehrfach angefordert werden, ist es deshalb wichtig, die Anzahl der Leseoperationen je Atomtyp zu minimieren. Falls also mehrere Atome referenziert werden, sollen nur die unbedingt benötigten gelesen werden. Dazu betrachten wir die Quantoren auf den betroffenen Atomtypen in einem Q_i.

Bei universell abgeschlossenen Ausdrücken Q_i kann jedes einzelne Molekül des AMT_i 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 wieder von der Selektivität der Bedingung ab; je höher sie ist, desto geringer ist der anzuwendende Parallelitätsgrad.

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 mit einer höheren Selektivität ausgewertet werden. (Die speziellen EXISTS-Quantoren werden hier nicht weiter betrachtet, 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.

4.1.3 Beispiele

Abschließend werden in Abb. 5 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. Der Einfachheit wegen sei angenommen, daß die Kosten für den Aufbau der $wAMT_i$ immer gleich seien.

4.2 Bottom-up-Strategie

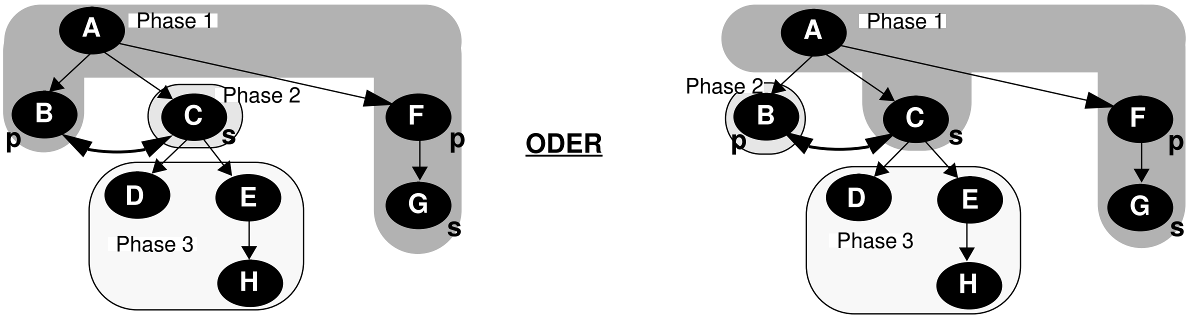
Eine inhärente Schwäche des Top-down-Molekülaufbaus ist, daß immer für alle Moleküle die Pfade von der Wurzel des Moleküls bis zu den Atomen, die für die Auswertung benötigt werden, aufgebaut werden müssen. Je länger diese Pfade sind und je weniger Atome auf diesen Pfaden für die Auswertung der Bedingung tatsächlich benötigt werden, umso größer ist der Zusatzaufwand, der für sich nicht qualifizierende Moleküle geleistet werden muß. Deshalb ist es sinnvoll, die Menge der möglichen Ergebnismoleküle bereits vorher einzuschränken.

Eine solche Einschränkung ist möglich, wenn man bei solchen Atomen mit dem Molekülaufbau anfängt, auf denen ein Ausdruck entschieden wird. Von dort 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 liefern muß. Wenn diese Eigenschaft nicht zugesichert werden kann, ist es nicht sinnvoll 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 dieses 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

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**
(FOR_ALL G: **EXISTS** F: G.att1 > F.att1 **);**

