Acta Informatica 4, 299—310 (1975)

Grundlagen einer Theorie der Datenstrukturen und Zugriffssysteme

Teil II: Zugriffssysteme

H.-D. Ehrich

Eingegangen am 8. Februar 1974

Summary. In part I of this paper, a mathematical model for data structures has been presented. Based on this model and on a corresponding formalized notion of retrieval operation, a model for storage access systems is introduced, which is called a Q-system. Equivalence, separation, and decomposition of Q-systems are studied, resulting in a set of basic principles for transforming Q-systems. These can be used for a systematical description of practical storage access systems for multi-attribute-retrieval.

1. Einleitung

Auf der Grundlage des im Teil I dieser Arbeit [6] eingeführten mathematischen Modells für Datenstrukturen wird in dieser Arbeit ein kombiniertes System (Datenstruktur-Anfrageoperationen), das Q-System, definiert und untersucht.

Die Einführung eines plausiblen Äquivalenzbegriffs gibt Anlaß zu drei grundlegenden Sätzen über äquivalente Umformungen von Q-Systemen. Insbesondere wird in Ergänzung früherer Ergebnisse [5] gezeigt, daß es zu jedem Q-System eine äquivalente Tabelle gibt. Die dann folgenden Abschnitte beschäftigen sich mit Aufteilungen und Zerlegungen von Q-Systemen. Die vor allem interessierende Frage ist, wann bei der Aufteilung eines Q-Systems in Teilsysteme Informationen verlorengehen. Im Abschnitt 4 wird eine notwendige und hinreichende Bedingung dafür angegeben, daß dies nicht geschieht. Zerlegungen sind spezielle (besonders „effektive") Aufteilungen, bei denen die komplette Antwort auf eine Anfrage bereits in einem der Teilsysteme gegeben wird. Das Hauptergebnis im Abschnitt 5 ist eine notwendige und hinreichende Bedingung dafür, daß eine Aufteilung eine Zerlegung ist. Q-Systeme sind Modelle für praktische Speicher- und Zugriffssysteme, wie sie zur Implementierung von Datenzugriffen im Speicher eines Rechners benutzt werden. Wie deren Konstruktionsprinzipien mit Hilfe der theoretischen Ergebnisse dieses Kapitels erklärt werden können, wird im Abschnitt 6 erläutert.

2. Q-Systeme

Der Zweck dieser Untersuchungen ist es, einen konzeptionellen Rahmen zu finden für die Untersuchung praktischer Speicher- und Zugriffssysteme, welche zur Implementierung von Datenstrukturen zur Auswahl stehen. Der hier formalisierte Begriff der Frage orientiert sich daher nicht an den Bedürfnissen leistungsfähiger Benutzer-Anfrage-Sprachen (s. etwa [10]). Komplexere Benutzeranfragen werden in der Regel in eine Reihe von Einzelzugriffen aufgelöst, und es sind diese Einzelzugriffe, welche hier untersucht werden sollen. Der größte Teil der Ergebnisse in diesem und im nächsten Abschnitt ist in [5] dargestellt. Wir referieren nur kurz das Wichtigste.

Sei $D=(\Omega, A, V, \varrho)$ eine Datenstruktur.

Definition 2.1. Eine *Frage* ist eine zweistellige Relation $q \subset A \times V$. Die *Antwort* auf q in D ist

$$\alpha_D(q) := \{\omega \in \Omega \mid q \subset R_\omega\}.$$

Die Länge von q ist $|q|$, und Fragen der Länge 1 heißen *Elementarfragen*.

Fragen sind Kombinationen von Eigenschaften, und die Antwort besteht aus allen Objekten, welche alle geforderten Eigenschaften haben. Für die Vereinigung und den Durchschnitt von Fragen gelten die Gesetze

$$1.\quad \alpha_D(q \cap q') \subset \alpha_D(q) \cup \alpha_D(q'),$$

$$2.\quad \alpha_D(q \cup q') = \alpha_D(q) \cap \alpha_D(q').$$

Daher lassen sich logische Konjunktionen von Fragen durch Vereinigung der Fragen darstellen. (Ähnliche Beziehungen wurden von Turski [13] hergeleitet.)

Wir definieren jetzt ein kombiniertes System, bestehend aus einer Datenstruktur und einer zugehörigen Menge von *Standardfragen*. Dieser Begriffsbildung liegt die Beobachtung zugrunde, daß praktische Zugriffssysteme zur Implementierung von Retrieval-Prozessen immer für eine bestimmte, vorgegebene Klasse von Suchargumenten konzipiert sind. Solche „Standardfragen" werden dann effizient beantwortet, während andere Fragen nach prinzipiell auch vorhandener Information sehr ineffizient sein können. (Ein Beispiel dafür ist das Telefonbuch, mit dessen Hilfe ein Suchprozeß Name$\rightarrow$Telefonnummer effizient durchführbar, ein Suchprozeß Telefonnummer$\rightarrow$Name jedoch sehr zeitaufwendig ist.)

Definition 2.2. Ein *Q-System* ist ein Paar $\mathfrak{Q}=(D, Q)$, wobei $D=(\Omega, A, V, \varrho)$ eine Datenstruktur und $Q \subset \mathfrak{P}(A \times V)$ eine Menge von Standardfragen ist.

Für die Implementierung von Q-Systemen sind die folgenden speziellen Eigenschaften von Bedeutung:

Definition 2.3. Ein Q-System $\mathfrak{Q}=(D, Q)$ heißt

$$\begin{array}{lll} 1. & \textit{Tabelle} & :\Leftrightarrow \forall q, q' \in Q\colon \mathrm{pr}_1(q) = \mathrm{pr}_1(q'), \\ 2. & \textit{elementar} & :\Leftrightarrow \forall q \in Q\colon |q| = 1, \\ 3. & \textit{boolesch} & :\Leftrightarrow V \subset \{0, 1\} \wedge \mathrm{pr}_2(Q) \subset \{1\}. \end{array}$$

Hierbei ist $\mathrm{pr}_2(Q) := \bigcup_{q \in Q} \mathrm{pr}_2(q)$.

