

**Seminar: Complex Objects in Data Bases**  
**Multi-dimensional Aggregation of Temporal  
Data**

Andreas Bierfert, Jost Enderle  
December 8, 2006

**Chair of Computer Science 9**  
**Data Management and Exploration**  
**Univ.-Prof. Dr. rer. nat. Thomas Seidl**  
**RWTH Aachen**

## Contents

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Bezeichnungen und Definitionen</b>	<b>3</b>
2.1	Zeitabhängige Daten (temporal data) . . . . .	3
2.2	Zusammenfassungsoperator (aggregate operator) . . . . .	3
2.3	Zeitintervall (timestamp/time interval) . . . . .	3
2.4	Relationales Datenbankschema (relation schema) . . . . .	3
2.4.1	Zeitabhängiges, relationales Datenbankschema (temporal relation schema) . . . . .	4
2.5	Charakteristika von Attributen (attribute characteristics) . . . . .	4
2.6	Anpassung von Attributwerten (Adjustment of Attribute Values) . . . . .	4
2.7	Konstante Zeitintervalle (Constant Intervals) . . . . .	5
2.7.1	Mächtigkeit (Cardinality) . . . . .	5
2.8	Feste Zeitintervalle (Fixed Intervals) . . . . .	5
2.9	Faktor der Zusammenfassung (Aggregation factor) . . . . .	5
<b>3</b>	<b>Zeitabhängiger, multidimensionaler Zusammenfassungsoperator TMDA</b>	<b>6</b>
3.1	Definition . . . . .	6
3.2	Stückweise Definition von Ergebnisgruppen . . . . .	6
3.3	Realisierung . . . . .	7
3.3.1	Konstante Intervalle (TMDA-CI) . . . . .	7
3.3.2	TMDA-CI Algorithmus . . . . .	7
3.3.3	festgelegte Intervalle (TMDA-FI) . . . . .	8
3.3.4	TMDA-FI Algorithmus . . . . .	9
3.3.5	Komplexität . . . . .	9
3.4	Beispielanwendung . . . . .	10
<b>4</b>	<b>Zusammenfassung</b>	<b>10</b>

## 1 Einleitung

Das Paper *Multi-dimensional Aggregation of Temporal Data* beschäftigt sich mit der Evaluation von zeitabhängigen Daten in einem DBMS-System. Es werden verschiedene aktuelle Ansätze betrachtet und ein Operator vorgestellt, der im Vergleich zu bisher bekannten Verfahren auf effiziente Weise das Problem von zeitabhängigen Datensätzen löst. In dieser Ausarbeitung werden die wichtigsten Definitionen erläutert und der Operator mit Algorithmen und einem abschließenden Beispiel vorgestellt.

## 2 Bezeichnungen und Definitionen

Im Folgenden werden die für das Verständnis wichtigen Bezeichnungen erläutert und einige für das Thema grundlegende Definitionen vorgestellt.

### 2.1 Zeitabhängige Daten (temporal data)

Zeitabhängige Daten sind Daten, die auf irgendeine Weise mit einem Zeitintervall verbunden sind, welches besondere Merkmale der Daten innerhalb des jeweiligen Zeitintervalls berücksichtigt. In etwa, daß ein Merkmal  $X$  nur auf dem Zeitintervall  $[a; b]$  gültig ist.

### 2.2 Zusammenfassingsoperator (aggregate operator)

Ein Zusammenfassingsoperator erreicht durch die Partitionierung der Argumente (**argument relation**) in Gruppen von Tupeln (mit identischen Werten) und dem berechnen einer Aggregationsfunktion wie z.B. dem Durchschnitt (**average**) oder der Aufsummierung (**sum**) für die einzelnen Gruppen eine Zusammenfassung (**summary result relation**) der Ergebnisse aus den Argumenten.