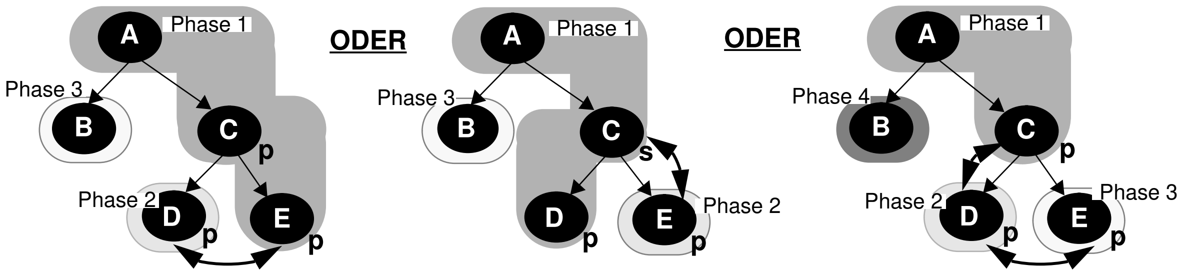


Selektivität von EXISTS B: B.att1 = 7 höher, als die von FOR_ALL C: C.att1 > 1

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**
EXISTS C: C.att1 = 6**);**



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. 5: Beispiele für Phaseneinteilungen bei Top-down-Strategien

die Atome dieser Wurzelatommenge müssen anschließend im Top-down-Verfahren die Moleküle um noch nicht gelesene Atome vervollständigt werden und noch nicht berücksichtigte Ausdrücke ausgewertet werden.

In Abb. 6 wird dieses Verfahren anhand eines 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 diese Moleküle zu vervollständigen. Ohne das Bottom-up-Verfahren müßten zur Auswertung der Bedingung alle Moleküle vom Molekültyp A-B-C aufgebaut werden.

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).

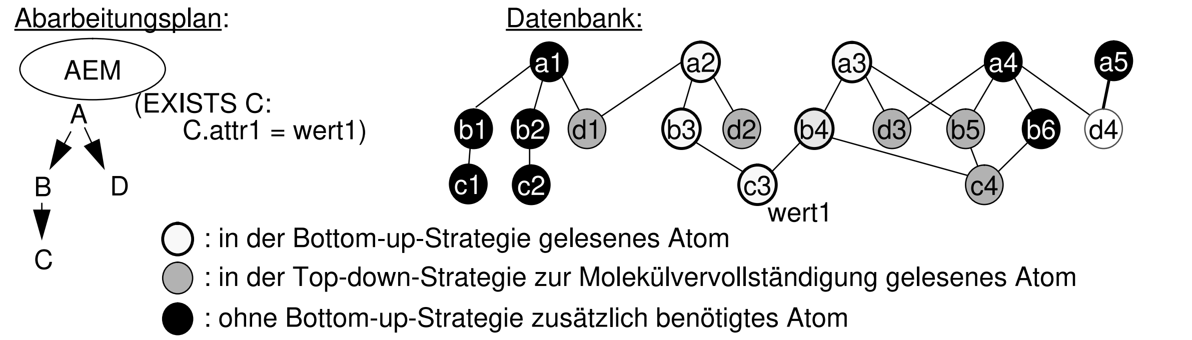


Abb. 6: Beispiel für die Bottom-up-Strategie

- 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. 6 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 referenzierte B-Atome, selbst wenn es für diese dann keine Referenzen zu A-Atomen gibt.

  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 Verteilung der Referenzen in den beteiligten Referenzattributen und den Werteverteilungen auf den zur Bedingungsauswertung benötigten Attributen abhängig [Sch93].
- 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. 6 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 werden nur solche Wurzelatome bestimmt, 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 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 notwendigen Informationen sind 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 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 einfachen 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 für jeden Ausdruck eine Wurzelatommenge 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 in der sie bei den entsprechenden logischen Operationen definiert sind. Auf der resultierenden Menge werden anschließend die weiteren Abschnitte des Molekülaufbaus im Top-down-Verfahren durchgeführt. Um eine vollständige Wurzelatommenge zu bestimmen, ist bei Disjunktionen wiederum 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 Ausdrükken 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 jeden 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 möglich und sinnvoll, 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.

Die Bottom-up-Strategie 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 beschriebenen 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.

5. Zugriffspfade

Nun wenden wir uns der Frage zu, wie die in Kap. 2 vorgestellten Zugriffspfade in der bisher vorgestellte Verarbeitung genutzt werden können.