3. Äquivalenz

Für die Behandlung der Frage, ob und gegebenenfalls wie sich Q-Systeme umformen lassen, um sie unter Umständen in eine für die Implementierung günstigere Form bringen zu können, benötigen wir den Begriff der Äquivalenz von Q-Systemen.

Definition 3.1. Zwei Q-Systeme $\mathfrak{Q}_i=(D_i, Q_i)$, $i=1, 2$, heißen *äquivalent*, in Zeichen $\mathfrak{Q}_1 \sim \mathfrak{Q}_2$, genau dann, wenn folgende Bedingungen erfüllt sind ($\alpha_i := \alpha_{D_i}$, $i=1, 2$):

$$1.\quad \alpha_1(Q_1) \cup \alpha_2(Q_2) \subset \Omega_1 \cap \Omega_2,$$

$$2.\quad \exists \varkappa\colon Q_1 \twoheadrightarrow Q_2\ \forall q_1 \in Q_1\colon \alpha_1(q_1) = \alpha_2(\varkappa(q_1)).$$

Hierbei ist $\alpha(Q) := \bigcup_{q \in Q} \alpha(q)$.

Grob gesagt sind Q-Systeme äquivalent, wenn auf gleiche Fragen gleiche Antworten gegeben werden. Auf der Grundlage dieses Äquivalenzbegriffes erhalten wir folgende Ergebnisse:

Satz 3.1. Zu jeder Tabelle $\mathfrak{T}=(D,Q)$ gibt es eine äquivalente Tabelle $\mathfrak{T}^*=(D^*,Q^*)$, deren Attributmenge höchstens ein Attribut enthält (welche also insbesondere elementar ist).

In dem Beweis für diesen Satz (s. [5]) wird die Tabelle $\mathfrak{T}^*=(D^*,Q^*)$, $D^*=(\Omega^*,A^*,V^*,\varrho^*)$, für den Fall, daß Q mindestens eine nichtleere Frage enthält, folgendermaßen konstruiert: sei $A=\{a_1,\ldots,a_n\}$, und o.B.d.A. sei $\mathrm{pr}_1(q)=\{a_1,\ldots,a_p\}$, $1\leqq p\leqq n$, für alle $q\in Q$. Dann setzen wir $a^*:=(a_1,\ldots,a_p)$ und ordnen jeder Frage $q\in Q$, $q=\{(a_1,v_1),\ldots,(a_p,v_p)\}$, das Paar $\varkappa(q)=\{(a^*,v^*)\}$ zu, wobei $v^*:=(v_1,\ldots,v_p)\in V^p$ ist. Dann setzen wir

$$\begin{aligned}
&Q^*:=\{\varkappa(q)\mid q\in Q\},\\
&\varrho^*:=\{(\omega,a^*,v^*)\mid \forall i=1,\ldots,p\colon (\omega,a_i,v_i)\in\varrho\},\\
&V^*:=\mathrm{pr}_3(\varrho^*),\\
&A^*:=\{a^*\},\\
&\Omega^*:=\mathrm{pr}_1(\varrho^*).
\end{aligned}$$

Satz 3.2. Zu jedem Q-System $\mathfrak{Q}=(D,Q)$ gibt es ein äquivalentes Q-System $\mathfrak{Q}^*=(D^*,Q^*)$, welches elementar und boolesch ist, und dessen Datenstruktur D^* einfach ist.

Um die Konstruktion von $\mathfrak{Q}^*$ zu beschreiben, sei $\alpha:=\alpha_D$. Wir setzen

$$\begin{aligned}
&\Omega^*:=\alpha(Q),\\
&A^*:=Q,\\
&V^*:=\mathrm{pr}_3(\varrho^*),\\
&\varrho^*:=\{(\omega,q,1)\mid q\in Q,\ \omega\in\alpha(q)\}\\
&\qquad\cup\{(\omega,q,0)\mid q\in Q,\ \omega\in\Omega^*-\alpha(q)\},\\
&D^*:=(\Omega^*,A^*,V^*,\varrho^*),\\
&Q^*:=\{\{(q,1)\}\mid q\in Q\}.
\end{aligned}$$

Mit $\varkappa(q):=\{(q,1)\}$ ist dann $\mathfrak{Q}^*:=(D^*,Q^*)$ zu $\mathfrak{Q}=(D,Q)$ äquivalent [5].

Darüberhinaus gilt der folgende für die Anwendung sehr wichtige Satz.

Satz 3.3. Zu jedem Q-System $\mathfrak{Q}=(D,Q)$ gibt es eine Tabelle $\mathfrak{T}^*=(D^*,Q^*)$ mit $\mathfrak{Q}\sim\mathfrak{T}^*$.

Beweis. Wir setzen

$$\begin{aligned}
&\Omega^*:=\alpha(Q),\\
&A^*:=\{\bar\alpha\},\\
&V^*:=Q,\\
&\varrho^*:=\{(\omega,\bar\alpha,q)\mid \omega\in\alpha(q)\},\\
&D^*:=(\Omega^*,A^*,V^*,\varrho^*),\\
&Q^*:=\{\{(\bar\alpha,q)\}\mid q\in Q\}.
\end{aligned}$$

Dann ist $\mathfrak{T}^* := (D^*, Q^*) \sim \mathfrak{Q} = (D, Q)$, wenn wir für alle $q \in Q$ setzen: $\varkappa(q) := (\bar{\alpha}, q)$. Denn es ist $\alpha(Q) = \alpha^*(Q^*) = \Omega^* \subset \Omega$ (mit $\alpha := \alpha_D$ und $\alpha^* := \alpha_{D^*}$). Ferner gilt: $\omega \in \alpha(q) \Leftrightarrow (\omega, \bar{\alpha}, q) \in \varrho^* \Leftrightarrow \omega \in \alpha^*(\{(\bar{\alpha}, q)\}) \Leftrightarrow \omega \in \alpha^*(\varkappa(q))$. Wir bemerken darüber hinaus, daß $\mathfrak{T}^*$ nach Konstruktion eine elementare Tabelle ist.