### 2.3 Zeitintervall (timestamp/time interval)

Ein Zeitintervall besteht aus einer konvexen Menge von Zeitpunkten (**chronons**) die mit  $[T_s, T_e] =: T$  bezeichnet wird. Die Zeitpunkte sind Elemente aus einem diskreten Zeitbereich  $D^T$  mit der totalen Ordnung  $<^T$ . Für die Zeitpunkte gelten folgende Regeln:

- $t \in T$  - Zeitpunkt  $t$  ist in der Menge  $T$  enthalten
- $T, T'$  Zeitintervalle,  $T' \subseteq T$  gdw.  $\forall t(t \in T \rightarrow t \in T')$
- $T \cap T'$  - Zeitpunkte die in  $T$  und  $T'$  enthalten sind
- $T \cap T' \neq \emptyset$  - Die Zeitinvalle überschneiden sich

### 2.4 Relationales Datenbankschema (relation schema)

Ein relationales Datenbankschema  $S = (\Omega, \Delta, dom)$  besteht aus einer nicht-leeren Attributmenge  $\Omega$ , endlichen Intervallmengen  $\Delta$  und einer Funktion  $dom : \Omega \rightarrow \Delta$ , die Intervallmengen mit den Attributen verbindet.

### 2.4.1 Zeitabhängiges, relationales Datenbankschema (temporal relation schema)

Ein zeitabhängiges, relationales Datenbankschema beinhaltet mindestens ein Zeitintervall. Der Zeitbereich gehört hierbei zu  $\Delta$ . Für die weitere Verwendung benutzen wir die zeitabhängigen Schemata  $R = (A_1, \dots, A_n, T)$ ,  $G = (B_1, \dots, B_n, T)$ . Hierbei steht das Zeitintervall o.B.d.A. an letzter Stelle um eine bessere Übersicht zu haben.

Ein Tupel über Schema  $S = (\Omega, \Delta, dom)$  ist eine Funktion  $r : \rightarrow \cup_{\delta \in \Delta} \delta$ , so daß für jedes Attribut gilt:  $A \in \Omega, r(A) \in dom(A)$ . Dabei heißt das Tupel zeitabhängig, wenn das zugehörige Schema zeitabhängig ist. Wie bei den Schemata wird die Ordnung im Tupel o.B.d.A. wie folgt definiert:  $r = (v_1, \dots, v_n, t)$ . Eine Relation über einem Schema  $R$  ist eine endliche Menge von Tupeln bezeichnet mit  $\mathbf{r}$ .

Für ein Tupel  $r$  und ein Attribut  $A$  gilt die Regel:  $A$  ist Attribut von  $r$ , dann steht  $r.A$  für den Wert des Attributs. Für eine Menge von Attributen gilt somit:  $r[A_1, \dots, A_m] = (r.A_1, \dots, r.A_m)$ .

## 2.5 Charakteristika von Attributen (attribute characteristics)

Es werden drei verschiedene Charakteristika der Verbindung von zeitabhängigen Attributen zu zeitunabhängigen Attributen festgelegt: konstant, veränderbar und atomar (**constant, malleable, atomic**). Für ein Schema  $R$  mit  $R = (A_1, \dots, A_n, T)$  werden die Charakteristika im Bezug auf  $T$  angegeben mit:  $C_T = (c_1, \dots, c_n), c_i \in \{c, m, a\}; i \in \{1, \dots, n\}$ .

Als Beispiel gilt für ein Schema  $(X, Y, Z, T)$  und die Charakteristika  $C_T = (c, m, a)$  das  $X$  konstant,  $Y$  veränderbar und  $Z$  atomar im Bezug auf  $T$  sind.

## 2.6 Anpassung von Attributwerten (Adjustment of Attribute Values)

Nach der Definition von Charakteristika für Attribute im Bezug auf den zeitlichen Kontext ist es wichtig, daß bei einer Datenanfrage die Attribute entsprechend ihrer Charakteristika ausgewertet werden. Dies wird an einem kleinen Beispiel schnell deutlich:

Angenommen wir haben ein Tupel  $r = (243791, 100, [2006/08/15, 2006/09/15])$ , wobei der erste Eintrag eine Matrikelnummer der RWTH darstellt, die zweite Nummer die Anzahl der Arbeitsstunden für eine Seminararbeit, in dem durch den dritten Eintrag festgelegten Zeitrahmen. Wird nun auf dem Zeitintervall  $I = [2006/08/15, 2006/09/01]$  gesucht, so muß natürlich die Matrikelnummer nicht angepasst werden, die Anzahl der Stunden (bei angenommener Gleichverteilung auf die angegebenen Tage) jedoch schon. Somit ergeben sich für das Tupel  $r$  die Charakteristika  $C = (c, m)$ . Die Anpassung der Werte mit den Charakteristika  $C$  liefert dann das folgende Ergebnis:  $(243791, 50, I)$ .