5.1 Ausnutzung konventioneller Zugriffspfade

Aus der bisherigen Darstellung des Molekülaufbaus geht hervor, daß der wertabhängige Zugriff auf einzelne Atome über Zugriffspfade nur 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 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 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 kann der Molekülaufbau durch Atomcluster unterstützt werden. Durch die zusammenhängende Abspeicherung der Atome eines Moleküls wird dabei die molekülorientierte Bearbeitung im AEM-Operator 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

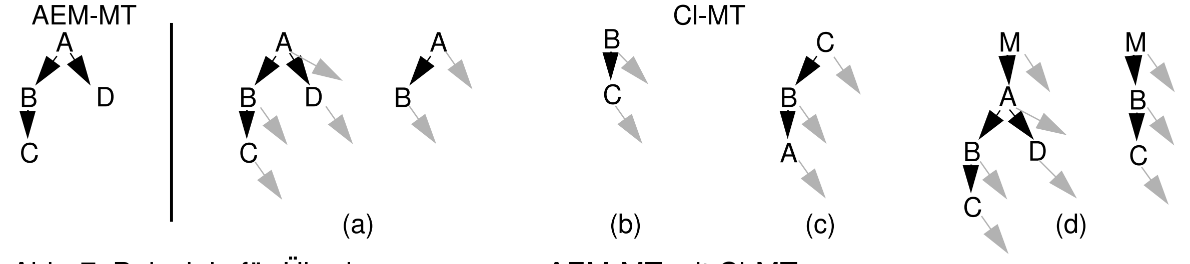


Abb. 7: Beispiele für Überlappungen von AEM-MT mit Cl-MT

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 Teile des AEM-MT im Cl-MT enthalten sind. 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. Zusammenfassend läßt sich jedoch feststellen, daß je genauer der AEM-MT im Cl-MT repräsentiert wird, umso besser kann der Cl-MT ausgenutzt werden (siehe Abb. 7):

- Falls der Cl-MT einen (Teil-) Molekültyp des AEM-MT enthält, wobei der Wurzelatomtyp des AEM-MT dem Wurzelatomtyp des Cl-MT entspricht (Abb. 7a), 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 des Cl-MT entspricht (Abb. 7b), können entsprechende Teilmoleküle direkt aus dem Cluster aufgebaut werden.
- Falls der Cl-MT Teile des AEM-MT enthält, wobei aber die Referenzen im Cl-MT in umgekehrter Richtung modelliert werden (Abb. 7c), ist eine Clusternutzung nur noch bedingt möglich, weil die Atome der aufzubauenden Moleküle nicht mehr vollständig in den Clustern enthalten sein müssen. Allerdings eignen sich diese Strukturen für die Bottom-up-Strategie.
- Falls die Wurzel des Cl-MT nicht gleichzeitig die Wurzel des (Teil-) Molekültypen des AEM-MT ist, der im Cluster bearbeitet werden soll (Abb. 7d), so enthalten die Cluster nicht mehr alle (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 ggf. außerhalb der Atomcluster gelesen werden müssen.
- Ein Cl-MT, der auch Teilmolekültypen enthält, die nicht im konkreten AEM-Operator benötigt werden (Abb. 7a-d), läßt sich umso besser nutzen, je kleiner die nicht benötigten Teilmoleküle sind. Unnötige Teilmoleküle im Cluster führen zu unnötigen Seitenanforderungen während der Leseoperationen auf dem Cluster.

Bisher wurde nur die Eignung von Atomclustern zur Unterstützung von Lesevorgängen betrachtet. Bei ihrem tatsächlich Einsatz zeigen sich folgende Auswirkungen auf das bisher vorgestellte Verarbeitungsmodell:

(1) Die erweiterte Funktionalität mit der atomclusterinterne Auswertung von Bedingungen auf Molekülen ermöglicht vor allen Dingen eine effiziente Verarbeitung auch von komplexeren Bedingungen, z.B. zur Bestimmung von Startatomen für das Bottom-up-Verfahren.