4. Aufteilungen

In der Praxis können größere Datenstrukturen meist nicht zusammenhängend abgespeichert werden. Sie müssen in kleinere Teilstrukturen aufgeteilt werden, sowohl aus technischen Gründen wie auch aus Gründen der Effizienz. Wir wenden uns nun der Frage zu, wie man solche Aufteilungen vornehmen kann, ohne daß Informationen verlorengehen. Im Anschluß daran werden besonders „effiziente" Aufteilungen, die Zerlegungen, erörtert.

Seien $D_i = (\Omega_i, A_i, V_i, \varrho_i)$, $i = 1, 2$, zwei Datenstrukturen. Ihre *Vereinigung*

$$D = D_1 \cup D_2 = (\Omega, A, V, \varrho)$$

ist definiert durch $\Omega := \Omega_1 \cup \Omega_2$, $A := A_1 \cup A_2$, $V := V_1 \cup V_2$ und $\varrho := \varrho_1 \cup \varrho_2$.

Die Vereinigung zweier Q-Systeme $\mathfrak{Q}_i = (D_i, Q_j)$, $i = 1, 2$, ist

$$\mathfrak{Q} = \mathfrak{Q}_1 \cup \mathfrak{Q}_2 = (D, Q),$$

wobei $D := D_1 \cup D_2$ und $Q := Q_1 \cup Q_2$ ist.

Die Antwortfunktion in $\mathfrak{Q}$, $\mathfrak{Q}_1$ und $\mathfrak{Q}_2$ sei α, α_1 bzw. α_2, und die Sätze eines Objekts $\omega \in \Omega$ in D, D_1 und D_2 seien R_ω, R_ω^1 bzw. R_ω^2, wobei $\alpha_i(q) := \emptyset$ ist für $q \notin Q_i$ und $R_\omega^i := \emptyset$ für $\omega \notin \Omega_i$, $i = 1, 2$.

Lemma 4.1. Ist $D = D_1 \cup D_2$ für Datenstrukturen D, D_1 und D_2, so ist $R_\omega = R_\omega^1 \cup R_\omega^2$ für alle $\omega \in \Omega$.

Beweis. $R_\omega = \{(a, v) \mid a \in A_1 \cup A_2 \wedge v \in V_1 \cup V_2 \wedge (\omega, a, v) \in \varrho_1 \cup \varrho_2\}$

$$= R_\omega^1 \cup R_\omega^2 \cup S_\omega,$$

wobei gilt:

$$R_\omega^i = \{(a, v) \mid a \in A_i \wedge v \in V_i \wedge (\omega, a, v) \in \varrho_i\}, \qquad i = 1, 2,$$

und

$$S_\omega = \{(a, v) \mid a \in A_i \wedge v \in V_j \wedge (\omega, a, v) \in \varrho_k \wedge \{i, j, k\} = \{1, 2\}\}.$$

Aufgrund der Definition einer Datenstruktur gilt für $k = 1, 2$: $(\omega, a, v) \in \varrho_k \Rightarrow \omega \in \Omega_k \wedge a \in A_k \wedge v \in V_k$. Daher ist $S_\omega \subset R_\omega^1 \cup R_\omega^2$.

Lemma 4.2. Seien $\mathfrak{Q}$, $\mathfrak{Q}_1$ und $\mathfrak{Q}_2$ Q-Systeme mit $\mathfrak{Q} = \mathfrak{Q}_1 \cup \mathfrak{Q}_2$. Dann gilt für alle $q \in Q$

$$\alpha(q) = \alpha_1(q) \cup \alpha_2(q) \cup S(q),$$

wobei gilt:

$$S(q) = \{\omega \in \Omega_1 \cup \Omega_2 \mid \exists q' \subset A_1 \times V_1 \exists q'' \subset A_2 \times V_2:$$
$$q = q' \cup q'' \wedge q' \cap q'' = \emptyset \wedge q' \neq \emptyset \wedge q'' \neq \emptyset \wedge \omega \in \alpha_1(q') \cap \alpha_2(q'')\}.$$

Beweis. Nach dem vorherigen Lemma gilt:

$$\begin{aligned}
\alpha(q) &= \{\omega \in \Omega_1 \cup \Omega_2 \mid q \subset R^1_\omega \cup R^2_\omega\} \\
&= \{\omega \in \Omega_1 \cup \Omega_2 \mid q \subset R^1_\omega\} \cup \{\omega \in \Omega_1 \cup \Omega_2 \mid q \subset R^2_\omega\} \\
&\quad \cup \{\omega \in \Omega_1 \cup \Omega_2 \mid \exists q' \subset A_1 \times V_1 \exists q'' \subset A_2 \times V_2 : \\
&\quad q = q' \cup q'' \wedge q' \cap q'' = \emptyset \wedge q' \neq \emptyset \wedge q'' \neq \emptyset \wedge q' \subset R^1_\omega \wedge q'' \subset R^2_\omega\} \\
&= \alpha_1(q) \cup \alpha_2(q) \cup S(q).
\end{aligned}$$

Korollar 4.1. Unter den Voraussetzungen des Lemmas ist

$$\alpha(q) \supset \alpha_1(q) \cup \alpha_2(q),$$

und für Elementarfragen gilt immer die Gleichheit.

Daß die Vereinigung der Antworten auf eine Frage in den beiden Teilsystemen echt enthalten sein kann in der Antwort auf die gleiche Frage im Gesamtsystem, wird durch das folgende einfache Beispiel demonstriert:

$$\begin{array}{ccc}
& D\colon (\omega, a_1, v_1) & \qquad q = \{(a_1 v_1), (a_2, v_2)\} \\
& (\omega, a_2, v_2) & \\
\swarrow & & \searrow \\
D_1\colon (\omega, a_1, v_1) & & D_2\colon (\omega, a_2, v_2)
\end{array}$$

Es ist $\omega \in \alpha(q)$, aber es gilt sowohl $\omega \notin \alpha_1(q)$ als auch $\omega \notin \alpha_2(q)$.

