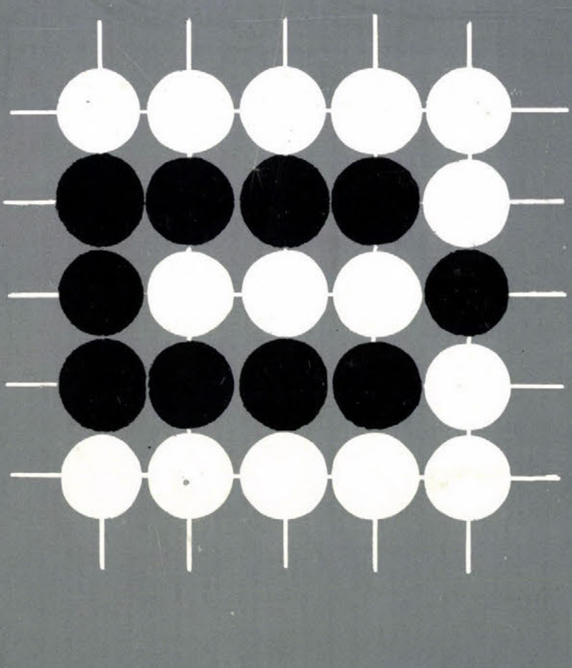
közlemények

36/1987

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



Magyar Tudományos Akadémia
Számítástechnikai és Automatizálási Kutató Intézete
Computer and Automation Institute, Hungarian Academy of Sciences

K Ö Z L E M É N Y E K

T R A N S A C T I O N S

36/1987

Felelős kiadó:

DR. KEVICZKY LÁSZLÓ

ISBN 963 311 229 X
ISSN 0133-7459

CONTENTS

TARTALOMJEGYZÉK

MTA SZTAKI Közlemények 36/1987 pp. 5-29

ЦИКЛИЧЕСКИЕ И АЦИКЛИЧЕСКИЕ СХЕМЫ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

ГАЛЯ АНГЕЛОВА

Институт математики Болгарской Академии Наук

В этой работе рассматриваются понятия ациклической схемы и циклической схемы реляционной базы данных. Эти понятия введены в [BFMMUY81] с целью дать ответ на вопрос - при каких условиях каждое состояние базы данных, которое является попарно совпадающим (pairwise consistent), является также сплошь совпадающим (join consistent). Этот вопрос поставлен в 1976-ом году, но его ответ дан в [BFMMUY81] в 1981-ом году с введением понятия циклической и ациклической схемы базы данных.

В параграфе 1 описаны основные понятия. В параграфе 3 даны определения попарного и сплошного совпадения. В параграфе 4 рассмотрен алгоритм редукции гиперграфов, который в настоящей работе используется для содержательного определения понятия ацикличности (вместо формального определения ациклического гиперграфа, использованного например в [Fa983] и в [Fa983a]). В параграфе 5 исследованы некоторые свойства гиперграфов в процессе редукции и доказано существование так называемого поглощающего ребра для ациклических гиперграфов.

1. Основные понятия

Codd [Cod70] предложил реляционную модель баз данных. В классической модели реляционных баз данных рассматривается множество атрибутов (attributes) U и множество отношений (relations) над атрибутами U. В последные годы введено понятие реляционной схемы. Здесь мы вводим основные понятия, используя [Ull82], [Mai83] и [Дри82].

Пусть дано множество атрибутов

$U = \{A_1, A_2, \dots, A_n\}$.

Множество U будем называть универсум (universum).

Реляционной схемой (relational scheme) R будем называть конечное множество атрибутов $\{A_{i1}, A_{i2}, \dots, A_{im}\}$, где $A_{ij} \in U$ для $1 \leq j \leq m$. Реляционные схемы будем обозначать через $R_1, R_2, \dots, R_k$.

Каждому атрибуту A_i сопоставляется множество значений - так называемый домен (domain). Домен атрибута A_i будем обозначать через $dom(A_i)$.

Пусть R - реляционная схема, $R = \{A_1, A_2, \dots, A_n\}$. Отношением r над реляционной схемой R будем называть конечное множество упорядоченных n-ок:

$r = \{<a_1 a_2 \dots a_n> \mid a_i \in dom(A_i),\ 1 \leq i \leq n\}$.

Элементы $<a_1 a_2 \dots a_n>$ будем называть кортежами отношения r. Если r - отношение над реляционной схемой $R = \{A_1, A_2, \dots, A_n\}$, будем обозначать этот факт через r(R) или $r(A_1 A_2 \dots A_n)$.

Пусть U - данное множество атрибутов. Каждое конечное множество реляционных схем $\mathbf{R} = \{R_1, R_2, \dots, R_k\}$ будем называть схемой базы данных (database scheme) тогда и только тогда, когда $R_i \subset U$ для $1 \leq i \leq k$ и $U = R_1 \cup R_2 \cup \dots \cup R_k$.

(Здесь предполагается, что $R_i \neq U$ для $1 \leq i \leq k$).

Пусть $\mathbf{R}=\{R_1, R_2, \ldots, R_k\}$ - схема реляционной базы данных. Каждое множество конкретных отношений $\{r_1, r_2, \ldots, r_k\}$ соответственно над реляционными схемами $\{R_1, R_2, \ldots, R_k\}$ будем называть состоянием базы данных (database state). Будем обозначать состояние через $d=\{r_1, r_2 \ldots, r_k\}$.

Пусть $U=\mathbf{R}$, т.е. будем рассматривать U как одинственная реляционная схема в данной схеме базы данных. Тогда каждое конкретное отношение (состояние) I над схемой U будем называть универсальной реляцией (universal relation, universal instance) [MUV84], [AnZ85].

2. Гиперграфы, соответствующие схемам реляционных баз данных

Одновременно с введением реляционной модели базы данных началось и исследование разных способов организации данных в отдельных массивах (файлях) [Cod70]. Создавались и соответствующие алгоритмы для определении возможной организации данных при наличии заданных ограничений. Например, алгоритм декомпозиции в 3НФ с сохранением фунциональных зависимостей [Улл80] предлагает способ разложения в "группы" атрибутов U, обеспечивая при этом известные преимущества [Cod72] сохранения данных в отдельных отношениях. Этот алгоритм в качестве формальной процедуры может быть использован в автоматизации проектирования реляционных баз данных. Легко видет, что для этого алгоритма всегда существует "выход", который является единственным с точностью до переименования атрибутов и полученных реляционных схем.

В последнее время рассматривается и другой способ разделения атрибутов U на группы, а именно, путем определения U с помощью предикатов [FMU82]. Рассмотрим следующий пример:

ПРИМЕР 1. [FMU82] Пусть атрибуты U - C(курс), T(преподавател), R(аудитория), H(время), S(студент) и G(оценка). При дефинировании универсальной реляции над этими атрибутами записываем:

{ ctrhsg | t преподает c, курс c собирается в r во время h, s получает g по c } .

Будем пользовател тремя неформально заданными отношениями, которые являются предикатами [FMU82] и которые по нашему мнению имеют смысл в реальном мире: "преподает", "собирается в во время", "получает оценку по".

В этом примере универсальная реляция является множеством тех и только тех кортежей, которые проходят через тест, имплицированный каждым предикатом. Следовательно, универсальную реляцию для этого примера можно записать в виде:

$$\{ ctrhsg \mid P1(c,t) \& P2(c,r,h) \& P3(c,s,g) \} , \qquad (1)$$

где P1(c,t) - предикат "t преподает c",

P2(c,r,h) - предикат "c собирается в r во время h" и

P3(c,s,g) - предикат "s получает оценку g по предмету c". □

Отметим, что определение универсальных реляций с помощью предикатов отражает субъективное представление проектанта базы данных о реальном мире и следовательно, вряд ли может быть автоматизированно; ясно, однако, что таким образом также можно обеспечить разделение атрибутов на семантические группы и следовательно некоторую степень нормализации.

Введение гиперграфов соответствует описанию универсальной реляции с помощью предикатов.

ОПРЕДЕЛЕНИЕ 1. [FMU82] Пусть **E**-конечное множество вершин и **N**-конечное множество ребер, при чем ребра являются непустыми множествами вершин. Тогда упорядоченную пару H=(**E**,**N**) будем называть <u>гиперграфом</u>. □

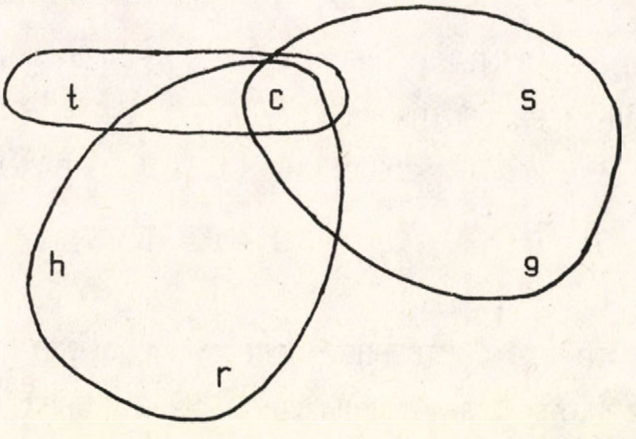
Гиперграф является обобщением обычного графа [Кри78], где каждому ребру соответствуют ровно две вершины.

Каждой универсальной реляции над U будем сопоставлять гиперграф следующим образом:

- каждому атрибуту из U сопоставляется вершина гиперграфа, соответствующий универсальной реляции;

- каждому предикату в определении универсальной реляции над U сопоставляется ребро гиперграфа.

Для универсальной реляции в примере 1 определенной с помощью (1), получаем гиперграф:



с вершинами соответственно c,t,r,h,s,g и ребрами {c,t}, {c,h,r}, {c,s,g}. □

При задании универсальной реляции с помощью предикатов возможно использовать различные множества предикатов. Тогда видно, что одной универсальной реляции можно сопоставлять разные гиперграфы - где каждый гиперграф соответствует отдельного описания универсальной реляции с помощью предикатов.

Пусть H=(**E**,**N**) - гиперграф и пусть **E**={e_1,e_2,...,e_s}. Если ясно какие вершины участвуют в ребрах $e_1, e_2, \ldots, e_s$, для удобства иногда будем обозначать H через $H = \{e_1, e_2, \ldots, e_s\}$.

3. Попарное и сплошное совпадения

Дадим основные определения.

ОПРЕДЕЛЕНИЕ 2. [Mai83] Состояние базы данных над реляционными схемами $R_1, R_2, \ldots, R_n$ называется попарно совпадающим, если значения одинаковых атрибутов в отношениях над схемами $R_1, R_2, \ldots, R_n$ совпадают. □

ПРИМЕР 2. Пусть дана схема реляционной базы данных **R**={ABC,BCD,AD} и ее состояние

$r_1(ABC)$ =

A	B	C
0	0	0
1	1	1

, $r_2(BCD)$ =

B	C	D
0	0	0
1	1	1

, $r_3(AD)$ =

A	D
0	1
1	0

.

Покажем, что это состояние - попарно совпадающее.

Каждый атрибут для каждой схемы имеет значения {0,1}; поэтому значения одинаковых атрибутов в отношениях $r_1(ABC)$, $r_2(BCD)$ и $r_3(AD)$ совпадают.

Из определении 2 вытекает, что это состояние схемы **R**={ABC,BCD,AD} является попарно совпадающим. □

ОПРЕДЕЛЕНИЕ 3. [Mai83] Состояние базы данных над реляционными схемами $R_1, R_2, \ldots, R_n$ называется сплошь совпадающим, если все отношения этого состояния являются проекциями некоторой универсальной реляции с атрибутами $\cup_{i=1}^{n} R_i$. □

ПРИМЕР 3. Рассмотрим состояние из примера 2, отыскивая такое состояние универсальной реляции {ABCD}, что r_1, r_2 и r_3 являлись проекциями $r_1 = \pi_{ABC}(ABCD)$, $r_2 = \pi_{BCD}(ABCD)$, $r_3 = \pi_{AD}(ABCD)$.

Покажем, что $r_1(ABC)$, $r_2(BCD)$ и $r_3(AD)$ не являются сплошь совпадающими.

Отношение $r_1(ABC)$ содержит кортеж (0,0,0) и $r_2(BCD)$ содержит кортеж (0,0,0). Отсюда вытекает, что универсальная реляция ABCD должна содержать кортеж (0,0,0,0). Следовательно, если $r_1(ABC)$, $r_2(BCD)$ и $r_3(AD)$ являлись бы сплошь совпадающими, то отношение $r_3(AD)$ должно было содержать кортеж (0,0) - что противоречит содержанию $r_3(AD)$.

И так, показано, что $r_1(ABC)$, $r_2(BCD)$ и $r_3(AD)$ не являются сплошь совпадающими.

Рассмотрим другое состояние для схемы базы данных {ABC,BCD,AD}. Пусть

$r_1(ABC)$ =

A	B	C
0	0	0
1	1	1

, $r_2(BCD)$ =

B	C	D
0	0	0
1	1	1

, $r_3'(AD)$ =

A	D
0	0
1	1

.

Легко видеть, что $r_1(ABC)$, $r_2(BCD)$ и $r_3'(AD)$ являются сплошь совпадающими. Универсальная реляция ABCD, из которой получаются проекции $r_1(ABC)$, $r_2(BCD)$ и $r_3'(AD)$, имеет следующие кортежи:

r(ABCD) =

A	B	C	D
0	0	0	0
1	1	1	1

и тогда $r_1(ABC) = \pi_{ABC}(r)$, $r_2(BCD) = \pi_{BCD}(r)$, $r_3'(AD) = \pi_{AB}(r)$. □

Очевидно, сплошное совпадение имплицирует попарное совпаденое;

пример 3 показывает, что обратное неверно.

4. Алгоритм редукции гиперграфов

Рассмотрим следующий алгоритм, который называется алгоритмом редукции Грэема (Graham reduction algorithm) [Gra80].

АЛГОРИТМ 1. [FVa84] Алгоритм редукции гиперграфа данной схемы реляционной базы данных.

ВХОД: Гиперграф данной схемы реляционной базы данных записан в виде строк следующим образом:

каждому ребру гиперграфа отводится одна строка записи; при этом одинаковые атрибуты отдельных реляционных схем располагаются всегда один под другим.

ОПЕРАЦИЯ 1. Зачеркнуть имена всех атрибутов, которые появляются только один раз в входной записи;

ОПЕРАЦИЯ 2. Эсли какая-то строка с атрибутами $\{A_1, A_2, \ldots, A_K\}$ целиком содержит другую строку с атрибутами $\{A_{i1}, A_{i2}, \ldots, A_{ip}\}$, т.е. $\{A_{i1}, A_{i2}, \ldots, A_{ip}\} \subset \{A_1, A_2, \ldots, A_K\}$, зачеркнуть строку с атрибутами $\{A_{i1}, A_{i2}, \ldots, A_{ip}\}$.

МЕТОД: Применять операции 1 и 2 в произвольном порядке, сколько раз возможно.

ВЫХОД: Пустая запись - когда после применения операции 1 и 2 все символы входной записи зачеркнуты;

Непустая запись - когда после применения операции 1 и 2 не могут быть зачеркнуты все символы входной записи. □

Операции 1 и 2 алгоритма редукции не добавляют новые символы к входной записи, а только зачеркивают символы с входной записи. Имея ввиду конечности входной записи видно, что алгоритм 1 всегда заканчивает работу.

Легко видеть, что алгоритм 1 работает в полиномиальном периоде времени [FVa84].