(2) Atomclustertypen, die Teile eines oder den ganzen bottom-up zu verfolgenden Pfad materialisieren, ermöglichen ein schnelleres Bottom-up-Verfahren, weil als Startatome nur noch die Atome am Endpunkt dieses dort materialisierten (Teil-) Pfades gelesen werden müssen.

(3) Die bisher ermittelte Phaseneinteilung erfolgte unter ausschließlicher Berücksichtigung der Struktur der angegebenen Qualifikationsbedingung. Dabei wurde implizit angenommen, 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 der Aspekt der parallelen Bearbeitung eines einzelnen AEM-Operators näher betrachtet werden. Dazu gehen wir zunächst auf die Vor- und Nachteile dieser Parallelität ein, ehe konkrete Realisierungsformen vorgestellt werden.

6.1 Beurteilung paralleler AEM-Operator-Ausführungen

Da der CPU-Bedarf zur Bearbeitung komplexer Objekte, deutlich höher ist, als der bei der Bearbeitung von Tupeln in relationalen Systemen (vgl. [SPS90]), kann das Zugriffssystem durch parallele Verarbeitungsstrategien mehr Atome produzieren, als sie von einer einzelnen Ausführungseinheit für den AEM-Operator verarbeitet werden können. Da AEM-Operatoren gleichzeitig auch sehr umfangreiche Molekülmengen bearbeiten, sind parallele Verarbeitungsstrategien sinnvoll, um kurze Antwortzeiten des Systems zu garantieren.

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 Obermenge der auszugebenden Molekülmenge, weil auch die Moleküle, die sich gemäß der gegebenen Qualifikationsbedingung nicht qualifizieren, zu einem bestimmten Teil aufgebaut werden müssen. Bei der Untersuchung dieses Problems zeigen sich allerdings neben den immer auftretenden zusätzlichen Verwaltungskosten weitere Abhängigkeiten zu anderen Verarbeitungsparametern, z.B. zu den benutzten Speicherungsstrukturen, die bei der Implementierung berücksichtigt werden müssen.

Bei der unabhängigen Bearbeitung verschiedener Molekülteilmengen in unterschiedlichen Ablaufeinheiten eines AEM-Operators entstehen als Folge kontextfreie Aufträge an das Zugriffssystem. Wenn die benutzten Speicherungsstrukturen nicht an dieses Aufträge angepaßt sind, kann dies, wie bereits in 4.1.2 beschrieben, zu wiederholten Seitenanforderungen im Zugriffssystem führen, wenn verschiedene, unabhängige Leseoperationen unterschiedliche Atome in derselben Seite anfordern.

Weiterer Zusatzaufwand kann durch Überlappungen bei von verschiedenen Molekülen gemeinsam genutzten Teilmolekülen entstehen. Je größer diese Überlappungen sind, umso größer wäre 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, können diese Überlappungen und die Tatsache, ob ein Atom bereits gelesen oder angefordert wurde, relativ einfach erkannt werden. Im Falle der 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

Eine wertabhängige Partitionierung der Ergebnismolekülmenge, bei der jede Partition in einem eigenen Operatoraufruf aufgebaut wird, eignet sich in der Regel nicht. In diesem Fall müßte sich die Partitionierung wiederum an den Speicherungsstrukturen (z.B. Atomclustern) orientieren, um die Lokalität der Seitenzugriffe nicht zu zerstören. Ohne die Atomcluster 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 2.3). Aus diesem Grund wird die folgende Strategie gewählt:

Nicht die konkrete Ergebnismenge sondern der zu bearbeitende Molekültyp wird in verschiedene Teilmolekültypen zerlegt, von denen jeder in einem eigenen Operator realisiert wird. Die einzelnen Ergebnismoleküle werden anschließend durch einen Verbindungsoperator zu den Endergebnissen zusammengesetzt (2. in Abb. 8). Mit dieser Zerlegung können die üblichen, in [Ge95] 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 werden. Im gegebenen Beispiel gilt diese Voraussetzung. Wenn aber auf dem Molekültyp A-B eine selektive Bedingung angegeben wäre, würden auch weniger C-Atom benötigt. In diesem Fall bietet sich das Pipelining zur Ausnutzung von Parallelität an. Die Submoleküle (C-Atome) werden dabei 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 für die bereits hergestellten Wurzelmoleküle die anzuhängenden Submoleküle.