Der Fall, daß für alle Fragen $q \in Q$ die Gleichheit $\alpha(q) = \alpha_1(q) \cup \alpha_2(q)$ besteht, ist von besonderem Interesse. Wir führen dafür eine besondere Bezeichnung ein.

Definition 4.1. Seien $\mathfrak{Q}$, $\mathfrak{Q}_1$ und $\mathfrak{Q}_2$ Q-Systeme. Das Paar $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ heißt *Aufteilung* von $\mathfrak{Q}$ genau dann, wenn die folgenden Bedingungen erfüllt sind:

1. $\mathfrak{Q} = \mathfrak{Q}_1 \cup \mathfrak{Q}_2$,
2. $\forall q \in Q : \alpha(q) = \alpha_1(q) \cup \alpha_2(q)$.

Offenbar ist die zweite Bedingung genau dann erfüllt, wenn für alle $q \in Q$ und alle $\omega \in \Omega$ gilt:

$$q \subset R^1_\omega \cup R^2_\omega \Rightarrow q \subset R^1_\omega \vee q \subset R^2_\omega.$$

Dies folgt aus Lemma 4.1 und der Definition der Antwortfunktion α. Der nächste Satz gibt eine von der Fragemenge Q unabhängige hinreichende Bedingung dafür, daß $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eine Aufteilung ist. Enthält Q alle Fragen der Länge 2, so ist diese Bedingung auch notwendig.

Satz 4.1. Sei $\mathfrak{Q} = \mathfrak{Q}_1 \cup \mathfrak{Q}_2$ für drei Q-Systeme $\mathfrak{Q}$, $\mathfrak{Q}_1$ und $\mathfrak{Q}_2$. Gilt für alle $\omega \in \Omega$

$$R^1_\omega \subset R^2_\omega \vee R^2_\omega \subset R^1_\omega,$$

so ist das Paar $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eine Aufteilung von $\mathfrak{Q}$.

Für den Fall, daß Q alle Fragen der Länge 2 enthält, gilt auch die Umkehrung.

Beweis. Sei $q \in Q$. Ist $\alpha(q) = \emptyset$, so ist trivialerweise $\alpha(q) \subset \alpha_1(q) \cup \alpha_2(q)$, also gilt wegen Korollar 4.1 die Gleichheit. Sei nun $\alpha(q) \neq \emptyset$, etwa $\omega \in \alpha(q)$. Wegen Lemma

4.1 ist dann $q \subset R^1_\omega \cup R^2_\omega$. Da nach Voraussetzung $R^1_\omega \subset R^2_\omega$ oder $R^2_\omega \subset R^1_\omega$ ist, ist $q \subset R^1_\omega$ oder $q \subset R^2_\omega$, also $\omega \in \alpha_1(q) \cup \alpha_2(q)$. Mit Korollar 4.1 folgt daraus $\alpha(q) = \alpha_1(q) \cup \alpha_2(q)$.

Um die Umkehrung zu zeigen, setzen wir voraus, daß Q alle Fragen der Länge 2 enthält. Wir nehmen an, daß für ein $\omega \in \Omega_1 \cup \Omega_2$ sowohl $R^1_\omega \not\subset R^2_\omega$ als auch $R^2_\omega \not\subset R^1_\omega$ ist. Dann gibt es Paare (a, v), $(a', v') \in A \times V$ mit

$$(a, v) \in R^1_\omega, \qquad (a, v) \notin R^2_\omega,$$
$$(a', v') \notin R^1_\omega, \qquad (a', v') \in R^2_\omega,$$

und $q = \{(a, v), (a', v')\}$ ist eine Frage aus Q, für die $\omega \notin \alpha_i(q)$ ist, $i = 1, 2$, aber für die $\omega \in \alpha(q)$ ist. Also ist dann $\alpha(q) \neq \alpha_1(q) \cup \alpha_2(q)$.

5. Zerlegungen

Hat man eine Aufteilung $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eines Q-Systems $\mathfrak{Q}$, so kann man die Beantwortung einer Frage in $\mathfrak{Q}$ zurückführen auf die Beantwortung der Frage in den Teilstrukturen $\mathfrak{Q}_1$ und $\mathfrak{Q}_2$, ohne daß dabei Information verlorengeht. Und umgekehrt muß man bei einer solchen Vorgehensweise der Realisierung eines Q-Systems durch Aufspalten in Teilstrukturen darauf achten, daß man Aufteilungen bekommt, will man nicht Informationen verlieren.

Besonders effiziente Realisierungen (bzgl. der Zugriffszeit) durch Aufteilung erhält man dann, wenn man jede Frage aus Q bereits in einem dieser Frage zugeordneten Teilsystem vollständig beantwortet bekommt.

Definition 5.1. Seien $\mathfrak{Q}$, $\mathfrak{Q}_1$ und $\mathfrak{Q}_2$ Q-Systeme. Das Paar $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ heißt *Zerlegung* von $\mathfrak{Q}$ genau dann, wenn die folgenden Bedingungen erfüllt sind:

1. $\mathfrak{Q} = \mathfrak{Q}_1 \cup \mathfrak{Q}_2$,
2. $\forall q \in Q: \alpha(q) = \alpha_1(q) \vee \alpha(q) = \alpha_2(q)$.