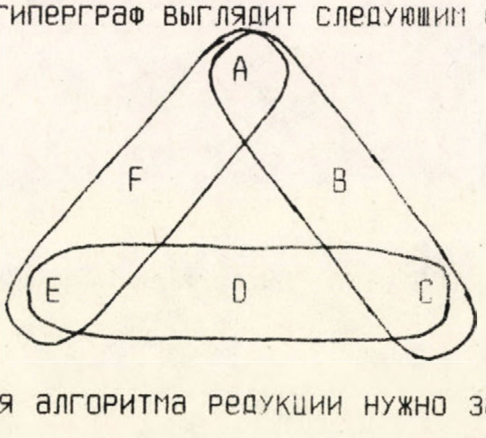
ТЕОРЕМА 1. [BFMY83] Гиперграф является ациклическим тогда и только тогда, когда алгоритм редукции заканчивает работу над этим гиперграфом с пустой записью. □

Теорема 1 позволяет нам ввести следующее определение:

ОПРЕДЕЛЕНИЕ 4. Пусть H - гиперграф, над которым алгоритм 1 заканчивает работу с пустой записью. Тогда H будет называть <u>ациклическим гиперграфом</u>. □

Проиллюстрируем алгоритм редукции на следующем примере.

ПРИМЕР 4. Пусть дана схема реляционной базы данных {ABC}, {CDE}, {AEF}. Ее гиперграф выглядит следующим образом:



(2)

Для применения алгоритма редукции нужно записать этот гиперграф как следует:

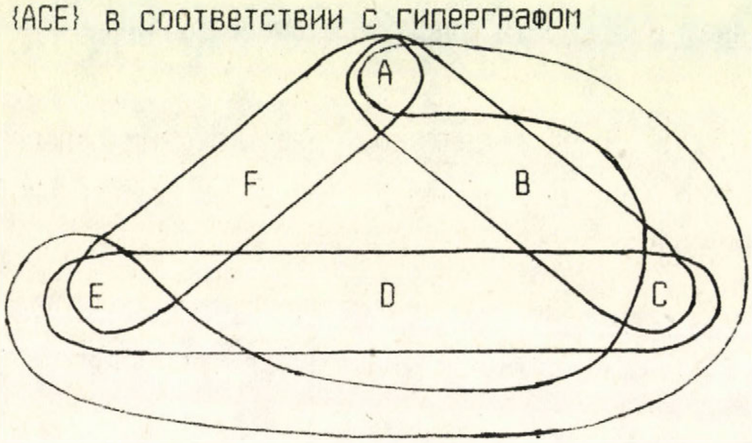
A	B	C			
		C	D	E	
A				E	F

Над этой записью применяем операцию 1 из алгоритма редукции и получаем

A	C	
	C	E
A		E

Эта запись является выходом алгоритма редукции, потому что дальнейшее применение как операции 1, так и операции 2 алгоритма 1 невозможно. Следовательно, гиперграф (2) является циклическим. □

ПРИМЕР 5. Пусть дана схема реляционной базы данных {ABC}, {CDE}, {AEF}, {ACE} в соответствии с гиперграфом



(3)

Покажем, что гиперграф (3) является ациклическим.

Гиперграф (3) можно представить в следующем виде:

A	B	C			
		C	D	E	
A				E	F
A		C		E	

(3′)

Применяем операцию 1 алгоритма редукции над записью (3′) и получаем

A	C	
	C	E
A		E
A	C	E

(4)

Применяем три раза операцию 2 из алгоритма 1 соответственно для строк

{A,C} ⊂ {A,C,E}

{C,E} ⊂ {A,C,E} ,

{A,E} ⊂ {A,C,E} ,

и получаем строку

A C E (4′)

Применяем операцию 1 из алгоритма 1 над записью (4′) и получаем пустую запись. Следовательно, гиперграф (3) является ациклическим. □

Отметим, что гиперграф (3), являясь ациклическим, содержит циклическое подмножество (2). Для обычных графов это не верно [Кри78].

Оказывается, что алгоритм редукции дает нам необходимое и достаточное условие для ответа на вопрос "Когда каждое состояние базы данных, которое является попарно совпадающим, является так же и сплошь совпадающим?". Следующие теоремы дают ответ на этот вопрос.

ТЕОРЕМА 2. [BFMY83] Пусть схема реляционной базы данных является ациклической. Каждое ее состояние является попарно совпадающим тогда и только тогда, когда это состояние - сплошь совпадающее. □

Следовательно, для ациклических схем баз данных, попарное совпадение эквивалентно сплошном совпадению.

ТЕОРЕМА 3. [BFMY83] Если схема реляционной базы данных является циклической, тогда существует ее состояние, которое является попарно совпадающим, но не является сплошь совпадающим. □

Примеры 2 и 3 показывают такое состояние для схемы {ABC},{BCD},{AD} (которая является циклической).

СЛЕДСТВИЕ 1. [BFMY83] Из теоремы 2 и 3 вытекает, что попарное совпадение эквивалентно сплошном совпадению тогда и только тогда, когда схема базы данных является ациклической. □

Формальное определение понятия ациклического гиперграфа дано в [BFMMUY81, BFMY83, Fa983] и тоже в [MaUl84].

5. Некоторые замечания о ацикличности

Докажем некоторые свойства ациклических схем.

Будем использовать запись гиперграфов в виде строк, которая представляет собой вход в алгоритме редукции.

ОПРЕДЕЛЕНИЕ 5. Изолированной вершиной гиперграфа H называется такая вершина, которая участвует только в одном ребре гиперграфа H. □

Изолированные вершины участвуют также только в одной строке записи гиперграфа H в виде строк и отстраняются от этой записи путем применения операции 1 из алгоритма редукции.

ОПРЕДЕЛЕНИЕ 6. Пусть $H = \{e_1, e_2, \ldots, e_p\}$ - гиперграф с ребрами соответственно $e_1, e_2, \ldots, e_p$. Тогда вершины, которые участвуют в хотя бы двух ребрах гиперграфа H, будем называть связанными вершинами. □

Если $H = \{e_1, e_2, \ldots, e_p\}$ - гиперграф, то через $e_1, e_2, \ldots, e_p$ будем обозначать и ребра гиперграфа H, и строки записи для гиперграфа H в виде строк. Будем обозначать множество всех связанных вершин в данной строке e_i через $\mathrm{con}(e_i)$. Следовательно, для каждой строки e (для каждого ребра e) гиперграфа H, мы разделяем вершины строки (ребра) в двух непересекающихся множествах: множество con(e) и множество изолированных вершин.

Отметим, что если запись $\{e_1, e_2, \ldots, e_s\}$ содержит строки, состоящие только из изолированных вершин, то тогда после применении операции 1 алгоритма редукции в полученной записи участвуют и пустие строки. Далее мы не будем отмечать существования пустых строк, если это не нужно.

ОПРЕДЕЛЕНИЕ 7. Будем говорить, что ребро $e_1 = \{A_1, A_2, \ldots, A_n\}$ гиперграфа H <u>поглощает</u> ребро $e_2 = \{A'_1, A'_2, \ldots, A'_k\}$ этого гиперграфа, если

$$con(\{A'_1, A'_2, \ldots, A'_k\}) \subset con(\{A_1, A_2, \ldots, A_n\}). \quad \Box$$

Будем обозначать этот факт через "$>$", т.е. $e_1 > e_2$.

Очевидно реляция "$>$" является реляцией частичной упорядоченности.

Пусть для двух строк $\{A_1, A_2, \ldots, A_n\}$ и $\{A'_1, A'_2, \ldots, A'_k\}$ выполнено $\{A_1, A_2, \ldots, A_n\} > \{A'_1, A'_2, \ldots, A'_k\}$. Тогда в процессе применения алгоритма редукции можно применять операцию 1 над изолированными вершинами в $\{A_1, A_2, \ldots, A_n\}$ и в $\{A'_1, A'_2, \ldots, A'_k\}$, а потом и операцию 2 над этими строками. При этом строка, содержащая вершин con $\{A'_1, A'_2, \ldots, A'_k\}$, исчезает с входной записи. Таким образом, каждая пара строк $e_1 > e_2$ может быть редуцирована до получения строки $con(e_1)$.

ПРИМЕР 6. Рассмотрим изолированные и связанные вершины гиперграфа (3) в примере 5.

Вершины B,D и F являются изолированными для этого гиперграфа. Все остальные вершины этого гиперграфа - т.е. вершины A,C и E, являются связанными. При этом выполнено: con({A,B,C})={A,C}, con({C,D,E})={C,E}, con({A,E,F})={A,E} и con({A,C,E})={A,C,E}.
□

Определения 5 и 6 введены для записи гиперграфа, которая является входом в алгоритм редукции (потому что ясно, что некоторые связанные вершини гиперграфа могут превратиться в изолированными после применения операции 2 алгоритма редукции гиперграфов). Обобщим определения 5 и 6 для каждого шага алгоритма 1.

ОПРЕДЕЛЕНИЕ 8. Пусть дана входная запись гиперграфа H соответственно со строками $\{e_1,e_2,\ldots,e_s\}$. Запись $\{e'_1,e'_2,\ldots,e'_k\}$ будем называть <u>редукцией</u> записи $\{e_1,e_2,\ldots,e_s\}$, если запись $\{e'_1,e'_2,\ldots,e'_k\}$ получена из записи $\{e_1,e_2,\ldots,e_s\}$ путем применения операции 1 возможное число раз и одного применения операции 2 алгоритма редукции гиперграфов. □

Пусть операция 2 алгоритма редукции применяется j раз над входной записю. Полученную записю будем называт редукцией входной записи на j-том уровне.

ОПРЕДЕЛЕНИЕ 9. Пусть дана запись гиперграфа $H=\{e_{i1},e_{i2},\ldots,e_{im}\}$, которая является редукцией входной записи на j-том уровне. Все вершины в записи $\{e_{i1},e_{i2},\ldots,e_{im}\}$, которые участвуют только в одном ребре множества $\{e_{i1},e_{i2},\ldots,e_{im}\}$, будем называть <u>изолированными вершинами (на j-том уровне редукции)</u>; остальные вершины будем называть <u>связанными вершинами (на j-том уровне редукции)</u>. □

ПРИМЕР 7. Рассмотрим изолированные и связанные вершины на разных уровнях редукции для гиперграфа (4).

Рассмотрим запись (4) из примера 5:

A C

C E

A E

A C E

На первом уровне редукции этой записи можно получить запись

C E

A E

A C E

В этой записи все вершины связанны на первом уровне редукции.

На втором уровне редукции можно получит запись

A E (5)

A C E

В этой записи вершина C является изолированной на втором уровне редукции, а вершины A и E являются связанными на втором уровне редукции. Применяя еще раз операцию 1 над вершиной C в (5) и операции 2 над вершинами {A,E}, получаем

A E

В этой записи все вершины изолированы на третом уровне редукции. □

Будем говорить об изолированных и связанных вершинах, подразумевая, где это возможно, соответствующий уровень редукции.

Легко видеть, что если данная схема базы данных является ациклической, то выполнено одно из следующих утверждений:

а) выход алгоритма редукции (т.е. пустая запись) получается путем применения только операции 1 алгоритма редукции. В этом случае все вершины гиперграфа (все атрибуты схемы) являются изолированными и никакой атрибут не участвует хотя бы два раза в отдельных реляционных схемах, т.е. для всех ребер e гиперграфа H, $con(e)=\varnothing$. Такие базы данных содержат только семантически несвязанные данные [Ап981], сгруппированные в отдельные несвязанные отношения, не имеющие общих атрибутов. Таких схем баз данных мы рассматривать не будем.

б) выход алгоритма редукции получается путем применения как операцию 1 алгоритма редукции, так и операцию 2 этого алгоритма. В этом случае схема базы данных содержит реляционные схемы, которые имеют общие атрибуты между собой и поэтому являются семантически связанными [Ап981].

ОПРЕДЕЛЕНИЕ 2.10. Пусть $H=\{e_1,e_2,\ldots,e_p\}$ - данный гиперграф. H будем называт <u>связанным гиперграфом</u>, если для каждой пары ребер (e_i,e_j), $1\le i\le p$, $1\le j\le p$, $i\ne j$, существует цепочка различных ребер

$e_i=e'_1,e'_2,\ldots,e'_s=e_j$ такие, что для $1\le r\le s$

1) $e'_r\ne e_i$, если $r\ne 1$;

2) $e'_r\ne e_j$, если $r\ne s$;

3) $e'_r\cap e'_{r+1}\ne\varnothing$ если $r\ne s$;

4) $e'_r\in H$. □

ЛЕММА 1. Если H - связанный гиперграф, то на всех уровнях алгоритма редукции над H получается запись, соответствующая некоторому связанному гиперграфу.

ДОКАЗАТЕЛЬСТВО. Пусть n - соответствующий уровень редукции. Лемму докажем с помощью индукции по отношению к n.

n=1. Покажем, что полученная запись на первом уровне редукции соответствует некоторому связанному гиперграфу H'. Пусть

$H=\{e_1,e_2,\ldots,e_p\}$, а $H'=\{e'_1,e'_2,\ldots,e'_p\}$. Докажем, что H' - связанный гиперграф. На первом уровне редукции имеем $e'_i=\mathrm{con}(e_i)$, $1\leq i\leq p-1$. Строка $e_p\in H$ зачеркнута с помощью операции 2 алгоритма редукции - она поглощена некоторой строкой $e_r\in H$.

Пусть e'_i,e'_j - ребра из H'. Покажем, что они связанны.

Пусть $e_i\in H$, $e_j\in H$ связанны при помощи цепочки

$$e_i=e_{i1},e_{i2},\ldots,e_{is}=e_j.$$

Если $e_{ik}\neq e_p$ для $k=1,2,\ldots,s$, то тогда e'_i и e'_j связанны при помощи цепочки

$$e'_i=e'_{i1},e'_{i2},\ldots,e'_{is}=e'_j.$$

Если $e_{ik}=e_p$ для $k=m$, то тогда e'_i и e'_j связанны при помощи цепочки

$$e'_i=e'_{i1},\ldots,e'_{im-1},e'_r,e'_{im+1},\ldots,e'_{is}=e'_j.$$

Следовательно, полученная на первом уровне редукции запись соответствует некоторому связанному гиперграфу.

Допустим, что на n-том уравне редукции полученная запись соответствует некоторому связанному гиперграфу.

Видно, что на $n+1$-ом уровне редукции (анналогично, как на первом уравне редукции) получается запись соответствующая некоторому связанному гиперграфу. □

ЛЕММА 2. Гиперграф H является ациклическим тогда и только тогда, когда на всех уровнях редукции гиперграфа H получается запись, соответствующая некоторому ациклическому гиперграфу.

ДОКАЗАТЕЛЬСТВО. Следует непосредственно из теоремы 1 и из определения 4. □

Отметим, что в процессе редукции ациклического гиперграфа пустую запись можно получить только с помощью операции 1. Последная строка, над которой в этом случае применяется операция 1, поглотила все остальные строки на последном уровне редукции данного гиперграфа.

ПРИМЕР 8. Покажем, что поглощающая строка на последном уровне редукции ациклического гиперграфа зависит от порядка применения операции 2 алгоритма редукции.

Рассмотрим гиперграф

A	B		
	B	C	
		C	D

Если применить операцик 2 над парой {A,B} и {B,C}, где con{A,B}⊂con{B,C}, получим в редуцированной на первом уровне записи строку {B,C}; затем если применить операцию 2 над парой {B,C} и {C,D}, где con({B,C})⊂con({C,D}), то тогда поглощающая строка этого гиперграфа будет {C,D}. Но если применить операцию 2 на первом уровне редукции над парой con{C,D}⊂con{B,C}, то тогда поглощающая строка на втором уровне редукции может быть {B,C}. □

Лемма 2 и пример 8 показывают, что поглощающая строка всегда существует для ациклических гиперграфов, но какая она конкретно - это зависит от выбора порядка применения операцию 2 алгоритма редукции гиперграфов.