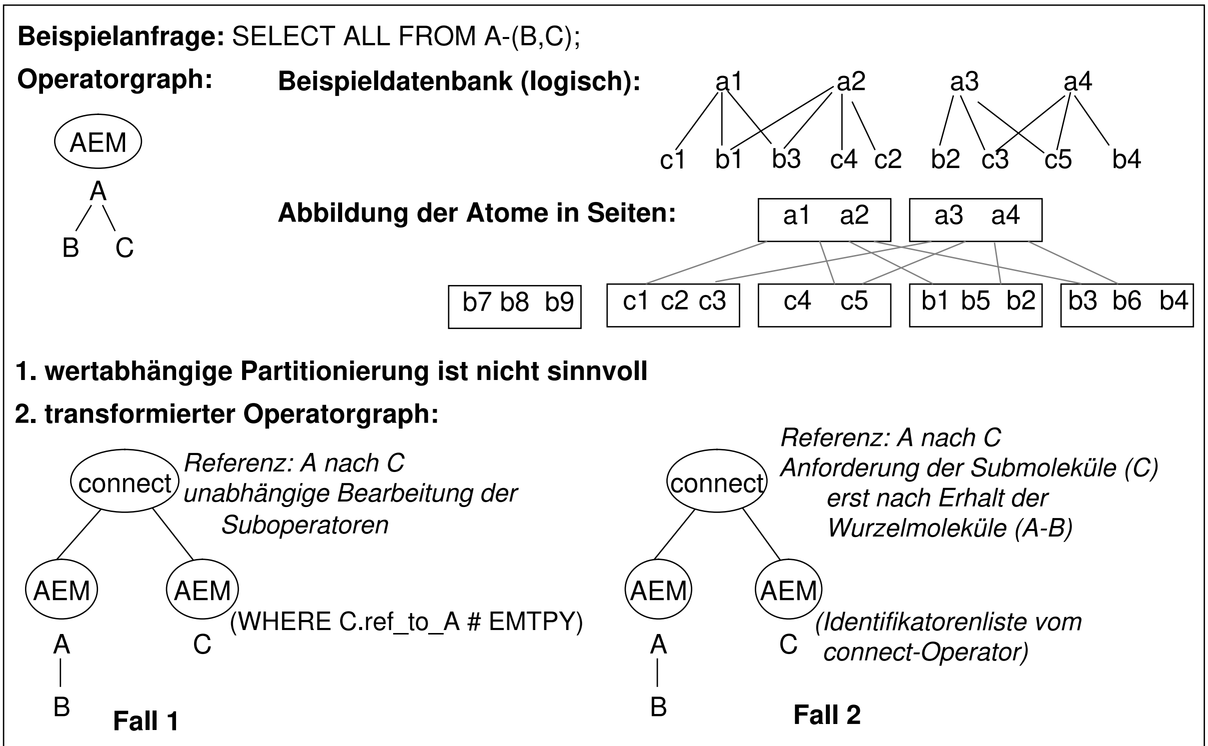


Abb. 8: Parallelitätsformen zur Erzeugung von Molekülen einfacher Molekültypen

Da solche Zerlegungen aber die Molekültypstruktur des ursprünglichen AEM-MT zerstören, muß sichergestellt werden, daß zur Bedingungsauswertung benötigte Atomtypen in den AEM-Operatoren auch zusammenhängend bleiben, da sonst die im Phasenmodell beschriebenen Strategien zur frühzeitigen Disqualifikation nicht mehr anwendbar sind. Eine Auswertung der Bedingung könnte erst nach dem Zusammensetzen der Teilmoleküle im Verbindungsoperator erfolgen. Zusätzlich muß sichergestellt werden, daß in Atomclustern genutzte Molekültypstrukturen bestehen bleiben, damit deren Nutzung 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 Verfahren zur Bestimmung eines geeigneten Abarbeitungsplans.