Allgemein definiert man hierfür die Anpassungsfunktion *adj*.

Gegeben sei das Tupel  $r = (v_1, \dots, v_n, t)$  über dem Schema  $(A_1, \dots, A_n, T)$ , ein Zeitintervall  $I$  und die Charakteristika  $C = (c_1, \dots, c_n)$ . Dann ist die Anpassungsfunktion wie folgt definiert:

$$adj(r, I, C) = (adj(r.A_1, r.T, I, c_1), \dots, adj(r.A_n, r.T, I, c_n), I)$$

$$adj(v, T, I, c) = \begin{cases} v & c = c' \\ v \cdot |I \cap T|/|T| & c = m' \\ v & c = a' \wedge T = I \\ UNDEF & c = a' \wedge T \neq I \end{cases}$$

## 2.7 Konstante Zeitintervalle (Constant Intervals)

Ein konstantes Zeitintervall ist ein maximales, nicht überlappendes Intervall in dem die Menge der Argumenttupel konstant ist. Wichtig sind hierbei zwei Dinge:

- Die Ergebnisintervalle dürfen die Grenzen der Argumentintervalle nicht überschreiten.
- Die Ergebnisintervalle müssen maximal sein.

Formal ergibt sich:

$r$  sei eine zeitabhängige Relation abhängig von einem Zeitintervall  $T$ . Dann ergeben sich die konstanten Intervalle von  $r$  mit:

$$CI(r) = \{T | \forall r \in r(r.T \supseteq T \vee r.T \cap T = \emptyset) \wedge \forall T' \supset T (\exists r \in r(r.T \not\supseteq T' \wedge r.T \cap T' \neq \emptyset))\}$$

### 2.7.1 Mächtigkeit (Cardinality)

Die Mächtigkeit eines konstanten Intervalls über einer zeitabhängigen Relation  $r$  mit  $n$  Tupeln ( $n > 0$ ) ist durch folgende Formel gegeben:

$$|CI(r)| \leq 2n - 1$$

Der Beweis hierzu ergibt sich aus der Beschaffenheit der Zeitintervalle der Tupel, denn diese können (mit der Ordnung  $<^T$ ) linear geordnet werden, so daß es maximal  $2n$  Zeitpunkte gibt. Da  $n$  Zeitpunkte aber maximal  $n - 1$  aufeinander folgende Zeitintervalle ergeben können, ergibt sich die o.g. Formel.

## 2.8 Feste Zeitintervalle (Fixed Intervals)

Von festen Zeitintervallen spricht man, wenn sich überlappende Zeitintervalle verwendet werden. Wichtig ist hierbei, daß sich die festen Intervalle mit Intervallen der Argumentenrelation schneiden. Somit kann man mit einer angemessenen Definition der Ergebnisse die Mächtigkeit der Ergebnisrelation steuern. Allgemein gilt für feste Zeitintervalle:

$$\forall T \in FI(r) (\exists r \in r(r.T \cap T \neq \emptyset))$$

## 2.9 Faktor der Zusammenfassung (Aggregation factor)

Das Verhältnis des zeitabhängigen Zusammenfassungsoperators ergibt sich aus der Mächtigkeit von Ergebnisrelation  $z$  und Argumentenrelation  $r$  als

$$af = |z|/|r|$$

### 3 Zeitabhängiger, multidimensionaler Zusammenfassungsoperator TMDA

#### 3.1 Definition

Der wichtigste Teil dieser Arbeit beschäftigt sich mit dem zeitabhängigen, multi-dimensionalen Zusammenfassungsoperator kurz TMDA. Die Definition der Ergebnisgruppen ist nun unabhängig von der Verbindung zu den Eingabetupeln. Der TMDA ist wie folgt definiert:

$\mathbf{r}, \mathbf{g}$  sind Relationen mit Zeitintervall  $T$ .  $\mathbf{F} = \{f_{A_{i_1}}, \dots, f_{A_{i_p}}\}$  sind Zusammenfassungsfunktionen über  $\mathbf{r}$ .  $\theta$  sei die Mächtigkeit der Attribute von  $\mathbf{g}$  und  $\mathbf{r}$ .  $C$  sind die Charakteristika der Attribute von  $\mathbf{r}$ . Dann gilt für den TMDA mit der Abbildung  $\pi$ , die zur Sicherstellung von doppelten Einträgen verwendet wird:

$$G^T[\mathbf{F}][\theta][T][C](\mathbf{g}, \mathbf{r}) = \\ \{x | g \in \mathbf{g} \wedge \mathbf{r}_g = \{\{r' | r \in \mathbf{r} \wedge \theta(g, r) \wedge r' = adj(r, g, T, C)\}\} \\ \wedge x = g(f_{A_{i_1}}(\pi[A_{i_1}](\mathbf{r}_g)), \dots, f_{A_{i_p}}(\pi[A_{i_p}](\mathbf{r}_g)))\}$$