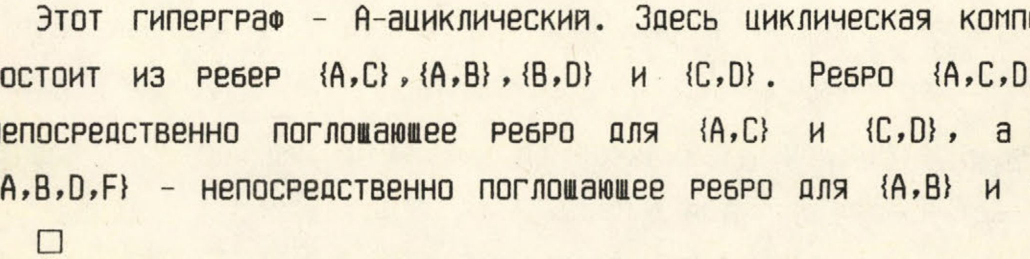
ОПРЕДЕЛЕНИЕ 11. [Fa983a] Гиперграф H называется <u>A-ациклическим</u>, если алгоритм редукции закончивает работу с пустой записю над этим гиперграфом. □

ОПРЕДЕЛЕНИЕ 12. Пусть $H=\{e_1,e_2,\dots,e_s\}$ - A-ациклический гиперграф. Циклической компонентой в H будем называть такую совокупность ребер H′, H′⊂H, для которой алгоритм редукции не заканчивает работу с пустой записью над входом H′. □

Как уже видно из леммы 2, для каждого A-ациклического гиперграфа H существует ребро, которое поглощает все остальные ребра при применении алгоритма редукции над H.

ОПРЕДЕЛЕНИЕ 13. Пусть $H=\{e_1,e_2,\dots,e_s\}$ - гиперграф. Пусть его ребро e_i поглощает ребро e_j путем одного применения операцию 2 алгоритма редукции. Тогда e_i будем называть непосредстенно поглощающим для ребра e_j. □

ПРИМЕР 9. Лемма 2.2 показывает, что для циклических компонентов в A-ациклическом гиперграфе всегда существует поглощающее ребро на некотором уровне редукции. Но как видно из следующей схемы базы данных, непосредственное поглощающее ребро может не быть единственным для циклических компонентов в A-ациклических гиперграфах.



Этот гиперграф - A-ациклический. Здесь циклическая компонента состоит из ребер {A,C}, {A,B}, {B,D} и {C,D}. Ребро {A,C,D,E} - непосредственно поглощающее ребро для {A,C} и {C,D}, а ребро {A,B,D,F} - непосредственно поглощающее ребро для {A,B} и {B,D}.
□

ОПРЕДЕЛЕНИЕ 14. [Fa983a] Пусть H - A-ациклический гиперграф, который не содержит циклических компонент. Тогда будем называть H В-ациклическим гиперграфом. Все гиперграфы, которые не являются В-ациклическими, будем называть В-циклическими. □

ТЕОРЕМА 4. [Fa983a] Гиперграф H является В-циклическим тогда и только тогда, когда в нем существует цепочка

$(S_1, x_1, S_2, x_2, \ldots, S_m, x_m, S_{m+1})$

такая, что

1) $x_1, x_2, \ldots, x_m$ - разные вершины гиперграфа;

2) $S_1, S_2, \ldots, S_m$ - разные ребра гиперграфа, а $S_{m+1}=S_1$;

3) $m \geq 3$, т.е. цепочка содержит по крайней мере три ребра;

4) $x_i \in S_i \cap S_{i+1}$ $(1 \leq i \leq m)$ и никакому другому S_j. □

ЛЕММА 3. Алгоритм редукции заканчивает работу с пустой записью над каждым подмножеством данного гиперграфа H тогда и только тогда, когда H является В-ациклическим.

ДОКАЗАТЕЛЬСТВО. а) Достаточность. Пусть H является В-ациклическим, т.е. по теореме 4 в нем нет цепочки циклически связанных ребер. Следовательно, в каждом H',$H' \subset H$, тоже нет такой цепочки и по теореме 4 каждое множество H' является тоже В-ациклическим. Тогда алгоритм редукции заканчивает работу с пустой записью над H'.

б) Необходимость. Пусть алгоритм 1 заканчивает работу с пустой записью над каждым подмножеством данного гиперграфа H. Допустим, что H не является В-ациклическим. Тогда по теореме 4 в нем существует цепочка $(S_1, x_1, S_2, x_2, \ldots, S_m, x_m, S_{m+1})$

Покажем, что над этой цепочкой алгоритм редукции не может заканчивает работу с пустой записью.

Применяем операцию 1 над изолиранными вершинами из $S_1, S_2, \ldots, S_m$.

В полученную запись участвуют m строки:

$con(S_1) = X_m \cup X_1$

$con(S_2) = X_1 \cup X_2$

- - -

$con(S_m) = X_{m-1} \cup X_m$

Так как $X_i \neq X_j$ при $i \neq j$, $1 \leq i \leq m$, $1 \leq j \leq m$, то нельзя применить операцию 2 алгоритма редукции над полученной записю. Видно, что невозможно применять и операцию 1 алгоритма редукции - так как полученная запись состоит только из связанных вершин.

Следовательно, над цепочки $(S_1, x_1, S_2, x_2, \ldots, S_m, x_m, S_{m+1})$ алгоритм редукции не может закончить работу с пустой записю, что является противоречием. □

6. Заключение

В этой работе рассмотрено понятие ациклической и циклической схемы реляционной базы данных. Введено понятие поглощающей строки и показано, что для ациклических гиперграфов всегда существуют поглощающие строки. Они определяются выбором порядка операции из алгоритма редукции гиперграфов.

Реляционные базы данных с ациклическими гиперграфами представляют собой важный класс баз данных, потому что для них можно легко решать некоторые вопросы сохранения и обработки информации (см. [BFMU83], [Fag83], [Fag83a], [Yan81]). В [MaUl84] показано, что с помощью ациклических гиперграфов можно искать эффективные алгоритми оптимизации некоторых запросов в реляционных базах данных. В заключении можно сказать, что интерпретация схемы реляционной базы данных в качестве гиперграфа позволила исследовать разных связей между атрибутами универсума и таким образом ввести впервые классификацию схем реляционных баз данных.

Л И Т Е Р А Т У Р А

[Ang81] Angelova, G. The Use of Natural Language as a Query Language in a Relational Data Base. Proceedings of the Fourth Int. Seminar on Data Base Management Systems, Schwerin, GDR, December 1981, pp.226-238.

[AnZ85] Angelov, Zh. Towards a Universal Relation View. Proceedings of the Eigth International Seminar on Data Base Management Systems, Piestany, CSSR, September 1985, pp. 9-17.

[BFMY83] Beeri,C., Fagin,R., Maier,D. and Yannakakis, M. On the Desirability of Acyclic Database Schemes. Journal of the ACM, Vol. 30, No.3, July 1983, pp.479-513.

[BFMMUY81] Beeri, C., Fagin, R., Maier, D., Mendelzon, A., Ullman, J. and Yannakakis, M. Properties of Acyclic Database Schemes. Proceedings of the Thirteenth Annual ACM Symposium on the Theory of Computing (1981), pp. 355-362.

[Cod70] Codd, E.F. A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, Vol. 13, No. 6, June 1970, pp.377-387.

[Cod72] Codd, E.F. Further Normalization of the Data Base Relational Model. In Data Base Systems, Courant Computer Science Symposia, Vol. 6, Prentice-Hall, Englewood Cliffs, N.J., 1972, pp.33-64.

[FMU82] Fagin, R., Mendelzon, A. and Ullman, J. A Symplified Universal Relation Assumption and its Properties. ACM Transactions on Data Base Systems, Vol.7, No.3, September 1982, pp.nnn-mmm.

[Fag83] Fagin, R. Degrees of Acyclicity for Hypergraphs and Relational Data Base Schemes. Journal of the ACM, Vol. 30, No.3, July 1983, pp.514-550.

[Fag83a] Fagin, R. Acyclic Data Base Schemes of Various Degrees - a Painless Introduction. Lecture Notes in Computer Science, G. Ausiello and M. Protassi (eds.), Vol.159, 1983, pp.65-89.

[FVa84] Fagin, R. and Vardi, M. The Theory of Data Dependencies - a Survey. IBM Research Report No. 4321, 1984.

[Gra80] Graham, M. On the Universal Relation. In: A Panache of DBMS Ideas III, D. Tsichritzis (ed.), Technical Report CSRG-111, April, 1980, University of Toronto.

[Mai83] Maier, D. The Theory of Relational Databases. Computer Science Press, Rockville, Maryland, 1983.

[MUV84] Maier, D., Ullman, J. and Vardi, M. On the Foundations of the Universal Relation Model. ACM Transactions on Data Base Systems, Vol.9, No.2, June 1984, pp.283-308.

[MaUl84] Maier, D. and Ullman, J. Connections in acyclic hypergraphs. Theoretical Computer Science 32(1984), North-Holland, pp.185-199.

[Ull82] Ullman, J. Principles of Database Systems. Computer Science Press, 1982.

[Yan81] Yannakakis, M. Algorithms for Acyclic Database Schemes. Proceedings 1981 Very Large Data Bases Conference, pp. 82-94.

[Дри82] Дрибас, В.П. Реляционные модели баз данных. Минск, Издательство БГУ им. В.И.Ленина, 1982.

[Кри78] Кристофидес, Н. Теория графов. Москва, "Мир", 1978.

Cyclic and acyclic relational database schemes

G. Angelova

Summary

The paper discusses the acyclicity (and cyclicity) of relational database schemes. The concepts of pairwise consistency and join consistency are described. The Graham reduction algorithm for determining acyclicity is presented. The concept of so called "absorbent" hyperedge is introduced. It is shown that acyclic hypergraphs always contain at least one absorbent hyperedge.

Ciklikus és aciklikus relációs adat-bázis sémák

G. Angelova

Összefoglaló

A cikk a relációs adatbázis sémák aciklikusságát /ill. ciklikusságát/ vizsgálja. Ismerteti a páronkénti ill. együttes konzisztencia fogalmát és az aciklikusság megállapitására szolgáló Graham-féle redukciós algoritmust. Bevezeti az u.n. "elnyelő" /"absorbent"/ hiper-él fogalmát és megmutatja, hogy az aciklikus hiper-gráfok tartalmaznak "elnyelő" hiper-élt.

MTA SZTAKI Közlemények 36/1987 pp.31-47

ABOUT A METHODOLOGY TO SELECT A DBMS

M.F. ATAN

M. E. BRAGADO BRETANA

Institute Central de Investigacion Digital
Calle 198 1703
Siboney, C. de la Habana, Cuba

1. Introduction.

The rising use of database systems for the data management has resulted in an increasing number of systems entering the marketplace. The selection of a database system requires a structured, comprehensive investigation.

The following paper show a general methodology to select a DBMS in order to use it and also to select it as a pattern to be implemented in some hardware configuration.

2. Aims.

A complete evaluation methodology for database systems must integrate a feature analysis phase, human factors aspects /CNORT83/, and a performance analysis phase.

The objective of this evaluation is not only to choose a system for an application, but to take a DBMS as a pattern to be implemented. Because of this, several features of the systems do not have great importance; for example, the Operating System on which the system executes, arithmetic precision, error recovery, etc.; these are implementation characteristics that can be adapted to each necessities. Therefore, it complements the methodology with a phase about the characteristics of implementation of each system.

The main objetive in our methodology is to remark the integrity of these phases. Any phase itself can not be used in isolated form to determine which system must be selected. Each phase must be analyzed in complementary form to obtain a successful selection.

The figure 2.1 shows a summary of our methodology.

```
                        -----------------
                       |                 |
                       |   METHODOLOGY   |
                       |                 |
                        -----------------
                   /         |        \    \
                  /          |         \    \
                 /           |          \...\
 -------------------   ----------------  :  ---------------------
| Feature analysis |  | Human Factors |  :  | Performance anal. |
 -------------------   ----------------  :  ---------------------
        |                    |           :    |
        |                    |           :    |     ---------------
        |                    |           :___|___| Feat. of the  |
        |                    |                |   | implementat. |
        |                    |                |    ---------------
-Syst. features        *Syntax features       |          |
-Data base creation    *User data names       |          |
-Report Generation     *Error detect.         |          |
-Data retrieval         and recovery          |          |
-D. b. modification    *Menu interfaces       |          |
-Data editing                                 |          |
-Data concurrency                             |          |
-Creation/edition                 >Benchmark design      |
 of programs                        -the scope           |
-File compatibility                 -the tests           |
-Documentation                      -the environment     |
                                  >Benchmark execution   |
                                  >Benchmark analysis    |
                                                         |
                                                         |
                                          *Language analysis
                                          *Creation/edition
                                            of programs
                                          *Management   of
                                            data dictionary
                                          *File   control
```

Figure 2.1. A summary of our methodology.

2.1. Feature analysis phase.

The features and capabilities that a database system may provide is very large. A feature analysis fulfill two functions; it first serves as a process to decide those systems that are completely unsuitable for answering the needs of a particular application and second, it provides a ranking of the surviving candidate systems.

Feature analysis has a number of advantages over other methods of system evaluation.

i) Feature analysis provides a structured first cut. The final result of a feature analysis should be a small number of candidate systems. Performance analysis, which is much more costly, can then be performed with only this small number of systems.

ii)There are qualitative aspects of a database system that cannot be quantified in terms of system performance; for example: vendor support, documentation quality, security, user friendliness, etc. Since benchmark analysis cannot directly test the performance of these features, feature analysis remains the best method for their analysis.

iii) Little or no system costs are involved in performing a feature analysis because a database implementation is not required.

In spite of these advantages features analysis should not be used in isolation to evaluate and select database systems. There are several reasons for this.

i) The feature importance coefficients and the system support ratings are given values by a knowledgeable design expert. However, no two experts may come up with same values given the same application environment, because the feature analysis is a subjective exercise.

ii) Feature analysis is a paper exercise that cannot truly evaluate how a system will perform in an organization's application environment.

The following are the points included in the feature analysis for each system.

1) System characteristics. It is included the general features of the system: the type of DBMS (relational, CODASYL, etc), Operating System, main and auxiliary memory necessary, type of file organisation, characteristics of the arithmetic operators, etc.

2) Data base creation. It is described the means for the data base creation.

3) Report generation. It is shown the commands or means to obtain a report of a file.

4) Data retrieval. It is included the analysis of tools for data retrival, the commands which allow to know the structure of a file, etc.

5) Data base modification. In any application it is necessary to modificate the database. Here, it is described the several means which allow to asimilate the modifications and changes in a database.

6) Data editing. To study the tools to data edition: edition and modification fields of a file.

7) Data concurrency. To analyze the several form of concurrency control included in the DBMS evaluated.

8) Creation and edition of programs.

9) File compatibility. The compatibility of files between different systems is a good characteristic which allows to transfer files and to make applications in connection with different systems.

10) Documentation. It is described the quality and characteristics of all the documentation of the system.

2.2 Human factors aspects.

Relational technology was provided to a new class of users through simplified terminology and a relational algebraic command language. These new users knew their application areas well, but their main tasks were nonprogramming tasks. To the correct evaluation of a DBMS it is necessary to make a heavy analysis of the human factors or psychologic aspects in order to accept a system. A few authors include these factors in the general evaluation of the systems.

The following phase highlights the human factors aspects, the benefits, and the limitations of each system evaluated.

i) Syntax features of the commands.

In this point it is evaluated the syntax features of the names of the commands: friendly language, relation between the name of the command and the corresponding data base operation, and the full command names and keywords without abbreviations.

The use of a language close to the natural is very important to the assimilation and learning of a system. The user does not feel the difference between the way usually he thinks and the way he works with the computer. This is important to decrease the debbuging time of applications. The use of "noise" words helps to improve the readability of a command. Also, these characteristics improve the self documentation of programs.

ii) User data names.