Zunächst werden die gegebene WHERE-Klausel näher untersucht und eine entsprechende Phaseneinteilung erstellt. Dazu werden zunächst unter Berücksichtigung der verfügbaren konventionellen Zugriffspfade und Atomcluster die für das Bottom-up-Verfahren benötigten Teilmolekültypen, die durchzuführenden Bottom-up-Verfahren selbst sowie die Verknüpfung der einzelnen für den Wurzelatomtyp erhaltenen Identifiermengen spezifiziert. Anschließend wird das Top-down-Verfahren festgelegt. Dies umfaßt die Beschreibung der Reihenfolge, in der einzelne Atomtypen des AEM-MT behandelt werden, der bestehenden Abhängigkeiten sowie die Festlegung einer molekül- oder molekülmengenorientierten Verarbeitung der einzelnen Atomtypen.

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 werden.

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 Modell 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, die vorgestellte Aufgabe.

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 und ohne Ausführung unnötiger Bearbeitungsschritte entschieden werden kann. Ein dem Top-down-Verfahren vorgeschaltetes Bottom-up-Verfahren, in dem die resultierende Ergebnismenge frühzeitig eingeschränkt wird, vermeidet den Aufbau sich nicht qualifizierender Moleküle. 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 Verarbeitungsmodells wurde die Integration von Zugriffspfadstrukturen betrachtet. Es wurde gezeigt, daß neben den konventionellen Zugriffspfaden auch die Atomcluster sowohl in der Top-down- als auch in der Bottom-up-Verfahren einsetzbar sind. Ihre Nutzung hat allerdings Einfluß auf die Phaseneinteilung, so daß die in der Anfrageoptimierung ursprünglich festgelegte Phaseneinteilung bei einer Atomclusternutzung evtl. noch modifiziert werden muß.

Abschließend wurde die parallele Verarbeitung eines AEM-Operators untersucht. Es zeigte sich, daß diese grundsätzlich möglich, aber nicht immer tatsächlich sinnvoll einsetzbar ist. Eine wertbasierte Partitionierung der Ergebnismenge ist normalerweise 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 [Ge95] 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. Für die auch dort notwendigen ad-hoc Anfragesprachen, z.B. [De90, Ca94] gilt, daß sie eine Ausdrucksmächtigkeit haben, die die vom AEM-Operator realisierte umfaßt. Hier sei 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 dort somit, bei einer entsprechenden Umsetzung der OQL-Anfrage in einen dem AEM-Operator ähnlichen Algebra-Operator vorausgesetzt, in ähnlicher Form angewandt werden. Ausnahmen stellen das Bottom-up-Verfahren, das auf der Symmetrie der Referenzen beruht, und die konkrete Nutzung von Zugriffspfaden, die systemabhängig überprüft werden muß, dar.

In weiterführenden Arbeiten gilt es nun, zu dem hier vorgestellten 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, 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

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

Ge95 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. BTW 1991, IFB 270, Springer-Verlag, pp. 178-197, 1991

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

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

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

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

KD91 Keßler, U., Dadam, P.: Auswertung komplexer Anfragen an hierarchisch strukturierte Objekte mit Pfadindexen, Proc. BTW 1991, IFB 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 16th VLDB Conference, pp. 290-301, 1990

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

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

Sch91 Schöning, H.: Praktische Behandlung von Nullwerten - Realisierung im Molekül-Atom-Datenmodell, Proc. BTW 1991, IFB 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 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

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