$r$  ist hierbei eine Argumentenrelation, und  $g$  die Gruppenrelation, welche die Ergebnisgruppen festlegt mit denen die Ergebnistupel berechnet werden.  $\theta$  verbindet eine Zusammenfassungsgruppe  $\mathbf{r}_g \subseteq \mathbf{r}$  mit jedem  $g \in \mathbf{g}$ . Hierbei werden auch mit der  $adj$ -Funktion die Zeitintervalle entsprechend  $T$  angepasst. Zum Schluß werden noch die Zusammenfassungsfunktionen für jede Zusammenfassungsgruppe berechnet, wobei für jede Zusammenfassung je eine Spalte genutzt wird.

#### 3.2 Stückweise Definition von Ergebnisgruppen

TMDA benötigt komplett definierte Gruppenrelationen. Da die Zeitintervalle der Ergebnistupel (im konstanten Fall) aus den Argumenttupeln berechnet werden, sind sie nicht im Vorhinein verfügbar. Um diese Situation zu verbessern, können mit einem einfachen Ausdruck die konstanten Zeitintervalle berechnet werden, so daß de facto nur mit festen Zeitintervallen gerechnet werden kann. Es ergibt sich:

$$G^T[\mathbf{F}][\theta \wedge overlap(\mathbf{g}, T, \mathbf{r}, T)][T][C](CI(\mathbf{g}', \mathbf{r}, \theta) / \mathbf{g}, r)$$

So günstig dies auch klingt, desto kostspieliger ist die Berechnung, da die hierzu verwendeten Operationen rechenaufwendig sind. Hieraus ergeben sich die stückweise Definition von Ergebnisgruppen (**Partially Specified Result Groups**). Mit diesem Schritt wird die Berechnung dem TDMA-Algorithmus übergeben. Hier wird beim Einlesen der Datenrelation on-the-fly die Berechnung durchgeführt und die  $overlap$ -Relation implizit ausgeführt.

Eine Ergebnisgruppe mit dem Schema  $G = (B_1, \dots, B_m, T)$  ist dann stückweise definiert, wenn der Wert des Zeitintervalls nicht angegeben ist. Das Ergebnistupel wird dann dargestellt als  $g = (v_1, \dots, v_m, [*, *])$ . Eine Umwandlung in das Format herkömmlicher, zeitabhängiger Zusammenfassungsoperatoren ist auf einfache Weise möglich.

### 3.3 Realisierung

#### 3.3.1 Konstante Intervalle (TMDA-CI)

Bei konstanten Intervallen funktioniert eine Evaluation wie folgt: An einem Zeitpunkt  $t$  kann man mit den Werten der Argumentrelation die Ergebnistupel, deren Zeitintervalle vor  $t$  enden, berechnen. Somit werden bei der Ausführung laufend neue Ergebnistupel erzeugt und nur solche im Speicher gehalten, die für den aktuellen Zeitkontext  $t$  gültig sind (*open tuples*). Für den TMDA-CI Algorithmus werden fünf Parameter für die Berechnung von  $G^T$  benötigt:

1. Gruppenrelation  $\mathbf{g}$
2. Argumentrelation  $\mathbf{r}$
3. Zusammenfassungsfunktionen  $\mathbf{F} = \{f_{A_{i_1}}, \dots, f_{A_{i_p}}\}$
4. Entscheidungsprädikat  $\theta$
5. Charakteristika  $C$

Im Algorithmus werden zwei Datenstrukturen verwendet: Eine Gruppentabelle  $gt$  mit den Tupeln  $g \in \mathbf{g}$  mit einem Zeiger auf den Endpunktbaum  $T$ , sowie der Endpunktbaum  $T$ , der die potentiellen Endpunkte der konstanten Intervalle beinhaltet.