Offenbar ist jede Zerlegung von $\mathfrak{Q}$ eine Aufteilung von $\mathfrak{Q}$. Ist $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eine Zerlegung von $\mathfrak{Q}$, so kann man durch geeignete Umformung der Fragmengen Q_1 und Q_2 immer erreichen, daß jede Antwort in einem Teilsystem bereits die vollständige Antwort des ganzen Systems ist, daß also für alle $q \in Q_i$ gilt:

$$(*) \quad \alpha_i(q) = \alpha(q), \qquad i = 1, 2.$$

Um dies einzusehen, sei $\mathfrak{Q} = (D, Q)$ ein Q-System, und seien D_1, D_2 Datenstrukturen, so daß $D = D_1 \cup D_2$ ist. Für $\mathfrak{Q}_i = (D_i, Q)$ sei das Paar $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eine Zerlegung von $\mathfrak{Q}$. Wir setzen

$$Q'_i := \{q \in Q \mid \alpha_i(q) = \alpha(q)\}, \qquad i = 1, 2.$$

Aus der zweiten Bedingung der Definition 5.1 folgt sofort, daß $Q = Q'_1 \cup Q'_2$ ist, so daß also für $\mathfrak{Q}'_i = (D_i, Q'_i)$ gilt: $\mathfrak{Q} = \mathfrak{Q}'_1 \cup \mathfrak{Q}'_2$. Daß $(\mathfrak{Q}'_1, \mathfrak{Q}'_2)$ wieder eine Zerlegung ist, ist klar, und diese Zerlegung hat die Eigenschaft $(*)$.

Sind Q_1 und Q_2 Teilmengen der Fragemenge Q eines Q-Systems $\mathfrak{Q} = (D, Q)$ mit der Eigenschaft $Q_1 \cup Q_2 = Q$, so läßt sich leicht eine Zerlegung $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ von $\mathfrak{Q}$ finden, so daß $\mathfrak{Q}_i = (D_i, Q_i)$ ist für geeignete Datenstrukturen D_i, $i = 1, 2$. Trivialerweise gilt dies für $\mathfrak{Q}_i := (D, Q_i)$, und wir können weitere solche Zerlegungen etwa durch Weglassen überflüssiger Objekte, Attribute, Werte und Tripel in jedem der Teilsysteme bekommen.

Interessanter wird das Problem, wenn man von einer Aufteilung der Datenstruktur $D=D_1\cup D_2$ ausgeht und danach fragt, ob dadurch eine Zerlegung des Q-Systems $\mathfrak{Q}=(D, Q)$ impliziert wird. In Anbetracht der obigen Bemerkungen genügt es zu untersuchen, ob für $\mathfrak{Q}_i=(D_i, Q)$, $i=1, 2$, das Paar $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eine Zerlegung von $\mathfrak{Q}$ ist. Wir können dann nachträglich die Bedingung $(*)$ durch Einschränkung von Q für jedes Teilsystem $\mathfrak{Q}_i$ erfüllen, falls wir es wünschen.

Der folgende Zerlegungssatz gibt eine notwendige und hinreichende Bedingung dafür, daß eine Aufteilung eine Zerlegung ist.

Satz 5.1. Sei $\mathfrak{Q}=(D, Q)$ ein Q-System, und seien $D_i=(\Omega_i, A_i, V_i, \varrho_i)$, $i=1, 2$, Datenstrukturen, so daß $D_1\cup D_2=D=(\Omega, A, V, \varrho)$ ist. Für $\mathfrak{Q}_i=(D_i, Q)$, $i=1, 2$, sei $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eine Aufteilung von $\mathfrak{Q}$.

Dann ist $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ genau dann eine Zerlegung von $\mathfrak{Q}$, wenn für alle $q\in Q$ und alle $\omega, \omega'\in\Omega$ gilt:

$$q\subset R^1_\omega\cap R^2_{\omega'} \Rightarrow q\subset R^1_{\omega'} \vee q\subset R^2_\omega.$$

Beweis. Wir müssen zeigen, daß $\alpha(q)=\alpha_1(q)$ oder $\alpha(q)=\alpha_2(q)$ ist für alle $q\in Q$. Dazu formen wir die Bedingung des Satzes zunächst um:

$$\begin{aligned}
&(1)\ \forall q\in Q\,\forall\omega,\omega'\in\Omega\colon (q\subset R^1_\omega\cap R^2_{\omega'} \Rightarrow q\subset R^1_{\omega'} \vee q\subset R^2_\omega)\\
\Leftrightarrow\ &\ \forall q\in Q\,\forall\omega,\omega'\in\Omega\colon (q\not\subset R^1_\omega \vee q\not\subset R^2_{\omega'} \vee q\subset R^1_{\omega'} \vee q\subset R^2_\omega)\\
\Leftrightarrow\ &(2)\ \forall q\in Q\,\forall\omega,\omega'\in\Omega\colon (q\subset R^1_\omega \Rightarrow q\subset R^2_\omega) \vee (q\subset R^2_{\omega'} \Rightarrow q\subset R^1_{\omega'})\\
\Leftrightarrow\ &\ \forall q\in Q\colon (\forall\omega\in\Omega\colon q\subset R^1_\omega \Rightarrow q\subset R^2_\omega) \vee (\forall\omega'\in\Omega\colon q\subset R^2_{\omega'} \Rightarrow q\subset R^1_{\omega'})\\
\Leftrightarrow\ &(3)\ \forall q\in Q\colon (\exists\omega\in\Omega\colon q\subset R^1_\omega \wedge q\not\subset R^2_\omega) \Rightarrow (\forall\omega'\in\Omega\colon q\subset R^2_{\omega'} \Rightarrow q\subset R^1_{\omega'}).
\end{aligned}$$

Sei nun $q\in Q$. Gilt für alle $\omega\in\Omega$ die Äquivalenz

$$q\subset R^1_\omega \Leftrightarrow q\subset R^2_\omega,$$

so ist $\alpha_1(q)=\alpha_2(q)=\alpha(q)$, da $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eine Aufteilung ist. Gilt diese Äquivalenz für q nicht, so können wir z.B. annehmen, daß es ein $\omega\in\Omega$ gibt mit

$$q\subset R^1_\omega \wedge q\not\subset R^2_\omega.$$

Daraus folgt aber nach (3), daß für alle $\omega'\in\Omega$ gilt:

$$q\subset R^2_{\omega'} \Rightarrow q\subset R^1_{\omega'},$$

und das heißt, daß $\alpha_2(q)\subset\alpha_1(q)$ ist. Also ist $\alpha_1(q)=\alpha_1(q)\cup\alpha_2(q)=\alpha(q)$, da $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eine Aufteilung ist. Da die Aussage (1) bzw. (2) invariant gegenüber Vertauschung der Indizes 1 und 2 ist, können wir aus der Annahme

$$q\subset R^2_\omega \wedge q\not\subset R^1_\omega$$

folgern: $\alpha_2(q)=\alpha(q)$. Die Bedingung (1) ist also hinreichend dafür, daß $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eine Zerlegung ist.