Here, it is analyzed the possibilities that the system provides in order to express the names of the user data in legible form. Also, this aspect has influence in the keyboard errors. From some observations, users desire conciseness, but this is overshadowed by the need to express and document ideas in meaningful phrases. Users frequently try to condense abbreviations or use meaningless names such as X or ABC that make errors typing them or cannot remember the precise names that were used. A good system must allow that syntax of data names be legible.

iii) Error detection and recovery.

The time lost when errors are not handled properly for the user indicate the importance of good error handling. Here, it is evaluated if the systems have a good error detection, recovery, and informative messages.

iv) Menu interfaces.

In the same form that increases the interactive way of work with the computer, it increases the use of menu interfaces between the man and the computer. The systems driven by menu are very easy to use. With the combination of menus defined with meaningful English phrases and availability of "help" messages, users have not much trouble, becoming effective users. Users do not have to learn or remember commands; they simply make choices from a menu. This is an important human aspect for the easy assimilation of a system, specially for the non-specialized user.

v) Learning.

It means how long it takes the user to learn how to work with a system. This is a very important human factor to accept a system by the user. A very efficient system but with difficulty when it shows the form of use, will be difficult to be accepted for the common user.

This aspect must be fulfill with some kind of statistical investigation between several user with no equal levels of technical experiences.

2.3. Performance analysis phase.

The major methods of performance evaluation are Analytic modelling, Simulation modelling, and Benchmarking.

Analytic modelling represents a system by defining equations that relate performance quantities to known system parameters. The use of these equations allows a fast and accurate means to evaluate system performance. The principal disadvantages are that the equations are inadequate to model the complete range of functionality found in a data base system and also they fail to account for the dynamic behavior of the data base system. For these reasons analytic modelling has failed to receive wide acceptance as a tool for modelling data base systems.

Simulation is the process of developing a computer program to approximate the behavior of a system over a period of time. Simulation modelling has been applied to data base systems /HULTE77/, /NAKAM75/. The major concern with using simulation is the time and expense that are often necessary to develop a simulation model. Stochastic simulation models also produce only estimates of a model's true performance and the large volume of results returned by a simulation often creates a tendency to place more confidence in the results than may actually be warranted.

Benchmarking is used when a few data base systems are to be evaluated and compared. Benchmarking requires that the systems be implemented so that experiments can be run under similar system environments. Benchmarks are costly and time-consuming but provide the most valid performance results upon which data base systems can be evaluated. While both simulation and analytic modelling are limited in the scope of their system testing, benchmarking offers the chance to evaluate the actual data base system /GOFF73/.

The benchmark experiments publicated concentrate on the comparison of candidate commercial systems for a particular application /GLESE81/, /ASTRA80/, /KEENA81/, /TEMPL/, etc.

While benchmarking can be a useful and important technique for data base system evaluation; designing, setting up, and running a benchmark is a difficult and time-consuming task. Benchmarking is problematic and at worst, a gross distortion of reality but it is possible to obtain good conclusions if these aspects are known and if specific features are analyzed.

In order to aid in the development and analysis of benchmarks it is essential to show the methodology used. No one methodology has provided the necessary robustness demanded from a generalized methodology. No benchmark methodology can expect to incorporate every aspect of every benchmark.

Our methodology has been divided into 3 principal parts: benchmark design, benchmark execution and benchmark analysis.

2.3.1. Benchmark design.

The design of a benchmark involves: a) the scope of the tests, b) the tests to be performed, and c) the environment of the data base system to be tested.

As it was shown above, the success of benchmarks depends on the objectives be exactly detailed. It has been proved that general benchmarks distort the results and mask the deficiencies /HOUST84/. In our case, it uses the benchmark to complement the other phases that are included in our evaluation. Besides, the current systems allow to perform several classic processes of DBMS in interactive way as: creation and modification of data bases, edition of programs and data bases, report generation, etc., which are not possible to apply to any classic benchmark test. These features are included into another phase of our general evaluation.

2.3.2. Benchmarks execution.

When the experiment has been formally defined, the next step is to implement the design for each of the candidate systems.

2.3.3. Benchmark analysis.

The final phase of benchmarking is the analysis of results. Evaluation of the data generated during benchmarking must begin before the tests have been completed. It provides feedback during the testing by suggesting which types of experiments need to be repeated in more detail, or should be extended in some way. Summarizing the meaningful information from these results and discussing them in a report form is a key step in the benchmark testing.

2.4 Characteristic of the implementations.

As it has shown above, the objective of our methodology is to select a system to implement it in some configuration. Therefore, it is necessary that our methodology contains a phase about the difficulty to implement one or another DBMS.

In this phase the following modules will be analyzed: language, creation and edition of programs, management of data dictionary, and file control system. Each modules must be analyzed from the point of view of the difficulties of implementation.

i) Language.

The characteristics of implementation of the languages must be analyzed. It is necessary to analyze the structure of the language, characteristic of the syntax and semantic analysis, type of compiler (interpreter, compiler, etc), language ambiguity, inter-relation between the language and other modules, etc.

ii) Creation and edition of programs.

The creation and edition of programs are the means included in the DBMS to develop programs. It is necessary to analyze:

-level of full-screen editing of command files

-special features

-relation with other modules of the system

-if it has included some function of syntax analysis it is important to evaluate the level of relation with other modules of the system.

iii) Management of data dictionary.

This module includes all the software means necessary to control the operations with data dictionary. Here, it is included the analysis of the following aspects:

-structure of the data dictionary

-the means to create/maintenance of data dictionary

-the software to handle the dictionary

-level of complexity of the data dictionary

iv) File control system.

The difficulty of programming the file organization and its commands are shown in this section. The analysis must include the following:

-features of the organization used (indexed, sequential, etc)

-characteristics of the commands that have relation with the files

-characteristics of the data protection.

3. Conclusions.

It is shown an integrated methodology to select a DBMS in order to take it as a pattern to be implemented. This methodology must be used as a whole and complemented with particular analysis. The conclusion of the evaluation process must be shown with several summary tables which explain the result in each phase. This methodology is used in our Institute with succesful results.

4. Bibliography.

/ASTRA80/ Astrahan, M. et al. Performance of the System R Access Path Selection Mechanism. Proc. IFIP, 1980

/BENWE75/ Benwell, N. Benchmarking. Toronto, J. Wiley and Sons, 1975.

/BING85/ Bing, Yao S., and A. Hevner. Performance Evaluation of Database Systems: a Benchmark Methodology. c N.J. USA Datapro Research Corp., AS80-100-101, Aug. 1985.

/BITTO83/ Bitton, H. et al. Benchmarking Database Systems: A Systematic Approach. Computer Sciences Department, University of Wisconsin, Technical Report #526.

/BOAR84/ Boar, B. H. Ten Criteria For Selecting Mature DBMSs. c Auerbach Publishers Inc, Data Base Management, 1984.

/BOGDA83/ Bogdanowicz, R. et al. Experiments in Benchmarking Relational Database Machines. Munich, Proc. of the third International Workshop on Database Machines, Sept. 1983.

/BOIES74/ Boies, S.J. User Behavior on an Interactive System. IBM System Journal 13,1-18 1974.

/BOND84/ Bond, G. A Database Catalog. BYTE, Oct. 1984.

/BORAL84/ Boral, H. and D. DeWitt. A Methodology for Database System Performance Evaluation. Univ. of Wisconsin, Computer Sciences Department, Technical Report #532, January 1984.

/BOYLE84/ Boyle, B. Software Performance Evaluation. BYTE, 9(2): Feb. 1984.

/CARRO84/ Carrol, J.M., M.B. Rosson Beyond MIPS. Performance Is Not Quality. BYTE, 9(2): Feb. 1984.

/CNORT83/ c North-Holland Pub. Company. Human Factors Aspects of a Modern Data Base System. Information and Management, 6(1): Feb. 1983.

/CURNO76/ Curnow, H. J., B. A. Wichman A Synthetic Benchmark. Computer Journal, 19(1): Feb. 1976.

/DAVIE81/ Davies, D.J.M. Benchmarking in Selection of Timesharing Systems. Proceedings of the 14th. meeting of the CPEUG, Nov. 1981.

/DEARN78/ Dearnley, P. Monitoring Database System Performance.

The Computer Journal, 21(1): 1978.

/DEMET84/ Demetrovics, J., et al Some Remarks on Statical Data Processing. Hungary, MTA SZTAKI Kozlemenyek, 30: 37-51 1984.

/FERRE78/ Ferreri D. Computers Systems Performance Evaluation. Prentice-Hall, Inc., 1978.

/GILBR81/ Gilbreath, J. A High-Level Language Benchmark. BYTE, 180-198 Sept. 1981.

/GLESE81/ Gleser, M. et al. Benchmarking for the Best. Datamation, May 1981.

/GOFF73/ Goff, N.S. The case for Benchmarking. Computer and Automation, May 1973.

/HOUST84/ Houston, J. Don't Bench me in. BYTE, 9(2): Feb. 1984.

/HULTE77/ Hulten, C. and L. Soderlund. A simulation Model for Performance Analysis of Large Shared Data Bases. Proc. Third VLDB Conf., Tokio, 1977.

/KEENA81/ Keenan, M. A Comparative Performance Evaluation of Database Management Systems. Berkeley, EECS Dept., University of California, 1981.

/MARVI84/ Marvit, P., M. Nair. Benchmark Confessions. BYTE, 9(2): Feb. 1984

/NAKAM75/ Nakamura, F., et al. A Simulation for Data Base System Performance Evaluation. Procc. NCC, 1975.

/REISN81/ Reisner, P. Human Factors Studies of Database Query Languages: a Survey and Assesment. ACM Computing Surveys, 13(1):

13-32, 1981.

/ROBER84/ Roberts, B. Benchmarks and Performance Evaluation. BYTE, 9(2): Feb. 1984.

/RODRI75/ Rodriguez-Rosell, J. and D. Hilderbrand. A Framework for Evaluation of Data Base Systems. Research Report RJ 1587, IBM, San Jose, 1975.

/SIBLE84/ Sibley, E. H. DBMS Evaluation and Selection. c Auerbach Publishers Inc, Data Base Management,22-04-01.

/SPOON/ Spooner, C.R. Benchmarking Interactive Systems: Modeling the Application. Proceedings of the 15th. Meeting of the Computer Performance Evaluation Users Group (CPEUG), pp.53-63.

/SU81a/ Su, S. et al. A DMS Cost/Benefit Decision Model: Cost and Preference Parameters. National Bureau of Standards,Report NBS- GCR-82-373,1981.

/SU81b/ Su, S. et al. A DMS Cost/Benefit Decision Model: Analysis, Comparison, and Selection of DBMSs. National Bureau of Standards,Report NBS-GCR-82-375,1981.

/TEMPL/ Templeton, M. et al. Evaluation of 10 Data Management Systems. SDC document TM-7817/000/00.

/WALTE76/ Walters, R.E. Benchmark Techniques: A Constructive Approach. The Computer Journal, 19(1): Feb. 1976.

/WEISS81b/ Weiss, H. Which DBMS is Right for You. Mini-Micro Systems, October 1981.

Одна методолония для выборки DBMS

М. Фонфрия Атан, М.Е. Брагадо Бретана

Резюме

Уже существует много различных пакетов для разработки базы данных. Для того чтобы выбрать один из них, надо применить очень много тщательных и структурированных методов. В статье показана общая методология такой выборки.

A D B M S KIVÁLASZTÁSÁNAK EGY MÓDSZERTANA

M. Fonfria Atan, M.E. Bragado Bretana

Összefoglaló

A piacon már rengeteg különböző adatbázis-kezelő programcsomag létezik. Egynek a kiválasztásához igen sok, alapos és részletes vizsgálat szükséges. A cikk a kiválasztásnak egy lehetséges metodológiáját irja le.

MTA SZTAKI Közlemények 36/1987 pp. 49-78

The Equivalence Problem For Relational Database Schemes

Joachim Biskup and Uwe Räsch
Institut für Informatik
Hochschule Hildesheim
Samelsonplatz 1
D-3200 Hildesheim
Federal Republic of Germany

Abstract

Mappings between the sets of instances of database schemes are used to define different degrees of equivalence. The available class of mappings and the set of dependencies allowed for defining schemes deal here as parameters. A comparison of the equivalences shows that there is only one natural kind of equivalence. For various cases we prove its decidability or undecidability. Besides we get a characterization of mappings expressible in the relational algebra without the difference.

1. Introduction and Conventions

An intuitive definition of equivalence is given by [Gee]: "Two databases are equivalent if they represent the same set of facts about a certain piece of world". We will try to get a more exact definition of what is meant by equivalence. We will consider only relational databases.

The motivation for a comparison of the information capacity of database schemes stems from different areas:

- design of conceptual schemes, especially the so-called database normalization
- integration of different userviews into a single global scheme
- translations between different databases
- extensions and transformations of databases
- evaluation of different approaches in database theory.

We will develop a general model (chapter 1) to formalize some kinds of equivalence and to study differences between them (chapter 2). Then we are concerned with the decision problem for the chosen kind of equivalence. In chapter 3 we present some decidable cases, whereas in chapter 4 we prove various undecidability results.

We will neither assume a universal scheme assumption or a universal relation assumption (see [AP]), nor consider the update facilities used in a database (see [Codd] for update-equivalence).

More powerful mapping classes are NGEN* and FGEN*. They are related to the " M-internal mappings " of [Hull] and defined as follows:

q1 ∈ NGEN* iff SYMB(q1(A1)) ⊂ SYMB(A1) for all A1 ∈ TYPES1
(" no generation of new values "),

q1 ∈ FGEN* iff there exists a finite set M such that
SYMB(q1(A1)) - SYMB(A1) ⊂ M for all A1 ∈ TYPES1
(" generation of only a finite set of new values ").

Every mapping class Q is assumed to contain the identity w.r.t. the set of all states of an arbitrary database scheme and to be closed under composition, that means:

∀ q1: TYPE1 -> TYPE2 ∀ q2: TYPE2 -> TYPE3:
q1 ∈ Q and q2 ∈ Q ==> q2 o q1 ∈ Q.

These assumptions are obviously satisfied by RALG*, RANP*, RAND*, NGEN* and FGEN*.

We will also define two classes of database dependencies.

ALL denotes the set of dependencies which can be transformed into a sentence in prenex-normalform without existential quantifiers. Often the adjective "full" is used to characterize some subsets of ALL, see [FV] or [CLM]. Most prominent examples of such subsets are the functional dependencies, see e.g. [Ullm], the full inclusion dependencies, see [KCV], and the exclusion dependencies of [CV].

EX denotes the set of dependencies which can be transformed to a sentence in prenex-normalform without universal quantifiers. Additionally every predicate symbol other than "=" appears only under an even number of negation signs. An example would be " ∃ x1, x2, y1, y2: 1(x1,x2) and 1(y1,y2) and (x1 ≠ y1 or x2 ≠ y2) " , which demands that the first relation should contain at least two different tuples.

2. A Hierarchy of Equivalences

Most of the approaches to define equivalence of database schemes use the ability to construct mappings between their states as a criterion (see for example [AABM], [Biller], [CV], [IL1], [Hull], [KK], [Koba], [Riss]). Whereas the intention of these papers is sometimes a very special one we will try to be as general as it is possible.

As usual we only want to consider instances instead of arbitrary states, since there seems to be no reason to regard database states which do not correspond to a possible real world. Furthermore, we will not consider arbitrary mappings for the definition of equivalence. For if two schemes both have an infinite set of instances, then there always exists an (mostly pathological) bijection between their instances. One way would be to consider only renaming of values, but this seems much too restrictive.

Approaches which are engaged in database normalization (as [BBG], [BMSU], [IL1], [Riss]) consider only mappings built up by natural join, projection and selection. [Koba] uses four special kinds of mappings. [Hull] is interested in

Like we have done above, schemes will always be denoted as Si (where i is a subscript), states as Ai, Bi or Ci, instances as Ii, Ji or Gi, the set of states as TYPEi, the set of instances as INSi (where the subscription shows to which scheme they belong). SYMB(Ai) stands for the set of all values appearing in the state Ai.