#### 3.3.2 TMDA-CI Algorithmus

```
if  $\mathbf{g} = \pi[A_1, \dots, A_m](\mathbf{r})$  then
 $gt \leftarrow$  empty group table with columns  $B_1, \dots, B_m, T, T$ 
else initialize  $gt$  with  $(g, \text{empty } T), g \in \mathbf{g}$ , and replace timestamp  $T$  by  $[\infty, \cdot]$ ;
```

Zunächst wird die Gruppentabelle initialisiert, wenn  $\mathbf{g}$  eine Projektion über  $\mathbf{r}$  ist, dann ist die Gruppentabelle am Anfang leer und wird beim Einlesen der Argumententupel gefüllt. Ansonsten wird  $gt$  mit  $\mathbf{g}$  initialisiert und die Startzeiten der Einträge auf  $-\infty$  gesetzt und ein leerer Endpunktbaum für jeden Eintrag generiert.

```
create index for  $gt$  on attributes  $B_1, \dots, B_m; \mathbf{z} \leftarrow \emptyset$ ;
```

Dann wird ein Index über alle zeitunabhängigen Attribute generiert.

```
foreach tuple  $r \in \mathbf{r}$  in chronological order do
```

Nun wird unter Berücksichtigung der Startzeiten (und  $<^T$ ) die Argumentenrelation  $\mathbf{r}$  abgearbeitet.

```
if  $\mathbf{g} = \pi[A_1, \dots, A_m](\mathbf{r})$  and  $r.A_1, \dots, r.A_m$  not yet in  $gt$  then
insert  $(r.A_1, \dots, r.A_m, [-\infty, \cdot], \text{empty } T)$  into  $gt$ ;
```

Wenn es sich bei der Gruppenrelation um relationale Algebra handelt und dieser Ausdruck noch nicht in der Gruppentabelle enthalten ist, wird dieser

hinzugefügt.

foreach  $i \in \text{Lookup}(gt, r, \theta)$  do

Die *Lookup*-Funktion prüft für ein Datentupel, zu welcher Ergebnisgruppe es gehört. Für jede passende Ergebnisgruppe werden zwei Dinge durchgeführt:

if  $r.T_s > gt[i].T_s$  then

insert a new node with time  $r.T_s - 1$  into  $gt[i].T$  (if not already there);

foreach  $v \in gt[i].T$  in chronological order, where  $v.t < r.T_s$  do

$gt[i].T_e \leftarrow v.t$ ;

$\mathbf{z} \leftarrow \mathbf{z} \cup \text{ResultTuple}(gt[i], \mathbf{F}, C)$ ;

$gt[i].T \leftarrow [v.t + 1, \cdot]$ ;

remove node  $v$  from  $gt[i].T$ ;

Wenn  $r$  bis in das aktuelle Zeitintervall reicht, also  $r.T_s > gt[i].T_s$ , können ein oder mehrere konstante Intervalle geschlossen werden. Hierbei handelt es sich bei  $r.T_s - 1$  um einen potentiellen Endpunkt des konstanten Intervalls und daher wird er in  $gt[i].T$  eingefügt. Dann werden alle Blätter  $v$  des Baumes  $gt[i].T$  durchlaufen, bei denen die Bedingung  $v.t < r.T_s$  erfüllt ist. Das Ende des Zeitintervalls wird entsprechend gesetzt, die Ergebnistupel generiert und das Blatt aus dem Baum entfernt.

$v \leftarrow$  node in  $gt[i].T$  with time  $v.t = r.T_e$  (insert a new node if required);

$v.open \leftarrow v.open \cup r[A_1, \dots, A_p, T_s]$ ;

Nun wird der Endpunktbaum mit dem neuen Datentupel aktualisiert.

foreach  $gt[i] \in gt$  do

foreach  $v \in gt[i].T$  in chronological order do

create result tuple, add it to  $\mathbf{z}$ , and close past nodes in  $gt[i].T$ ;

return  $\mathbf{z}$

Am Ende werden die Ergebnistupel in  $\mathbf{z}$  zurückgegeben.