Die Notwendigkeit dieser Bedingung zeigen wir indirekt, indem wir annehmen, sie sei nicht erfüllt. Dann gibt es nach (2) eine Frage $q\in Q$ und Objekte $\omega, \omega'\in\Omega$, so daß gilt:

$$q \subset R^1_\omega \wedge q \not\subset R^2_\omega \wedge q \not\subset R^1_{\omega'} \wedge q \subset R^2_{\omega'}.$$

In diesem Falle ist $\omega \notin \alpha_2(q)$ und $\omega' \notin \alpha_1(q)$, aber beide Objekte sind in $\alpha(q)$ enthalten. Es muß also $\alpha_i(q) \neq \alpha(q)$ sein für $i=1, 2$, d.h. die Aufteilung $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ kann keine Zerlegung sein.

Für die Aussage des Satzes gibt es eine einfachere, von Q unabhängige hinreichende Bedingung, die im Falle $Q = \mathfrak{P}(A \times V)$ auch notwendig ist.

Korollar 5.1. Unter den Voraussetzungen des Satzes 5.1 ist die Bedingung

$$\forall \omega, \omega' \in \Omega\colon R^1_\omega \cap R^2_{\omega'} \subset R^1_{\omega'} \vee R^1_\omega \cap R^2_{\omega'} \subset R^2_\omega$$

hinreichend dafür, daß $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ eine Zerlegung ist. Ist speziell $Q = \mathfrak{P}(A \times V)$, so ist diese Bedingung auch notwendig.

Die Begriffe der Aufteilung und der Zerlegung lassen sich ohne Schwierigkeiten auf mehr als zwei Teilsysteme verallgemeinern. Aus den Definitionen 4.1 und 5.1 folgt unmittelbar, daß $(\mathfrak{Q}_1, \mathfrak{Q}_2)$ und $(\mathfrak{Q}_1 \cup \mathfrak{Q}_2, \mathfrak{Q}_3)$ genau dann beides Aufteilungen (Zerlegungen) sind, wenn $(\mathfrak{Q}_1, \mathfrak{Q}_2 \cup \mathfrak{Q}_3)$ und $(\mathfrak{Q}_2, \mathfrak{Q}_3)$ beides Aufteilungen (Zerlegungen) sind. Man kann in diesem Fall daher von einer Aufteilung (Zerlegung) in drei Teilsysteme $(\mathfrak{Q}_1, \mathfrak{Q}_2, \mathfrak{Q}_3)$ sprechen.

Im allgemeinen Falle ist ein n-Tupel von Q-Systemen $(\mathfrak{Q}_1, \ldots, \mathfrak{Q}_n)$ genau dann eine Aufteilung (Zerlegung) eines Q-Systems $\mathfrak{Q}$, wenn für alle $i=1, \ldots, n-1$ die Paare

$$\left(\mathfrak{Q}_i, \bigcup_{j=i+1}^{n} \mathfrak{Q}_j\right)$$

Aufteilungen (Zerlegungen) von

$$\bigcup_{j=i}^{n} \mathfrak{Q}_j$$

sind, und das läßt sich durch Satz 4.1 (bzw. 5.1) nachprüfen.

6. Anwendungen

In diesem Kapitel wollen wir zeigen, wie die theoretischen Ergebnisse dieser Arbeit für die Konstruktion praktischer Speicher- und Zugriffssysteme angewendet werden können. Wir gehen dabei von einem beliebigen Q-System $\mathfrak{Q} = (D, Q)$ aus und beabsichtigen, es durch Aufteilung bzw. Zerlegung und durch äquivalente Umformungen in eine Form zu bringen, die eine direkte Implementierung durch geläufige Techniken für elementare Tabellen erlaubt.

Es gibt zwei grundsätzliche Möglichkeiten, mit dem gegebenen Q-System $\mathfrak{Q}$ zu verfahren: zum ersten können wir $\mathfrak{Q}$ nach Satz 3.3 direkt in eine äquivalente Tabelle umformen und diese implementieren; zum zweiten können wir $\mathfrak{Q}$ in eine Menge von Tabellen zerlegen, deren Vereinigung zu $\mathfrak{Q}$ äquivalent ist (Satz 5.1).

Betrachten wir zunächst die letztere Möglichkeit. Eine naheliegende Verfahrensweise ist es, folgende Äquivalenzrelation $\equiv$ auf der Menge der Standardfragen Q einzuführen:

$$q_1 \equiv q_2 :\Leftrightarrow \mathrm{pr}_1(q_1) = \mathrm{pr}_1(q_2) \qquad \text{für alle } q_1, q_2 \in Q.$$

Die zugehörigen Äquivalenzklassen seien $Q_1', Q_2', \ldots, Q_n'$. Dann sind die Q-Systeme $\mathfrak{Q}_i' = (D, Q_i')$ Tabellen, welche nach Satz 3.1 in äquivalente elementare Tabellen $\mathfrak{Q}_i = (D_i, Q)$ umgeformt werden können. Mit den Kriterien des Satzes 5.1 können wir mit der Festlegung der D_i darauf achten, daß $(\mathfrak{Q}_1, \ldots, \mathfrak{Q}_n)$ eine Zerlegung von $\mathfrak{Q}$ wird. Haben wir dies erreicht, so finden wir die Antwort $\alpha(q)$ auf eine Frage $q \in Q$, indem wir mit Hilfe von $\mathrm{pr}_1(q)$ die Äquivalenzklasse Q_i', $1 \leqq i \leqq n$, bestimmen, dann in die Tabelle $\mathfrak{Q}_i$ hineingehen und dort die Antwort $\alpha_i(q)$ ermitteln. Da $(\mathfrak{Q}_1, \ldots, \mathfrak{Q}_n)$ eine Zerlegung ist, ist $\alpha_i(q) = \alpha(q)$.