For two states A1,B1 $\in$ TYPE1 A1 $\subset$ B1 holds iff for all i with $1 \leq i \leq |S1|$ A1[i] $\subset$ B1[i] holds. The number of tuples of A1 is denoted by |A1|.

Database mappings are denoted by q1, p1, r1 : TYPE1 -> TYPE2 or q2, p2, r2 : TYPE2 -> TYPE1. If not defined explicitly, the schemes S1 and S2 and the mapping class Q, to which all these mappings are assumed to belong, are given globally. The obvious conventions about the subscriptioning holds if not stated something else.

q1[i] should denote the i-th part of q1, that means q1 = < q1[1], ..., q1[m] > (for |S2| = m).

Mapping classes or query classes are denoted by Q, Q1 or Q2. The most interesting class is RALG*, the set of sequences of queries expressed in the relational algebra. More precisely q1 $\in$ RALG* iff for all i q1[i] $\in$ RALG holds, where RALG is the usual relational algebra, see [Ullm]. The operators are symbolized in the following way:

- " i " stands for the i-th relation scheme
- " $\cup$ " is the union sign
- " - " is the difference sign
- " [i1, ..., ik] " symbolizes a projection
  (ij are natural numbers, assumed to be different)
- " [i1 -> i2 ->...-> ik -> i1] " symbolizes a permutation
  (ij are different natural numbers)
- " [i comp_op j] " symbolizes a restriction
  (here i, j are natural numbers, comp_op $\in$ { "=", " $\neq$ " })
- " [i comp_op 'c'] " symbolizes a selection
  (here i is a natural number, comp_op $\in$ { "=", " $\neq$ " } and c $\in$ VALUES).

If the arity of a subexpression is lower than the arity of a projection, permutation, restriction or selection applied to it, or if the arities of the both subexpressions involved in a union or difference are not the same, or if "i" is greater than the number of relations of the database schemes, we assume that the expression always yields the empty relation state of arity 1. This is to avoid partially defined mappings.

RANP* (resp. RANP) is the subclass of the relational algebra which do not contain any projection. RAND* (resp. RAND) contains the expressions without the difference sign.

It should be noted, that, for notational convenience, we do not make a clear distinction between an expression and the denoted mapping.

Our general model is based on typed database schemes with dependencies defined in a (subset of the) first-order logic. Both the class of dependencies and the class of queries will play an important role in separating decidable and undecidable cases of the equivalence problem.

The following definitions and notations are used.

TYPES is the set of types , i.e. attribute domains allowed in the definitions of database schemes. The following property holds:

$\forall$ T1, T2 $\in$ TYPES: T1 $\cap$ T2 = ϕ or T1 $\subset$ T2 or T2 $\subset$ T1.

A type can be finite or infinite. It should be a countable set.

VALUES is the set of all values appearing in such a type, that means: VALUES = { v $\in$ T: T $\in$ TYPES }.

A relation scheme R is a finite sequence of types: R1 = < T1, ..., Tk >, where all Ti $\in$ TYPES. A state of the relation scheme is a finite set of tuples, where each tuple is a sequence of values corresponding to the types of that relation scheme.

A database scheme S1 consists of a finite sequence of relation schemes and a finite set of dependencies:

S1 = < R1, ..., Rm : D >
  = < <T11,...,T1a_1>,...,<Tm1,...,Tma_m> : D >.

We use the notation |S1| = m, S1[i] = Ri, |S1[i]| = ai (the arity of the i-th relation scheme of S1).

The set of dependencies D of S1 is a finite set of first-order sentences over a signature

- with the ai-ary predicate symbol "i" ($1 \leq i \leq m$), which corresponds to the i-th relation scheme S1[i] of S1,
- the binary identity sign, which always will be interpreted as the identity over VALUES,
- a finite set of individual constants "c1", ..., "cn", where all ci $\in$ VALUES; such a constant "ci" will always be interpreted by ci.

Quantifiers in such a sentence range over VALUES (and not over the set of values appearing in a database state). Otherwise a sentence will be interpreted as usual, see [GMN], [Reiter] or [FV].

The set of states of S1 is given by TYPE1 = { < A1[1], ..., A1[m] > :
for all $1 \leq i \leq m$ the set A1[i] is a finite subset of Ti1 * ... * Tia_i }.

The set of instances of S1 is given by INS1 = { I1 $\in$ TYPE1 : I1 satisfies all dependencies of D }. It is not assumed that a dependency is domain independent, see [FV], but the set of instances of a scheme is assumed to be decidable.

the whole relational algebra and shows the important role of the database dependencies for the definition of equivalence. To cover all cases we will define equivalence with respect to a given class of mappings Q. So our approach is very similar to that of [ABM] and [AABM].

First we will define some properties of mappings between states of two database schemes.

Definition 1

q1 is <u>consistent</u> iff q1(INS1) $\subseteq$ INS2.

q1 is <u>injective</u> iff $\forall$ I1, J1 $\in$ INS(S1): I1 $\neq$ J1 ==> q1(I1) $\neq$ q1(J1) .

q1 is <u>surjective</u> iff q1(INS1) $\supseteq$ INS2.

q2 is <u>inverse to</u> q1 iff $\forall$ I1 $\in$ INS1: q2(q1(I1)) = I1.

You should note that these properties depend on the set of instances of both schemes. It is easy to show the following facts.

Theorem 2

1. If q2 is inverse to q1, then q1 is injective.
2. If q2 is inverse to q1 and q2 is consistent, then q1 is injective and q2 is surjective.
3. If q2 is inverse to q1 and q1 is surjective, then q2 is consistent and injective and q1 is injective and inverse to q2.

Proof: omitted. ▪

Now we are able to present some kinds of conceptual inclusion and equivalence of database schemes.

Definition 3

S1 <1< S2 wrt.Q iff there exists a consistent and injective q1 $\in$ Q.

S1 <2< S2 wrt.Q iff there exists a surjective q2 $\in$ Q.

S1 <3< S2 wrt.Q iff there exits a consistent q1 $\in$ Q and a q2 $\in$ Q which is inverse to q1.

S1 <4< S2 wrt.Q iff there exists a surjective q2 $\in$ Q and a q1 $\in$ Q which is inverse to q2.

For i = 1,2,3,4 let

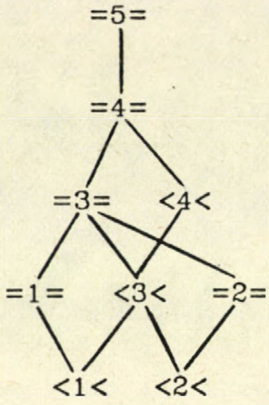
S1 =i= S2 wrt.Q iff S1 <i< S2 wrt.Q and S2 <i< S1 wrt.Q.

S1 =5= S2 wrt.Q iff there exist consistent, injective and surjective q1, q2 $\in$ Q each one being inverse to the other. ▪

Using the properties of a mapping class defined in chapter 1, it is easy to show, that all of the =i= predicates are reflexive, transitive and symmetrical.

The "weak - inclusion" of database schemes (see [AABM]) is exactly the same as our " <2< ". The "inclusion" of [AABM] is equivalent to " <3< " of our definition.

Using theorem 2 and some examples showing distinctness we are able to prove the following Hasse-diagram (the strongest property is at the top) :



It seems that all of these predicates only consider the possibility of translating any instance of one scheme into an instance of the other. A result of [ABM] however can be used to show that this can be equivalent to a comparison of the set of answers to queries.

Theorem 4

Since Q is assumed to contain the identity (on the set of states) and is closed under composition of mappings, the following holds: S1 <2< S2 wrt.Q iff $\forall$ q1 $\in$ Q $\exists$ q2 $\in$ Q $\forall$ I1 $\in$ INS1 $\exists$ I2 $\in$ INS2 : q1(I1) = q2(I2).

Proof: see [ABM], theorem 2.1 for the idea.

We reject equivalence =1=, because it is not compatible with such a comparison of answers to queries. But any of the equivalences =2=, =3=, =4=, =5= is proved to be as restrictive as the query-equivalence of [Codd]. Which of them should one choose?

An example will suggest us to reject the weakest of them.

Example 5

Let Q be RALG , S1 := < <AB>, <BC>, <AC> : ϕ >, S2 := < <AB>, <B>, <BC> : ϕ > and A, B, C $\in$ TYPES be disjoint.

Nobody would consider these schemes as being equivalent. But it turns out that S1 =2= S2 wrt.Q holds. To show this we have to construct two surjective mappings:

q1 := < "(1*3)[1=3][1,2]", "(1 - (1*3)[1=3][1,2])[2]","2">

q2 := < "1-(1*2)[2=3][1,2]", "3 - (2*3)[1=2][2,3]", " (1*2*3*)[2=3][3=4][1,5]" >

It should be noted, that none of the other equivalences holds.

So we will choose only one of the equivalences =3=, =4= and =5= to be the best. However, all our examples which show a difference between these properties look very unnatural. The next theorem states that for all usual query classes there is no real choice.

Theorem 6

If Q is a subset of FGEN, then S1 =3= S2 wrt.Q iff S1 =5= S2 wrt.Q.

Proof:

"<==" follows directly by theorem 2.3.

"==>"

We may assume four mappings in Q:

- q1 consistent and injective
- q2 inverse to q1
- p2 consistent and injective
- p1 inverse to p2.

Using the theorem of Cantor and Bernstein, especially with the more constructive proof of Koenig, one is able to construct a bijection between the set of instances of the schemes, see e.g. [Sier]. For our purpose this does not suffice, because we are looking for such a bijection in the class Q only. Furthermore, it must have an inverse mapping in Q.

In the following we will show that we don't need to construct a new mapping. It suffices to show, that

- q1 is consistent, injective and surjective
- q2 is inverse to q1.

Using theorem 2 this implies S1 =5= S2 wrt.Q.

Let [INS1] denote the set of classes of the partition of INS1 generated by the reflexive and transitive closure of the condition that two I1,J1 $\in$ INS1 with I1 = p2(q1(J1)) belong to the same class (symbolized by [I1] = [J1]).

Using q1 o p2 (instead of p2 o q1) we get a definition of [INS2] in an analogous way.

Every class consists of only a finite number of instances. Any instance of a class is mapped into another instance of the class by some iteration of p2 o q1 (respectively q1 o p2) and both p2 and q1 belong to FGEN.

(1)

For every [I1] $\in$ [INS1] the class [q1(I1)] $\in$ [INS2] is well-defined and q1([I1]) $\subseteq$ [q1(I1)].

[q1(I1)] is well-defined since q1 is consistent.

Now, let J1 $\in$ [I1], say J1=$(\text{p2} \circ \text{q1})^m$ (I1).

Then q1(J1) = q1 o (p2 o q1) o ... o (p2 o q1) (I1) = (q1 o p2) o ... o (q1 o p2) (q1(I1)).

i.e. $q1(J1) \in [q1(I1)]$.

(2)

For every $[I2] \in [INS2]$ the class $[p2(I2)] \in [INS1]$ is well-defined and $p2([I2]) \subseteq [p2(I2)]$.

This can be proved in an analogous way.

(3)

For every $[I2] \in [INS2]$ it holds that $q1([p2(I2)]) = [I2]$.

" $\subseteq$ ":

Using (1), respectively the definition of [INS2], we get $q1([p2(I2)]) \subseteq [q1(p2(I2))] \subseteq [I2]$.

" = ":

Using (2) we get $p2([I2]) \subseteq [p2(I2)]$ and therefore we can conclude that $q1(p2([I2])) \subseteq q1([p2(I2)])$ holds. Using the " $\subseteq$ " proof it follows that $q1(p2([I2])) \subseteq q1([p2(I2)]) \subseteq [I2]$.

Because both q1 and p2 are injective and [I2] is a finite set, it follows that these inclusions are really identities.

(4)

It follows directly from (3) that q1 is surjective. Since q1 is consistent and injective and q2 is inverse to q1, we can finish this proof. ▪

We argue that is reasonable to consider only query classes which do not contain mappings being able to generate an unrestricted set of new values. Therefore our attention is now directed on =5=, the sharpest formalization of equivalence.

3. The Decidability of Equivalence

In this chapter we are concerned with cases in which the equivalence =5= is decidable. One has to restrict both the set of dependencies in the schemes and the queryclass Q which determines the sharpness of equivalence. First we will introduce a simple algorithm.

Definition 7

Input: database schemes S1, S2; query class Q;

method:

```
FOR ALL q1 ∈ Q1 DO
 IF q1 is consistent
 THEN
  FOR ALL q2 ∈ Q2 DO
```

```
    IF q2 is consistent AND
       q2 is inverse to q1 AND
       q1 is inverse to q2
    THEN write ( "Proof of equivalence by", q1, q2 );
         GOTO endmark;
    FI;
   OD;
  FI;
 OD ;
 write ( "The schemes are not equivalent." );
 endmark:
```

The sets Q1 and Q2 must be subsets of Q.

Theorem 6 implies that a jump to the endmark only appears if S1 =5= S2 wrt.Q holds. The converse is usually not true. To cover exactly the equivalence we have to provide a lot more:

- an effective construction of Q1 and Q2
- which ensures that they are finite sets
- and contain mappings for the proof of equivalence iff Q contains such ones;
- an algorithm which is able to decide whether a mapping in Q is consistent;
- an algorithm which is able to decide whether a mapping is inverse to an other mapping.

In the following subchapters we will show, that for schemes with dependencies in ALL $\cup$ EX and query classes included in RANP* or in RAND* all these demands can be fullfilled.

The algorithm for consistency and the algorithm for inversion are based on the following theorem of logic on the Bernay - Schoenfinkel Class (BSC) of first-order logic sentences. The sentences of BSC are equivalent to sentences in prenex normal - form with no existential quantifiers on the right of an universal quantifier.

Theorem 8

There exists an effective algorithm which decides for an arbitrary finite subset of BSC without equality and function symbols whether it has a finite model or not.

Proof: see [DG], pp. 79.

3.1 Algorithm for Consistency

The consistency is closely related with the implied constraint problem of [Klug] and [JAK]. A mapping q1 is consistent iff all the dependencies of scheme S2 are satisfied for all states in q1(INS1).

In [Klug] the allowed dependencies are functional dependencies and the so-called equality statements. The mappings are restricted to be in RAND*.

[JAK] consider relational mappings built up by restrictions and products and generalized dependency constraints, see [GJ], as allowed dependencies.

As [JAK] we will use theorem 8 but in a different way. For the following of this chapter let D1 ⊆ ALL ∪ EX be the set of dependencies of S1, d ∈ ALL ∪ EX a dependency of S2 and q1 a mapping in RANP* or in RAND*. We have to decide whether d is satisfied by all q1(I1) where I1 ∈ INS1. To use theorem 8 we will construct a set of sentences IC in BSC such that every finite model of IC corresponds to an instance I1 ∈ INS1 for which ¬d is satisfied by q1(I1) and vice versa.

First we will mix ¬d and q1.
Let q1' be the transformation (see [Ullm]) of q1 into the domain relational calculus:

q1' = < { x11,...,x1m$_1$: f_1(x11,...,x1m$_1$ }, ..., { xn1,...,xnm$_n$: f_n(xn1,...,xnm$_n$) } >

Substitute every occurence of an "i(y1,...,ymi)" in ¬d by the formula "f_i(y1,...,ymi)" using appropriate renaming of variable symbols if needed. The resulting sentence is denoted by ¬dq1. If we add ¬dq1 as a dependency to those of S1 then for every instance I1 of this new scheme its image q1(I1) satisfy ¬d.