Die Funktion *Lookup* sucht zu einem Tupel  $r$ , der Gruppentabelle  $gt$  und dem Kriterium  $\theta$  die passende Ergebnisgruppe. Hierzu wird ein AVL-Baum verwendet (beim TMDA-FI sogar zwei AVL-Bäume). Die *ResultTuple*-Funktion berechnet aus der Gruppentabelle  $gt[i]$ , den Zusammenfassungsfunktionen  $\mathbf{F}$  und den Charakteristika  $C$  das Ergebnistupel für das konstante Intervall  $gt[i].T$ . Gibt es keine offenen Tupel in dem Intervall, so wird die leere Menge zurückgegeben.

### 3.3.3 festgelegte Intervalle (TMDA-FI)

Für die Berechnung von  $G^T$  wird wie beim TMDA-CI ein Tupel  $(\mathbf{g}, \mathbf{r}, \mathbf{F}, \theta, C)$  benötigt. Hierbei ist  $gt$  die Gruppentabelle, die die Gruppenrelation  $\mathbf{g}$  beinhaltet.  $\mathbf{g}$  ist jedoch im Vergleich zum TMDA-CI-Algorithmus um die Spalten der Zusammenfassungsrelationen  $f_{A_{i,j}} \in \mathbf{F}$  erweitert. Die Ergebnisgruppen sind, wegen der festgelegten Intervalle, komplett bekannt, so daß die Datentupel nicht im Endpunktbaum gespeichert werden müssen.

**3.3.4 TMDA-FI Algorithmus**

```

if  $\mathbf{g} = \pi[A_1, \dots, A_m, \text{cast}(T, G)](r)$  then
 $gt \leftarrow$  empty group table with columns  $A_1, \dots, A_m, T, f_{A_{i_1}}, \dots, f_{A_{i_p}}$ ;
else
initialize  $gt$  to  $\mathbf{g}$  and extend it with columns  $f_{A_{i_1}}, \dots, f_{A_{i_p}}$  initialized to NULL;
Create index for  $gt$  on attribute  $T$ ;
foreach tuple  $r \in \mathbf{r}$  do
if  $g = \pi[A_1, \dots, A_m, T](r)$  then
foreach  $t \in \text{cast}(r.T, G)$  do
Insert  $r.A_1, \dots, r.A_m, t$  into  $gt$  if not already there;
foreach  $i \in \text{Lookup}(gt, r, \theta)$  do
 $r' \leftarrow \text{ADJUST}(r, gt[i].T, C)$ ;
foreach  $f_j \in \mathbf{F}$  do  $gt[i].f_{A_{i_j}} \leftarrow gt[i].f_{A_{i_j}} \oplus r'.A_{i_j}$ ;
return  $gt$ ;

```

**3.3.5 Komplexität****TMDA-CI**

Für eine Komplexitätsanalyse des TMDA-CI ist nur die Berechnung der Datenrelation  $\mathbf{r}$  interessant, die sich in vier wichtige Teile unterteilt:

1. Aktualisierung des Index
2. *Lookup* des Index
3. Berechnung der Ergebnistupel
4. Einfügen der Tupel in den Endpunktbaum

Aktualisierung und Suchen des Index haben eine Komplexität von  $\log n_g$ , wobei  $n_g$  für die Mächtigkeit der Gruppentabelle steht. Die Produktion eines einzelnen Ergebnistupels ist linear über der Nummer offener Tupel  $n_o$ . Die Anzahl der Ergebnistupel eines Datentupels ist abhängig vom Zusammenfassungsfaktor  $af = n_z/n_r$ , wobei  $n_z$  die Mächtigkeit der Ergebnisrelation und  $n_r$  die Mächtigkeit der Datenrelation ist. Desweiteren ist die Anzahl der Ergebnisgruppen zu denen  $r$  beiträgt wichtig. Dieser Zusammenhang wird mit  $n_{g,r}$  bezeichnet. Das Einfügen eines Tupels in den Endpunktbaum hat eine Komplexität von  $\log n_o$ . Somit ergibt sich eine Komplexität von TMDA-CI von  $O(n_r \max(\log n_g, n_{g,r} af n_o, \log n_i))$ . Im Worst-Case gibt es einen Fall, an dem die Start- und Endpunkte aller Intervalle unterschiedlich sind und es einen Zeitpunkt gibt, an dem alle Tupel wahr sind. Dann ergibt sich  $n_o = n_r$  und für die Komplexität  $O(n_r^2)$ .