Um konkreter zu werden, sei $\mathfrak{Q}$ ein elementares Q-System. Zwei Standardfragen $q_1 = \{(a_1, v_1)\}$ und $q_2 = \{(a_2, v_2)\}$ sind genau dann äquivalent, wenn $a_1 = a_2$ ist. Die Tabellen $\mathfrak{Q}_i' = (D, Q_i')$ sind dann bereits elementar, und wir kommen zu den äquivalenten Tabellen $\mathfrak{Q}_i = (D_i, Q_i)$, indem wir $Q_i := Q_i'$ setzen und aus D jeweils alle überflüssigen Objekte, Attribute und Werte herauslassen. Alle (a, v)-Paare in Sätzen von Objekten einer Tabelle $\mathfrak{Q}_i$, $1 \leqq i \leqq n$, haben dann alle die gleiche erste Komponente a, und dieses Attribut ist verschieden von dem entsprechenden Attribut einer anderen Tabelle. Daher haben Sätze von Objekten aus verschiedenen Tabellen $\mathfrak{Q}_i, \mathfrak{Q}_j$, $1 \leqq i < j \leqq n$, immer leeren Durchschnitt. Damit sind aber die Voraussetzungen des Satzes 5.1 immer erfüllt, d.h. $(\mathfrak{Q}_1, \ldots, \mathfrak{Q}_n)$ ist eine Zerlegung. Die Datenstrukturen dieser Zerlegung haben folgende Gestalt:

$$\begin{array}{ll} D_i\colon & (\omega_1,\ a_i,\ v_1) \\ & (\omega_2,\ a_i,\ v_2) \\ & \ \ \vdots \quad\ \ \vdots \quad\ \vdots \\ & (\omega_{ni},\ a_i,\ v_{ni}). \end{array}$$

Die Standardfragen für D_i sind alle Fragen der Form (a_i, v_j), $j = 1, \ldots, n_i$. Da die Attribut-Komponente keine informative Funktion mehr erfüllt, kann man sie weglassen. Faßt man dann alle (ω, v)-Paare mit gleicher v-Komponente zusammen, so läßt sich jedes D_i folgendermaßen darstellen:

$$\begin{array}{lll} D_i \triangleq & (v_1^i\colon \Omega_1^i), & \quad i = 1, \ldots, n \\ & (v_2^i\colon \Omega_2^i) & \\ & \ \ \vdots \quad\ \ \vdots & \\ & (v_{ni}^i\colon \Omega_{ni}^i). & \end{array}$$

Dabei ist Ω_j^i die Menge der Objekte ω mit gleicher v-Komponente in D_i, und die Werte v_j^i sind paarweise verschieden. Diese Darstellung entspricht einer (invertierten) Datei mit den „Schlüsselwerten" $v_1^i, \ldots, v_{ni}^i$: ein Eintrag unter v_j^i verweist auf alle Objekte, die die Standardfrage (a_i, v_j^i) beantworten.

Diese Verfahrensweise entspricht der Realisierung eines Q-Systems durch Zerlegung in eine „direkte" und $n-1$ „invertierte" Dateien: die Schlüsselwerte der direkten Datei sind typischerweise Satz-Identifikationsnummern, auf die die ω-Einträge der invertierten Dateien verweisen, und deren ω-Einträge sind die entsprechenden Sätze R_ω [2].

Die andere Möglichkeit, mit dem gegebenen Q-System $\mathfrak{Q}$ zu verfahren, nämlich die direkte Umwandlung in eine Tabelle nach Satz 3.3, läuft im Prinzip auf folgendes hinaus: man legt eine Datei mit den (geeignet codierten) Fragen $q \in Q$

als Schlüsselwerten und den Antwortmengen $\alpha(q)$ als Einträgen an:

$$\begin{array}{l}(q_1 \colon \alpha(q_1))\\(q_2 \colon \alpha(q_2))\\ \vdots \quad \vdots\\(q_r \colon \alpha(q_r)).\end{array}$$

Dabei ist $Q = q_1, \ldots, q_r$. Jede Frage aus Q ist dann durch einen einfachen Tabellenzugriff zu beantworten. Der Nachteil dieser Methode ist, daß der Speicheraufwand meist unvertretbar groß wird. Die obige Datei hat

$$\sum_{i=1}^{r} |\alpha(q_i)|$$

Einträge, und unter diesen Einträgen kommen nur

$$|\Omega| = \left|\bigcup_{i=1}^{r} \alpha(q_i)\right|$$

verschiedene Objekte vor. Die Redundanz kann daher beträchtlich werden.

Der Speicheraufwand läßt sich dadurch reduzieren, daß man Q in disjunkte Klassen $Q_1, \ldots, Q_n$ aufteilt und eine Datei der folgenden Form anlegt:

$$\begin{array}{l}(Q_1 \colon \Omega_1)\\(Q_2 \colon \Omega_2)\\ \vdots \quad \vdots\\(Q_n \colon \Omega_n).\end{array}$$

Dabei ist

$$\Omega_i := \bigcup_{q \in Q_i} \alpha(q), \qquad i = 1, \ldots, n.$$

Zur Beantwortung einer Frage hat man dann die Element-Funktion $q \mapsto Q_i$ mit $q \in Q_i$ zu implementieren, mit der man auf den Tabelleneintrag $(Q_i \colon \Omega_i)$ zugreifen kann. Dadurch hat man das ursprüngliche Problem, q in $\mathfrak{Q} = (D, Q)$ zu beantworten, auf das (kleinere) Problem zurückgeführt, q in $\mathfrak{Q}_i = (D_i, Q_i)$ zu beantworten, wobei $D_i = (\Omega_i, \mathrm{pr}_1(Q_i), \mathrm{pr}_2(Q_i), \varrho_i^*)$ und ϱ_i^* die Einschränkung von ϱ auf die genannten Mengen ist. Für jeden Eintrag $(Q_i \colon \Omega_i)$ der obigen Datei ergibt sich so ein korrespondierendes Q-System $\mathfrak{Q}_i$, und $(\mathfrak{Q}_1, \ldots, \mathfrak{Q}_n)$ ist eine Zerlegung von $\mathfrak{Q}$; denn wenn für eine Frage $q \in Q_i$ gilt, daß $q < R_\omega$ ist in $\mathfrak{Q}$, so ist immer $q < R_\omega^i$ in $\mathfrak{Q}_i$, d.h. es ist $\alpha_i(q) = \alpha(q)$ nach Satz 5.1.