Now we want to show that ¬dq1 is expressible as a sentence in prenex normalform where no existential quantifier appears on the right of an universal quantifier. If q1[i] ∈ RANP then the substitution of f_i doesn't change anything, because in f_i there are no quantifiers at all. If q1[i] ∈ RAND then in f_i there appear only existential quantifiers. If d ∈ ALL, then in ¬d there are only existential quantifiers and no problems arise. In the other case, if d ∈ EX then in ¬d only universal quantifiers appear. The substitution of f_i behaves well because we assumed in the definition of EX that every "i(...)" appears only under an even number of negations.

Set IC' := D1 ∪ { ¬dq1 }.

To get a set of sentences in BSC we have to

- avoid constant symbols (as preinterpreted function symbols)
- simulate the typing of S1
- avoid the equality sign (as a preinterpreted predicate symbol).

For the first task we will introduce a special predicate symbol c of arity 1 for every constant c' appearing in IC'.

Every sentence s of IC' with constants c1,...,ck will be transformed into " ∃ x1,...,xk: c1(x1) and ... and ck(xk) and s " (here xi is different from the other variable symbols of s and from other xj). To cover the semantics of constants we add

" ∃ x: c(x)",

" ∀ x,y: c(x) and c(y) ==> x=y " and

" ∀ x,y: c(x) and d(y) ==> x≠y " for every constant c' (and every constant d' different from c') to IC'.

To simulate the concept of typed schemes we will introduce a special predicate symbol T of arity 1 for every type T' in TYPES appearing in S1. Let

{T1,...,Tn} be set of all these new predicate symbols.

We add the following sentences to IC':

" $\forall$ x: T1(x) or ... or Tn(x) "

" $\forall$ x: ¬Ti(x) or ¬Tj(x) " for all Ti' $\cap$ Tj' = ϕ .

" $\forall$ x: Ti(x) ==> Tj(x) " for all Ti' $\subseteq$ Tj'.

" $\forall$ x1,...,xm: i(x1,...,xm) ==> Ti1(x1) and ... and Tim(xm) "

for all i with S1[i] = < Ti1, ..., Tim >.

For every finite type T' $\in$ TYPES (with exactly n elements) we need additional sentences:

" $\exists$ x1,...,xn: T(x1) and ... and T(xn) and ¬ (x1=x2 or x1=x3 or ... or x1=xn or x2=x3 or ... or x2=xn or ... or x(n-1)=xn) ",

" $\forall$ x0,x1,...,xn: T(x0) and ... and T(xn) ==> x0=x1 or x0=x2 or or x1=x2 or ... or x1=xn or ... or x(n-1)=xn ".

At last we must bind constants to their types:

" $\forall$ x: c(x) ==> T(x) " for all predicate symbols c corresponding to the constant c' and all types T which contain c'.

To handle the equality sign as a normal predicate symbol we will use the axioms of equality of [Reiter]. Only universal quantifiers are needed here.

By construction it should be clear, that the resulting set IC of sentences is a (finite) subset of BSC. It should also be clear, how to use this construction for an algorithm for the decision of consistency. So we get :

Theorem 9

The consistency is effectively decidable with respect to

- mappings in RANP* or in RAND*
- schemes with dependencies in ALL $\cup$ EX.

3.2 Algorithm for Inversion

The decision whether a mapping is inverse to another one can be handled as a special case of the decision of the equivalence of two mappings (exactly: of their syntatical description).

Definition 10

q1 is <u>equivalent</u> to p1 iff $\forall$ I1 $\in$ INS1: q1(I1) = p1(I1).

Note that we use here equivalence restricted to the set of instances, in [Klug] called 'equivalence', and not equivalence with respect to all states, in [Klug] called 'strong equivalence'. [ASU] distinguish between 'algebraic equivalence', 'weak equivalence' and 'strong equivalence' of mappings. The last one as defined in [GM] is the same as our equivalence. All these references present algorithms to decide equivalence in special cases. [SY] generalize the results of [ASU]. [IL2] is concerned with the undecidability of the equivalence of mappings, whereas [IL1] shows how to decide it under the open-world assumption.

Theorem 11

q2 is inverse to q1 iff (q2 $\circ$ q1) is equivalent to id_INS1. Here id_INS1 denote the identity on INS1.

Proof: obvious.

Using the result of the previous chapter we easily get the following fact.

Theorem 12

The equivalence of relational mappings

- in RANP* or in RAND*
- between schemes with dependencies in ALL $\cup$ EX is effectively decidable.

Proof:

Let q1, p1 be given and let |S2| = n.

Define S3 := < S2[1], ..., S2[n], S2[1], ..., S2[n] : D3 >
and r1 := < q1[1], ..., q1[n], p1[1], ..., p1[n] >, where D3 consists of a set of dependencies, such that

$\forall$ I3 $\in$ TYPE3: I3 $\in$ INS3 <==> $\forall$ 1 $\leq$ i $\leq$ n : I3[i] = I3[i+n]

holds. It is obvious,that D can be constructed as a set of full inclusion dependencies, which can be expressed as " $\forall$ x1,...,xn: i(x1,...,xn) ==> j(x1,...,xn) " and thus D is in ALL. The mapping r1: TYPE1 -> TYPE3 is constructed in a way, such that q1 and p1 are equivalent iff r1 is consistent.

Since D3 is a subset of ALL $\cup$ EX, the dependencies of S1 are assumed to be in ALL $\cup$ EX and r1 is in RANP* or in RAND* we can finish this proof with a reference to Theorem 9. ▪

3.3 Finite Mapping Sets

The algorithm of definition 7 only works if we are able to construct finite subsets Q1 and Q2 of Q which contain mappings for the proof of the equivalence =5= (if Q contains such mappings at all). Our attention is directed on RANP* and RAND*. First we will define their normalforms.

Definition and Theorem 13

Every mapping q in RANP* is expressible in a way such that

- no selection refers to a subexpression in which a product appears (for short: selection before product)
- product before restriction
- restriction before permutation

- permutation before difference
- difference before union
- no projection appears.

An expression of this form will be called to be in normalform (for RANP*).

Every mapping q in RAND* is expressible in a way such that

- selection before product
- product before restriction
- restriction before projection
- projection before union
- no permutation and no difference appears.

An expression of this form will be called to be in normalform (for RAND*).

Proof:

For the proof for RAND* see [Klug].

Then for RANP* it suffices to show how to shift the difference on its right place (let Ei be arbitrary subexpressions):

(E1 - E2) [i='v'] --> E1 [i='v'] - E2 [i='v']
(E1 - E2) * E3 --> (E1 * E3) - (E2 * E3)
(E1 - E2) [i=j] --> E1 [i=j] - E2 [i=j]
(E1 - E2) [perm] --> E1 [perm] - E2 [perm] ,
where perm = " i1->i2->...->ik->i1"
(E1 ∪ E2) - E3 --> (E1 - E3) ∪ (E2 - E3)
E1 - (E2 ∪ E3) --> (E1 - E2) - E3

Next we will show what must be assumed to get finite mapping classes.

Theorem 14

Let C ⊆ VALUES be a finite set.

Then the set of mappings TYPE1 -> TYPE2 in RANP* with selections using only constants of C is finite and can be enumerated in an effective way.

Proof:

The proof is based on the following observations, whereas the details are left to the reader. Every such mapping has a normalform according to theorem 13. Since projection is not allowed the number of products is restricted by TYPE1 and TYPE2. ▪

In chapter 3.4 we will show how to construct such a finite set C of values.

To get an analogous result for RAND* we have to make an additional assumption to restrict the arity of a subexpression. For example there is a state of a binary relation with 100 tuples, whose transitive closure (see [AU]) is expressible in RAND*, but needs at least 99 product signs. So we would not get a finite set of mappings if we don't exclude such cases.

Theorem 15

Let C ⊆ VALUES be a finite set and l be a natural number.

Then the set of mappings TYPE1 -> TYPE2 in RAND* with selection constants in C and no more than l product signs in every subexpression which doesn't contain a union sign or a difference sign is finite and can be enumerated in an effective way.

Proof: Similar to the proof of theorem 14. ▪

In chapter 3.5 we are concerned with the question how to get such a natural number l with respect to the given schemes.

3.4 Relevant Selections

First we will characterize the selection constants appearing in a relational expressing in a better way. In the following K, L, M are finite subsets of VALUES.

Definition 16

A mapping f : VALUES -> VALUES is called a K-isomorphism iff

- f is bijective and totally computable
- $\forall k \in K$: $f(k) = k$
- $\forall T \in$ TYPES: $f(T) = T$. ▪

In the canonical way isomorphisms are extended to tuples, states and sets of states. See [CH] or [Hull] for similar definitions. The following properties hold:

- if f, g are K-isomorphisms, then $f \circ g$, f^{-1} are K-isomorphisms
- for any scheme S1: f(TYPE1) = TYPE1.

Definition 17

A scheme mapping q1 is compatible with K-isomorphisms iff for every K-isomorphism f $q1 \circ f = f \circ q1$ holds. ▪

In [Banc] and [CH] a similar property is used in the definition of completeness of a query language. It is obvious that every relational expression whose selection constants are contained in K is compatible with K-isomorphisms. The other direction is not as simple. This is because we allow types with a finite number of values. The construction in the proof of theorem 19 shows how to get a weaker result.

Corresponding to the compatibility of mappings we will formulate an analogous property for database schemes.

This property is related to the monotonicity (see e.g. [SY]) and the additivity (see e.g. [AU]) of mappings. It should be noted, that every mapping with breadth l is also monotone. The converse does not hold. The transitive closure (see e.g. [AU]) is a prominent counterexample for it.

The next theorem will state how we can use the breadth to restrict the number of products.

Theorem 21

Let TYPES contain only disjoint sets.

If there exist a natural number l and a finite subset M of VALUES, such that q1 is totally computable, compatible with M-isomorphisms, member of NGEN and has breadth l

then q1 is expressible in the normalform of RAND*, where no more than $l + m - 2$ product signs appear in every of its subexpressions which have no union or difference sign. Here $m := \max \{ |S2[j]| : 1 \le j \le |S2| \}$.

Proof:

Choose a finite set of states $B1, \dots, Bp \in TYPE1$ as substitutes of all instances with no more than l tuples. That means:
$\{ A1 \in TYPE1 : |A1| \le l \} = \{ f(Bi) : f \text{ is an M-isomorphism}, 1 \le i \le p \}$.

To do this one can define a finite set $L \subseteq VALUES - M$ (where $|L| \ge \max \{ SYMB(A1) : A1 \in TYPE(S1), |A1| \le l \}$) and enumerate all states with values in $L \cup M$ and with no more than l tuples.

Then for every $A1 \in TYPE1$:

$$q1(A1) = \bigcup_{1 \le i \le p} \ \bigcup_{\substack{f \text{ is M-isom.} \\ f(Bi) \subseteq A1}} q1(f(Bi)) \qquad \text{by construction of } B1,\dots,Bp \text{ and since } q1 \text{ has breadth } l$$

$$= \bigcup_{1 \le i \le p} \ \bigcup_{\substack{f \text{ is M-isom.} \\ f(Bi) \subseteq A1}} f(q1(Bi)) \qquad \text{since } q1 \text{ is compatible with M-isomorphisms}$$

$$= \bigcup_{1 \le i \le p} \ \bigcup_{\substack{f \text{ is M-isom.} \\ f(Bi) \subseteq A1}} f\Big(\bigcup_{\substack{D \subseteq q1(Bi) \\ |D| = 1 \\ D \in TYPE2}} D \Big)$$

$$= \bigcup_{1 \le i \le p} \ \bigcup_{\substack{D \subseteq q1(Bi) \\ |D| = 1 \\ D \in TYPE2}} \ \bigcup_{\substack{f \text{ is M-isom.} \\ f(Bi) \subseteq A1}} f(D) \qquad \text{since } f \text{ is a canonical extension}$$

For every $1 \le j \le |S2|$ then

Now f is defined by exchanging every $w \in SYMB(A1) \cap (L - M)$ with its associated element, i. e.

$$f(x) := \begin{cases} w' & \text{if } x = w \in SYMB(A1) \cap (L - M) \\ w & \text{if } x = w' \text{ with } w \in SYMB(A1) \cap (L - M) \\ x & \text{else.} \end{cases}$$

By definition $f = f^{-1}$ holds.

(3)

We will show for an arbitrary $I1 \in INS1$: $p1(I1) \in INS2$. Let f be the M-isomorphism for I1 as constructed in (2).

Because of $K \subseteq M$ we get $f(I1) \in INS1$ and $f(INS2) \subseteq INS2$, since both schemes are compatible with K-isomorphisms. So

$$\begin{aligned} p1(I1) &= p1(f(f(I1))) \\ &= f(p1(f(I1))) && \text{,since p1 is compatible with M-isomorphisms} \\ &= f(q1(f(I1))) && \text{,using (1)} \\ &\in f(INS2) && \text{,since q1 is consistent} \\ &\subseteq INS2 \ (4) \end{aligned}$$

We will show $p2(p1(I1)) = I1$ for an arbitrary $I1 \in INS1$.

Using the argumentation of (3) we know that $p1(I1) = f(q1(f(I1)))$ and so $p2(p1(I1)) = p2(f(q1(f(I1))))$.

Since p2 is compatible with M-isomorphisms we get
$p2(p1(I1)) = f(p2(q1(f(I1))))$.

Because $q1 \in NGEN$ we know that $SYMB(q1(f(I1))) \cap (L - M) = \phi$
and by (1) we get $p2(p1(I1)) = f(q2(q1(f(I1))))$.
Since q2 is inverse to q1 $p2(p1(I1)) = f(f(I1)) = I1$. ▪

Using theorem 6 we see that this approximation can also be used for =5=. In the case of RANP* we then get finite Q1 and Q2 for the algorithm of definition 7 by theorem 14.

3.5 Number of Products

In chapter 3.4 we are concerned with the compatability with isomorphisms to characterize the set of selections needed to describe a mapping as a relational expression. Now we will formulate a property which corresponds to the number of products.

Definition 20

Let l be a natural number.

A mapping q1 is of <u>breadth</u> l iff for all $A1 \in TYPE1$

$$q1(A1) = \bigcup_{\substack{B \in TYPE1 \\ B \subseteq A1 \\ |B| \leq l}} q1(B) \quad \text{holds.}$$

Definition 18

A scheme S1 is <u>compatible with K-isomorphisms</u> iff for every K-isomorphism f $f(INS1) \subseteq INS1$ holds. ▪

From our definition of database schemes (especially of their dependencies) it follows that a scheme is compatible with K-isomorphisms if in its dependencies only constants out of K appears. Therefore we are able to construct such a finite set of values for a given scheme in a simple way.

Now we are able to give the main theorem of this chapter.

Theorem 19

Let $Q \subseteq RALG^*$ be a mapping class which contains the restriction (e.g. RANP* or RAND*). Let $K \subseteq VALUES$ be a finite set and S1, S2 be both compatible with K-isomorphisms.

Then we can effectively construct a finite set M, such that
S1 <3< S2 wrt. Q iff S1 <3< S2 wrt. M_Q,
where $M\_Q := \{ q \in Q :$ in q appears no selection constant not contained in M $\}$.

Proof:

"<==" is trivial for every M.

"==>"

Let $q1 \in Q$ be consistent and $q2 \in Q$ be inverse to q1. We will construct two new mappings $p1, p2 \in M\_Q$ having the same properties.

Let $L := \{ w : w$ is a selection constant of q1 or q2 $\}$ and
$M := K \cup \{ w \in T: T \in TYPES$, T appears in S1 or S2, T is finite $\}$.

Both L and M are finite sets. Obviously q1 and q2 are both compatible with L-isomorphisms.

Let p1 (resp. p2) be the transformation of q1 (resp. q2) implied by the following rules :

r [i='w'] --> r [i≠i] for $w \in L - M$,

r [i≠'w'] --> r for $w \in L - M$, here r stands for a relational subexpression.