**TMDA-FI**

Bei festgelegten Intervallen müssen keine offenen Datentupel verwaltet werden und die Werte der Zusammenfassungsfunktionen können beim Lesen der Datenrelation berechnet werden. Daher ergibt sich eine Komplexität für TMDA-FI von  $O(n_r \max(\log n_g, n_{g,r}))$

### 3.4 Beispielanwendung

Zur Veranschaulichung der Algorithmen hier ein Beispieldatensatz einer Feuerwehr:

	Name	PNr.	Abteilung	Dauer	Vergütung	Zeit
$r_1$	FM1	1	A	4	7	[11,15]
$r_2$	FM1	1	A	4	8	[18,22]
$r_3$	FM2	2	VB	2	12	[12,14]
$r_4$	FM2	2	VB	1	12	[14,15]
$r_5$	FM3	3	A	6	9	[17,23]
$r_6$	FM3	3	A	2	7	[13,15]

Die Abfrage, die wir mit Hilfe des TMDA-CI lösen wollen, sieht wie folgt aus:

Wir wollen wissen, zu welchen Zeitintervallen die meisten Kosten entsehen pro Abteilung. Mit  $<^T$  erhalten wir für die Datensätze folgende Ordnung:

$r_1, r_3, r_6, r_4, r_5, r_2$

Für  $gt$  ergibt sich mit  $r_1$ :

Abteilung	Zeitintervall	$T$
A	[11, ·]	$T_1$
VB	$[-\infty, \cdot]$	$T_2$

Der Endpunktbaum hat dann den Eintrag  $15 - \{3, 7, 11\}$  und die Ergebnisrelation  $\mathbf{z}$  ist leer. Das nächste Tupel ist  $r_3$ , welches zur Abteilung  $VB$  gehört. Da es pro Abteilung einen Endpunktbaum gibt, folgt das Ergebnis analog zum ersten Schritt. Nach der Berechnung von  $r_6$  wird  $r_4$  als letzter Datensatz von der Abteilung  $VB$  berechnet. Da jedoch das Zeitintervall von  $r_4$  später beginnt als das Ende von  $r_3$ , wird das aktuell offene Intervall geschlossen und ein Ergebnistupel erzeugt. Als nächstes ist Datensatz  $r_5$  an der Reihe. Da aber  $r_5$  ausserhalb des offenen Intervalls liegt, wird das erste Ergebnistupel berechnet und der Baum angepasst, so daß nur noch  $r_5$  enthalten ist. Die Ergebnisrelation sieht dann wie folgt aus:

Abteilung	Zeitintervall	Summe Stunden	Summe Vergütung
A	[13 – 15]	6	14

Führt man den Algorithmus zu Ende aus ergibt sich folgende Ergebnisrelation:

Abteilung	Zeitintervall	Summe Stunden	Summe Vergütung
A	[13 – 14]	6	14
A	[17 – 18]	1	9
A	[18 – 22]	8	17
A	[22 – 23]	1	9
VB	[12 – 14]	2	12
VB	[14 – 15]	1	12

## 4 Zusammenfassung

Der TMDA-Operator ist ein wichtiger Fortschritt bei der Verarbeitung von zeitabhängigen Daten. Zwar gab es, wie in der Ausarbeitung von Böhlen, Gamper

und Jensen erwähnt, schon einige andere Ansätze, die hier skizzierten Probleme zu lösen, jedoch waren diese meist nicht so effizient oder konnten für bestimmte Bedingungen (Vielfalt der Zusammenfassungsfunktionen) nur schlechte Ergebnisse liefern. Nicht nur ist der TMDA-Operator, wie in der Komplexitätsanalyse gezeigt effizient, sondern skaliert er auch auf Testdatensätzen im Vergleich zu anderen Algorithmen gut. In der Zukunft können die beiden Algorithmen TMDA-CI und TMDA-FI noch weiter verbessert werden, hierzu wird von den Autoren vorgeschlagen, z.B. die Initialisierung der Gruppentabelle on-the-fly zu machen oder die Gruppentabelle mit einem Index zu versehen oder mit verschiedenen Baummodellen für den Endpunktbaum zu experimentieren. In jedem Fall bietet der hier vorgestellte Operator gute und weitreichende Möglichkeiten, um mit zeitabhängigen Daten zu arbeiten und in Zukunft den Umgang mit diesen noch zu verbessern.

## References

- [1] Multi-dimensional Aggregation for Temporal Data, Michael Böhlen, Johann Gamper, and Christian S. Jensen