Nach diesem Grundprinzip verfahren die „balanced-file organization schemes" von Abraham, Ghosh und Ray-Chaudhury [1, 7, 8, 12], die Methode von Bose, Abraham und Ghosh [3] und die „new balanced-file organization schemes" von Chow [4].

Die Komponenten $\mathfrak{Q}_i$ der Zerlegung von $\mathfrak{Q}$ haben typischerweise nur kleine Fragemengen. Für diesen Fall kommen weitere Implementierungsmethoden in Betracht, z.B. die Anwendung von Satz 3.2. Stellen wir $\mathfrak{Q}_i = (D_i, Q_i)$ mit Hilfe der dort beschriebenen Konstruktion durch ein elementares und boolesches Q-System $\mathfrak{Q}_i^* = (D_i^*, Q_i^*)$ mit einfacher Datenstruktur D_i^* dar, so können wir den Satz jedes Objektes aus $\Omega_i^* = \alpha(Q_i)$ als Bitstring darstellen, wobei jeder Frage aus

Q_i genau eine Position in diesem Bitstring zugeordnet ist. Der Wert 0 oder 1 in einer Position zeigt an, ob das betreffende Objekt in der Antwort auf die entsprechende Frage enthalten ist oder nicht. Die Bestimmung von $\alpha_i^*(q)$ geschieht dann durch einen einfachen Suchalgorithmus.

Eine weitere Möglichkeit ist die nochmalige Anwendung von Satz 3.3: die Fragen $q \in Q_i$ werden in einer Tabelle angeordnet, und jeder Tabelleneintrag bildet den Kopf einer Link-Kette, in der die Objekte aus der Antwort auf diese Frage miteinander verkettet sind. Man kommt so zu den bekannten Link- und Gittertechniken (s. [9]).

Die direkte Umwandlung des gegebenen Q-Systems $\mathfrak{Q}$ in eine äquivalente Tabelle nach Satz 3.3 führt, wie schon erwähnt, i.a. zunächst zu einer hohen Redundanz. Die Bildung von Klassen von Fragen ist eine Möglichkeit, diese Redundanz zu verringern. Eine andere interessante Möglichkeit ist von Lum [11] beschrieben worden. Hier werden Inklusionsbeziehungen zwischen Fragen und Antworten ausgenutzt. Es gilt nämlich für Fragen $q, q' \in Q$ und die Antwortfunktion α mit Bezug auf eine Datenstruktur D:

$$\begin{aligned} q \subset q' &\Rightarrow \exists q'' \subset A \times V: q' = q \cup q'' \\ &\Rightarrow \exists q'' \subset A \times V: \alpha(q') = \alpha(q) \cap \alpha(q'') \\ &\Rightarrow \alpha(q') \subset \alpha(q). \end{aligned}$$

Bei der Methode von Lum werden nun gewisse Antworten, die in anderen enthalten sind, als zusammenhängender Block innerhalb einer umfassenden Antwort gespeichert, und die Inklusionsbeziehungen zwischen Fragen werden zur Adreßberechnung dieses Antwortblocks herangezogen.

Literatur

1. Abraham, C. T., Ghosh, S. P., Ray-Chaudhury, D. K.: File organization schemes based on finite geometries. Information and Control **12**, 143—163 (1968)
2. Bloom, B. H.: Some techniques and trade-offs effecting large data base retrieval times. Proc. 24th Nat. Conf. ACM, Boston 1969, p. 83—95
3. Bose, R. C., Abraham, C. T., Ghosh, S. P.: File organization of records for multiple valued attributes for multi-attribute queries. IBM Res. Rep. RC-1886, 1967
4. Chow, D. K.: New balanced-file organization schemes. Information and Control **15**, 377—396 (1969)
5. Ehrich, H.-D.: Datenstrukturen und Q-Systeme — eine mathematische Studie. In: Brauer, W. (Hrsg.): Gesellschaft für Informatik, 3. Jahrestagung, Hamburg 1973. Lecture Notes in Computer Science **1**. Berlin-Heidelberg-New York: Springer 1973, p. 363—371
6. Ehrich, H.-D.: Grundlagen einer Theorie der Datenstrukturen und Zugriffssysteme. Teil I: Datenstrukturen und Schemata. Acta Informatica **4**, 201—211 (1975)
7. Ghosh, S. P.: Organization of records with unequal multiple-valued attributes and combinatorial queries of order 2. Information Sciences **1**, 363—380 (1969)
8. Ghosh, S. P.: On the problem of query oriented filing schemes using discrete mathematics. Proc. IFIP Congr. 68. Amsterdam: North-Holland 1969, p. 1226—1232
9. Knuth, D. E.: The art of computer programming, vol. 1: Fundamental algorithms. Reading (Mass.): Addison-Wesley 1968
10. Krägeloh, K.-D., Lockemann, P. C.: Struktur eines Frage-Antwort-Systems auf mengentheoretischer Grundlage. GMD Bericht Nr. 55, Birlinghoven 1972

11. Lum, V. Y.: Multi-attribute retrieval with combined indexes. Comm. ACM **13**, 660—665 (1970)
12. Ray-Chaudhury, D. K.: Combinatorial information retrieval systems for files. SIAM J. Appl. Math. **16**, 973—992 (1968)
13. Turski, W. M.: On a model of information retrieval systems based on thesaurus. Information Storage and Retrieval **7**, 89—94 (1971)

H.-D. Ehrich
Abteilung Informatik
Universität Dortmund
D-4600 Dortmund-Hombruch
Postfach 500
Bundesrepublik Deutschland