Since in p1 and p2 only selection constants of M appear, both of the mappings belong to M_Q. So they are compatible with M-isomorphisms.

We have to show that p1 is consistent and that p2 is inverse to p1.

(1)

Obviously $qi(Ai) = pi(Ai)$ holds for all $Ai \in TYPEi$ (i = 1,2) with $SYMB(Ai) \cap (L - M) = \phi$.

(2)

For a given $A1 \in TYPE1$ there exists a M-isomorphism f such that $SYMB(f(A1)) \cap (L - M) = \phi$.

In order to define such a M-isomorphism we consider any $w \in SYMB(A1) \cap (L - M)$. Let T be the smallest type containing w. By definition of M T is infinite and thus we can associate w with a new element $w' \in T - ((L - M) \cup SYMB(A1))$.

$$q1(A1)[j] = \bigcup_{1 \le i \le p} \;\; \bigcup_{\substack{s \in q1(Bi)[j] \\ s \text{ is a tupel}}} \left(\bigcup_{\substack{f \text{ is M-isom.} \\ f(Bi) \subseteq A1}} f(s) \right) \quad \text{holds.}$$

The number of different i and s in that formula is finite. Let i and s be fixed. It suffices to show that the mapping

$$:= \bigcup_{\substack{f \text{ is M-isom.} \\ f(Bi) \subseteq A1}} f(s) \quad \text{does not need more than } l + m - 2 \text{ product signs.}$$

To do this we first choose an enumeration of the tuples of Bi such that $Bi = < \{t11,...,t1n_1\}, \{t21,...,t2n_2\}, ..., \{tv1,...,tvn_v\} >$.

Let $t := < t11, ..., t1n_1, t21, ..., t2n_2, ..., tv1,..., tvn_v >$.

Define the sequence of products, corresponding to t, which maps Bi into t. Since $|Bi| \le l$ there appear no more than l-1 product signs. Then use a sequence of selections, so that
for all $w \in M$ and $1 \le i \le |t|$ there is a " [i = 't[i]'] " iff $t[i] \in M$,
and there is a " [$i \neq$ 'w'] " iff $t[i] \notin M$.

Finally we need a sequence of restrictions, such that for all $i,j \in \{ 1,...,|t| \}$, $i < j$ there is a " [i = j] " iff t[i] = t[j] and there
appears a " [$i \neq j$] " iff $t[i] \neq t[j]$.

Let p1 denote the constructed relational mapping.

Obviously $\forall$ A1 $\in$ TYPE1: Bi $\subseteq$ A1 <==> t $\in$ p1(A1) holds.
Since p1 is compatible with M-isomorphisms f
$\forall$ A1 $\in$ TYPE1: f(Bi) $\subseteq$ A1 <==> f(t) $\in$ p1(A1) holds.

$$\text{Therefore we know that } p1(A1) \supseteq \bigcup_{\substack{f \text{ is M-isom.} \\ f(Bi) \subseteq A1}} f(s) .$$

The other inclusion is obtained by
$\forall$ A1 $\in$ TYPE1: p1(A1) $\subseteq$ { f(t) : f is M-isomorphism }.

This only holds, because we have assumed that all types contain different values. In this case we are able to construct a M-isomorphism f for a given t' $\in$ p1(A1).

The last step of the construction of the relational mapping must build up f(s) from f(t). Because of q1 $\in$ NGEN, a projection usually suffices to describe it. But if a value appears in s more times than it does in t, we need additional products (and restrictions to link such a copied tuple tij with t). A simple reflection shows that no more than m - 1 additional product signs are needed. The normalform can be reached without new products. ▪

A simple induction would show that the converse implication of theorem 21 is also true. So one would get a characterization of RAND* not using syntactic criterions, provided that TYPES contains only disjoint sets of values.

We will now define a suitable property for database schemes.

Definition 22

Let l be a natural number.
The database scheme S1 is of <u>breadth</u> l iff

$$\text{for all } I1 \in INS1 \quad I1 = \bigcup_{\substack{J \in INS1 \\ J \subset I1 \\ |J| \leq l}} J \quad \text{holds.}$$

The breadth of a scheme is a special case of the locality of [IS] and the distributivity of [IS]. Although looking similar the boundedness of [GV] and the boundedness of [AV] are scarcely related to the breadth.

We will now formulate the main theorem of this chapter.

Theorem 23

Let S1 and S2 be database schemes with breadth l.
Then one can effectively compute a natural number w such that

S1 =5= S2 wrt. NGEN* ∩ { q is monotone } iff

S1 =5= S2 wrt. NGEN* ∩ { q is monotone, totally computable and has breadth w }.

Proof:

"<==" is trivial for every number w.

"==>"

Let q1, q2 ∈ NGEN* be monotone, consistent, injective and surjective, each one being inverse to the other.

Choose w such that

(1) $\forall$ I1 ∈ INS1: $|I1| \leq l$ ==> $|q1(I1)| \leq w$ and
(2) $\forall$ I2 ∈ INS2: $|I2| \leq l$ ==> $|q2(I2)| \leq w$ holds.

One way to do this is the following:

ms1 := max { |SYMB(A1)| : A1 ∈ TYPE1, $|A1| \leq l$ }
w1 := max { |A2| : $|SYMB(A2)| \leq$ ms1 }
Since q1 ∈ NGEN* holds, w1 can be used as w to fullfill (1).
In the same manner we get w2 to fullfill (2) and we finally set
w := max { w1, w2 } to fullfill both (1) and (2).

We will define substitutes p1, p2 for q1, q2 which are in the demanded mapping class:

$$\text{for all } A1 \in TYPE1 \quad p1(A1) := \bigcup_{\substack{B \in TYPE1 \\ B \subset A1 \\ |B| \leq w}} q1(B)$$

and

for all $A2 \in TYPE2$ $\quad p2(A2) := \bigcup_{\substack{B \in TYPE2 \\ B \subseteq A2 \\ |B| \leq w}} q2(B)$.

It suffices to show that $q1(I1) = p1(I1)$ holds for all instances $I1 \in INS1$ (but not necessarily for all states). The analogous result for p2 can be obtained in the same way.

" $\subseteq$ ":

Since q1 is consistent and S2 has the breadth 1 we get for an

arbitrary $I1 \in INS1$: $\quad q1(I1) = \bigcup_{\substack{G2 \in INS2 \\ G2 \subseteq q1(I1) \\ |G2| \leq 1}} G2$

Since q1 is surjective each G2 is represented by $q1(J1)$ for an $J1 \in INS1$:

$$q1(I1) = \bigcup_{\substack{q1(J1) \in INS2 \\ q1(J1) \subseteq q1(I1) \\ |q1(J1)| \leq 1 \\ J1 \in INS1}} q1(J1)$$

$$= \bigcup_{\substack{q1(J1) \in INS2 \\ q1(J1) \subseteq q1(I1) \\ |q1(J1)| \leq 1 \\ J1 \in INS1 \\ |J1| \leq w}} q1(J1) \qquad \text{using (2), q2 is inverse to q1}$$

$$\subseteq \bigcup_{\substack{q1(J1) \in INS2 \\ q1(J1) \subseteq q1(I1) \\ J1 \in INS1 \\ |J1| \leq w}} q1(J1)$$

$$= \bigcup_{\substack{q1(J1) \in INS2 \\ J1 \subseteq I1 \\ J1 \in INS1 \\ |J1| \leq w}} q1(J1) \qquad \text{using the monotonicity of q2, which is inverse to q1}$$

$$= \bigcup_{\substack{J1 \in INS1 \\ J1 \subseteq I1 \\ |J1| \leq w}} q1(J1) \qquad \text{since q1 is consistent}$$

$$\subseteq \bigcup_{\substack{J1 \in TYPE1 \\ J1 \subseteq I1 \\ |J1| \leq w}} q1(J1) \qquad \text{since } INS1 \subseteq TYPE1$$

$$= p1(I1)$$

" ⊇ ": By monotonicity of q1. ▪

The property of a scheme to have finite breadth is essentially a demand on its dependencies. We must really restrict the class of allowed dependencies as the following example will show.

Example 24

S := < <N,N>, <N,N> : { " ∀ x,y: 1(x,y) ==> ∃ z: 2(x,z) ",
" ∀ x,z: 2(x,z) ==> ∃ y: 1(y,z) " } >,
where N should denote the set of all natural numbers.

I := < { <n,n> : 1 ≤ n ≤ m }, { <n-1,n> : 2 ≤ n ≤ m } >,
where m is a given natural number.

Any instance J with J ⊆ I and <1,1> ∈ J[1] is identical to I. This is implied by the two inclusion dependencies of S.

Therefore S cannot have breadth l if l < 2*m - 1.

Because m is choosen arbritrary S does not have finite breadth at all. ▪

If we restrict our attention on dependency classes already shown to be good-natured with respect to the decision algorithms, we are able to compute an approximation of the breadth of a scheme.

Theorem 25

Let S1 be a scheme with dependencies in ALL ∪ EX. Let TYPES contain only disjoint sets.

Then one can effectively compute a number l such that S has breadth l.

Sketch of the proof:

Set l := (m + k + e) ** a , where

- a is the number of attributes of S,
- m is the greatest number of attributes of a relation scheme of S1,
- k is the number of constants appearing in dependencies of S1,
- e is the number of occurences of existential quantification in the dependencies of S, provided they are written in prenex normalform.

Since for all I1 ∈ INS1 I1 = ∪ { B : B ⊆ I1, |B| = 1, B ∈ TYPE1 } holds, it suffices to show that for given I1 ∈ INS1, B ∈ TYPE1, |B| = 1 there is an instance J1 ∈ INS1, so that B ⊆ J1 ⊆ I1 and |J1| ≤ l holds.

We will build a set of sentences in BSC to characterize such an J1. If they have a finite model at all, they have a model with no more than m + k +

e individuals. The correspondence of suc [illegible] lel to a state with no more than l tuples is obvious.

$K := \{ k : k \in SYMB(B)$ or k appears as a constant in a dependency of $S \}$

(1)

Let in_I $\in$ ALL be a sentence, such that for every finite model M there exist a corresponding state $A1 \in TYPE1$ and a K-isomorphism f such that $f(A1) \subseteq I1$. These sentences can be constructed in an obvious manner. The proof of theorem 21 contains a similar construction (if translated into the relational calculus). Only constants in K appear in this sentence.

(2)

Let B_in be a sentence without quantifier, so that every finite model M corresponds to a state A1 such that $B \subseteq A1$ holds.
A sentence "i(v1,....,vk)" suffices, if $\langle v1,\ldots,vk \rangle \in B[i]$. Only constants of K appear in B_in.

(3)

Let D be the set of dependencies of S. It is assumed that $D \subseteq ALL \cup EX$.

Let $E := D \cup \{$ in_I, B_in $\}$.

Every model of E corresponds to a state $A1 \in TYPE1$, so that

- there exists a M-isomorphism f with $f(A1) \subseteq I1$ (by (1))
- $A1 \in INS1$ (by $D \subseteq E$)
- for the above chosen f $f(A1) \in INS1$ holds (since S is compatible with K-isomorphisms)
- $B \subseteq A1$ (by (2))
- for the above chosen f $B \subseteq f(A1)$ holds (since $SYMB(B) \subseteq K$ and so $f(B) = B$ holds).

Therefore we know of an $f(A1) \in INS1$ with $B \subseteq f(A1) \subseteq I1$ and must finally show that there is a model of E with no more than $m + k + e$ individuals.

(4)

We want to use the theorem of Herbrand.

To do this we have to eliminate the equality and the constants as preinterpreted objects.

First we will eliminate the constants. We build the conjunction of all sentences of E, substitute every constant symbol with a new specific variable symbol, bind these variables globally with existential quantifiers and add atoms which demand that they all have different values. Let E' denote this new sentence.
Since $E \in ALL \cup EX$ it is obvious that E' belongs to BSC.
Every model of E is also a model of E'. For every model of E' there is a model of E having the same number of individuals.

Next we use the axioms of equality of [Reiter] to handle the equality symbol as a normal predicat symbol. Let $E'' \subseteq BSC$ denote the constructed set of sentences. Every model of E' is also a model of E''. For every model of E'' there is a model of E' having no more individual symbols.

The number of existential quantification in E'' is not greater than m + k + e. This is also the number of terms of the Herbrand universe of E'', since the Skolemization of E'' only generate function symbols with arity 0. This is because E'' is in BSC.

Using Herbrand's theorem we can conclude that there is a model of E with no more than m + k + e individuals if there is a model at all. ▪

3.6 Summary

Theorem 26

Let S1 and S2 be schemes with dependencies in ALL ∪ EX.
Let TYPES contain only disjoint sets.
Let Q be a subset of RANP* or of RAND*, which contains the restriction.

Then the algorithm of definition 7 can be used to decide whether S1 =5= S2 wrt.Q or not.

Proof:

In chapter 3.1, theorem 9 we have suggested a way to decide if a mapping is consistent. In chapter 3.2, theorem 12 we have shown how to decide the equivalence of two mappings. Using theorem 11 it is obvious how to use this for a decision whether a mapping is inverse to another mapping. In chapter 3.3, theorem 14 it is shown how to enumerate finite sets Q1 and Q2 in the case of Q ⊂ RANP* if we know a finite set of selection constants relevant for the proof of equivalence. In chapter 3.4, theorem 19 can be used to get such a set of constants. In chapter 3.3, theorem 15 we have seen, that in the case of Q ⊂ RAND* we need additionally an approximation of the number of products to get finite sets Q1 and Q2. In chapter 3.5, theorem 21 we have formulated the breadth of a mapping as a sufficient criterion for this matter. In chapter 3.5, theorem 25 we suggested a way to approximate the breadth of a scheme. In chapter 3.5, theorem 23 we finally have proved that there is no need to consider mappings with a breadth not related to the breadth of the schemes. ▪

4. Undecidability of Equivalence

In chapter 3 we have only considered schemes with dependencies in ALL ∪ EX and mapping classes below the relational algebra. Now we want to show the reason for these restrictions.

4.1 More Powerful Dependency Classes

In the following we will use "implication" as "finite implication" (see [CFP], [CLM]), because we only deal with finite database states. We use a known result concerning with the implication problem for database dependencies.

Theorem 27

It is not decidable whether S1 =3= S2 wrt.Q holds

- where S1, S2 ranges over all schemes with functional dependencies and (binary) inclusion dependencies
- Q is a mapping class which includes the identity mapping (as assumed in chapter 1) and is contained in FGEN.

Proof:

See [Mitch] for the undecidability of the (finite) implication problem for functional dependencies and binary inclusion dependencies.

Let D, { d } be arbitrary sets of dependencies of these classes, expressed in the notation of chapter 1. It should be noted, that inclusion dependencies are not included in ALL ∪ EX.

Using the signature of both dependency sets it is simple to construct two schemes S1 and S2 so that TYPE1 = TYPE2 and their set of dependencies is D resp. D ∪ {d}. Only one infinite type should appear in the scheme definitions.

We want to show that this is already a correct reduction of the implication problem into the equivalence problem for database schemes.

"==>"

If d is implied by D, then INS1 = INS2 holds. The identity mapping can be used to prove S1 =3= S2 wrt.Q.

"<=="

If S1 =3= S2 wrt.Q holds, then there is a mapping q1 in Q which is consistent and injective.
For finite subsets V ⊆ VALUES let (for i = 1,2)
INSSYMB(Si,V) := { Ii ∈ INSi: SYMB(Ii) ⊆ V }.

Since q1 ∈ FGEN there is a finite N ⊆ VALUES such that q1 only generates new values in N.
Since q1 is consistent for any finite V ∈ VALUES
q1 (INSSYMB(S1,V∪N)) ⊆ INSSYMB(S2,V∪N).

By definition of S1 and S2 it is obvious, that
INSSYMB(S2,V∪N) ⊆ INSSYMB(S1,V∪N).
Because q1 is injective and INSSYMB(S1,V∪N) is a finite set, it follows that INSSYMB(S1,V∪N) = INSSYMB(S2,V∪N).

Clearly every instance of S1 is contained in INSSYMB(S1,V∪N) for some finite V (since any instance is assumed to be finite). So we get INS1 = INS2.

This means that d is implied by D.

4.2 More Powerful Mapping Classes

We are not aware of a result concerning the equivalence directly. We will show the undecidability of the consistency, injectivity and the surjectivity of mappings ranging over the whole relational algebra. The following theorem deals as the base for it.

Theorem 28

Let $T \in$ TYPES be an infinite set.

It is not decidable whether $q1(A1) = \phi$ holds for all $A1 \in$ TYPE1, where

- S1 ranges over all database schemes without dependencies
- q1 ranges over all mappings TYPE1 -> T in the relational algebra.

Proof:

A simple construction reduces the equivalence problem for relational expressions on the above mentioned problem since we can use the difference operator.

It is known that the equivalence problem of relational expressions is not decidable, see [IL2]. [Klug] cites a result of [Solo] considering expressions without selections. A direct way to prove it can be based on the undecidability of the first-order logic (finite models only). ▪

Theorem 29

The consistency with respect to

- mappings in RALG
- from a database scheme without dependencies
- into a database scheme with one functional dependency, one exclusion dependency (see [CV]), or one unary inclusion dependency (see [KCV])

is undecidable.

Proof:

Let q1: TYPE1 = INS1 -> T be given. We will construct a scheme S2 and a mapping p1, which is consistent iff $q1(A1) = \phi$ for all $A1 \in$ TYPE1. This suffices to use theorem 28.

one functional dependency:

p1 := < (SYM * SYM * q1)[1,2] >, where SYM is a relational expression which supply the relation consisting of all values appearing in a state A1,

d := " $\forall$ x, y, z: 1(x,y) and 1(x,z) ==> y = z ", which is a functional dependency,

S2 := < < T, T > : { d } >.

Then p1 is consistent iff $p1(\text{INS1}) \subseteq \text{INS2}$ iff $\forall$ $I1 \in$ TYPE1: $|\text{SYMB}(I1)| \leq 1$ or $q1(I1) = \phi$.

Since we are able to decide whether $q1(I1) = \phi$ holds for all $I1 \in$ INS1 with $|\text{SYMB}(I1)| \leq 1$, this suffices to use theorem 28.

one exclusion dependency:

$p1 := < q1, q1 >$,
$d := " \forall x: \neg 1(x) \text{ or } \neg 2(x) "$, which is an exclusion dependency,
$S2 := < < T >, < T > : \{ d \} >$.

Then p1 is consistent iff $q1(I1) = \phi$ for all $I1 \in INS1$.

one unary inclusion dependency:

$p1 := < q1, q1 - q1 >$,
$d := " \forall x: 1(x) \implies 2(x) "$, which is a unary inclusion dependency,
$S2 := < < T >, < T > : \{ d \} >$.

Then p1 is consistent iff $q1(I1) = \phi$ for all $I1 \in INS1$.

Theorem 30

The injectivity with respect to

- mappings in RALG
- schemes without dependencies

is undecidable.

Proof:

Let the mapping $q1: TYPE1 = INS1 \to T$ be given, where w.l.o.g. $|S1| = n$, $|S1[i]| = ai$ (for $1 \le i \le n$).

For $1 \le i \le n$ define
$p1[i] := (i * (SYM - (SYM * q1)[1]))[1, 2, ..., ai]$,
where SYM is the same as in the proof of theorem 35.
It is obvious, that for every $A1 \in TYPE1$ $p1(A1) \in \{ \phi , A1 \}$ holds.

Since p1 does not generate values we know that $p1(\phi) = \phi$.

Therefore p1 is injective iff $p1(A1) = A1$ for all $A1 \in TYPE1$. This means that $q1(A1) = \phi$ for all $A1 \in TYPE1$ and theorem 28 can be used to show the undecidability of injectivity.

Theorem 31

The surjectivity with respect to

- mappings in RALG
- schemes without dependencies

is undecidable.

Proof:

The same construction as for the proof of theorem 30 is used.

It is obvious that p1 is surjective
iff $p1(A1) = A1$ for all $A1 \in TYPE1$ holds, i.e.
iff $q1(A1) = \phi$ for all $A1 \in TYPE1$ holds.

5. Conclusions

There remain some open questions. Using theorem 21 (and assuming that the typing of database schemes is not of interest or behaves well) we get an exact characterization of mappings expressible by the relational algebra without the difference operator. They are totally computable, are compatible with M-isomorphisms (where M is a finite set of values), does not generate new values and have finite breadth. There is no idea of a set of properties characterizing the relational algebra without the projection operator.

When we considered the whole relational algebra in chapter 4 we have only shown the undecidability of some properties necessary for equivalence. There is no result dealing with the equivalence itself.

In chapter 3 we have shown the decidability of consistency and inversion. We do not know whether there are algorithms for the injectivity and surjectivity, too. [IL2] deals with the losslessness (the same as injectivity) under the open-world assumption, but this is a much more weaker property than our injectivity. Restricting the attention on the algebra without projection (and schemes with dependencies in ALL $\cup$ EX) the decidability of the surjectivity can be proved in a similar manner as the decidability of the consistency.

In chapter 3 we only consider mappings in RANP* or in RAND*, but do not handle with (RANP $\cup$ RAND)*, which would be the "mixing of both classes". We do not know how to change chapter 3.5 for this purpose.

Although defining TYPES as a hierarchy in chapter 1 in theorem 21 we have assumed that it contains only disjoint sets of values. There are some possibilities to change the definition of an M-isomorphism in a way that this assumption would not be necessary, but other difficulties would appear elsewhere in chapter 3.5.

At last it should be mentioned that this paper does not deal with very efficient ways to decide the equivalence. Full use of the typing of database schemes will speed up our algorithm. Cardinality comparisons as described in [Hull] can probably be used in more advanced decision algorithms.

References:

[ABM] G.Ausiello, C.Batini, M.Moscarini: 'Conceptual relations between databases transformed under join and projection', in: Proc. Symp. Math. Found. of Comp. Sc., 9, 1980, pp. 123 - 136

[AABM] P.Atzeni, G.Aussiello, C.Batini, M.Moscarini: 'Inclusion and equivalence between relational database schemes', in: Theoretical Computer Science, 19, 1982, pp. 267 - 285

[AP] P.Atzeni, D.S.Parker: 'Assumptions in relational database theory', in: ACM Symp. on Princ. of Database Systems, 1, 1982, pp. 1 - 9

[ASU] A.V.Aho, Y.Sagiv, J.D.Ullman: 'Equivalences among relational expressions', in: SIAM Journ. of Computing, 8, 1979, pp. 218 - 246

[AU] A.V.Aho, J.D.Ullman: 'Universality of data retrieval languages', in: ACM Symp. on Princ. of Programming Languages, 6, 1979, pp. 110 - 117

[AV] S.Abiteboul, V.Vianu: 'Transactions in relational databases', in: Proc. ACM Int. Conf. on Very Large Data Bases, 1984

[Banc] F.Bancilhon: 'On the completeness of query languages for relational data bases', in: Proc. Symp. Math. Found. of Computer Science, 7, 1978, pp. 112 - 123

[BBG] P.A.Bernstein, C.Beeri, N.Goodman: 'A sophisticated introdution to database normalization theory', in: Proc. ACM Int. Conf. on Very Large Data Bases, 4, 1978, pp. 113 - 124

[Biller] H.Biller: 'On the equivalence of database schemes - a semantic approach to data translation', in: Information Systems 4. 1979 , pp. 35 - 47

[BMSU] C.Beeri, A.O.Mendelzon, Y.Sagiv, J.D.Ullman: 'Equivalence of relational database schemes', in: SIAM Journ. of Computing, 10, 1981, pp. 352 - 370

[CFP] M.Casanova, R.Fagin, C.H.Papadimitiou: 'Inclusion dependencies and their interaction with functional dependencies', in: ACM Symp. on Princ. of Database Systems, 1, 1982, pp. 171 - 176

[CH] A.K.Chandra, D.Harel: 'Computable queries for relational data bases', in: Journ. of Computer and System Sciences, 21, 1980, pp. 156 - 178

[CLM] A.K.Chandra, H.R.Lewis, J.A.Makowsky: 'Embedded implicational dependencies and their inference problem', in: ACM Symp. on Theory of Computing, 13, 1981, pp. 342 - 354

[Codd] E.F.Codd: 'Further normalizations on data base relational model', in: Data Base Systems (R.Rustin ed.), Prentice-Hall, Englewood Cliffs, 1972, pp. 33 - 64

[CV] M.A.Casanova, V.M.P.Vidal: 'Towards a sound view integration methodology', in: ACM Symp. on Princ. of Database Systems, 2, 1983, pp. 36 - 47

[DG] B.Dreben, W.D.Goldfarb: 'The decision problem: solvable classes of quantificational formulas', Addison-Wesley Publishing Company, Reading, 1979

[FV] R.Fagin, M.Y.Vardi: 'The theory of data dependencies - a survey', IBM Research Report 4321, San Jose, 1984

[Gee] W.C.Mc Gee: 'A contribution to the study of data equivalence', in: Database Management (J.W.Klimbie etc. ed.), Cargese, Amsterdam, 1974, pp. 123 - 148

[GJ] J.Grant, B.E.Jacobs: 'On the family of generalized dependency constraints', in: Journ. of the ACM, 29, 1982, pp. 986 - 997

[GM] J.Graham, A.O.Mendelzon: 'Strong equivalence of relational expressions under dependencies', in: Information Processing Letters, 14, 1982, pp. 57 - 62

[GMN] H.Gallaire, J.Minker, J.M.Nicolas: 'Logic and databases: a deductive approach', in: ACM Computing Surveys, 16, 1984, pp. 153 - 185

[GV] M.H.Graham, M.Y.Vardi: 'On the complexity and axiomatizability of consistent database states', in: Proc. ACM Symp. on Princ. of Database Systems, 3, 1984, pp. 281 - 289

[Hull] R.Hull: 'Relative information capacity of simple relational database schemata', Techn. Rep. 84-300, Comp. Science Department, Univ. South. Calif., Los Angeles, 1984

[Hull] R.Hull: 'Relative information capacity of simple relational database schemata',in: Proc. ACM Symp. on Princ. of Database Systems, 3, 1984, pp. 97 - 109

[IL1] T.Imielinski, W.Lipski: 'A technique for translating states between database schemata', in: ACM Int. Conf. on Management of data, 1982, pp. 61 - 68

[IL2] T.Imielinski, W.Lipski: 'On the undecidability of equivalence problems for relational expressions', in: Advances in Data Base Theory, 2, 1985, pp. 393 - 409

[IS] T.Imielinski, N.Spyratos: 'On lossless transformation of database states not necessarily satisfying universal instance assumption', in: Proc. ACM Symp. on Princ. of Database Systems, 3, 1984, pp. 258 - 265

[Koba] I.Kobayashi: 'Losslessness and semantic correctness of database scheme transformation: another look of schema equivalence', in: Information Systems, 11, 1986, pp. 41 - 59

[KK] P.Kandzia, H.J.Klein: 'On the equivalence of relational data bases in connection with normalization', Techn. Rep. 7901, Univ. Kiel, 1979

[JAK] B.E.Jacobs, A.R.Aronson, A.C.Klug: 'On interpretations of relational languages and solutions to the implied constraint problem', in: ACM Transactions on Database Systems, 7, 1982, pp. 291 - 315

[KCV] P.C.Kanellakis, S.S.Cosmodakis, M.Y.Vardi: 'Unary inclusion dependencies have polynomial time inference problem', in: Proc. ACM Symp. Theory of Computing, 15, 1983, pp. 264 - 277

[Klug] A.Klug: 'Calculating constraints on relational expressions', in: ACM Transactions on Database Systems, 5, 1980, pp. 260 - 290

[Mitch] J.C.Mitchell: 'The implication problem for functional and inclusion dependencies', in: Information and Control, 56, 1983, pp. 154 - 173

[Reiter] R.Reiter: 'Equality and domain closure on first - order databases', in: Journ. of the ACM, 27, 1980, pp. 235 - 249

[Riss] J.Rissanen: 'On the equivalence of database schemes', in: Proc. ACM Symp. Princ. of Database Systems, 1, 1982, pp. 23 - 26

[Solo] M.K.Solomon: 'Undecidability of the equivalence problem for relational expressions', in: Bell Lab. Memo

[Sier] W.Sierpinski: 'Cardinal and ordinal numbers', PWN Polish Scientific Publishers, Warschau, 1965

[SY] Y.Sagiv, M.Yannakakis: 'Equivalence among relational expressions', in: Proc. ACM Int. Conf.on Very Large Data Bases, 4, 1978, pp. 535 - 548

[Ullm] J.D.Ullman: 'Principle of Database Systems', Computer Science Press, Rockville, 1982

Проблема эквивалентности в схемах реляционных базах данных

Й. Бискуп, У. Реш

Резюме

Авторы используют отображения между схемами для определения эквивалентности, и так самые схемы играют роль параметров. Одним из результатов этого подхода есть то, что существует только одно естественное понятие эквивалентности. Они изучают также разрешимость или неразрешимость этого понятия, а также описывают отображения как реляционную алгебру без дифференции.

EKVIVALENCIA PROBLÉMA A RELÁCIÓS ADATBÁZIS SÉMÁKBAN

J. Biskup, U. Räsch

Összefoglaló

Az adatbázis sémák előfordulásai közötti leképezéseket fel lehet használni az ekvivalencia különböző fokainak definiálására. Ez lehetővé teszi, hogy maga a séma paraméterként fogható fel. Az ekvivalenciák összehasonlításai azt eredményezték, hogy csak egy természetes ekvivalencia-fogalom létezik. Különböző esetekben a szerzők bebizonyitják ennek az ekvivalencia-fogalomnak az eldönthetőségét ill. eldönthetetlenségét. Emellett a szerzők a leképezéseket különbség nélküli reláció-algebrák segitségével is jellemezték.

MTA SZTAKI Közlemények 36/1987 pp. 79-92

ON PREDICTIVE DECONVOLUTION OF LONG-RUN STATIONARY TIME SERIES

PHAN DANG CAU

Computer and Automation Institute
Hungarian Academy of Sciences
Budapest, Hungary

ABSTRACT

Robinson's statistical minimum-delay model / or the method of predictive deconvolution / has been effectively used in seismic prospecting for oil and gas. It is used to eliminate multiple reflections from surface layers and reverberations in water layer. However, in our opinion, this model is not clear in some respects.

In this paper we try to give a new interpretation and more general condition for this model, which are possibly more suitable to practice. We also point out that, with the new conditions, the computation process based on observations is just the same as in the case of Robinson's model. In other words, the Robinson's assumptions contain some simplifications, however his computation gives practically correct results.

We also give examples for the predictive deconvolution of the new process.

§.1. ROBINSON'S MODEL

In order to fix ideas, let us consider a specific physical situation, namely the problem of seismic exploration for oil in the earth's sedimentary strata. The source is an explosion or another form of energy, which is introduced into the ground at the surface. The reflection response x_n is the seismic reflection record / time series / which is digitally recorded at the surface. The reflection coefficient sequence ε_n is a digitized representation of the reflectivity of the earth as a function of depth. More exactly speaking, ε_n is the reflection coefficient of the interface n, where the travel time of the input signal in going to the interface n is $\frac{n}{2}$. As a result, knowledge of the sequences for various geographic locations on the surface allows the seismic interpreter to make contour maps of the earth's sedimentary structure at depth.

By certain assumptions / see [8] , p.457 / Robinson introduced the following equation:

$$x_n + a_1 x_{n-1} + a_2 x_{n-2} + \cdots a_p x_{n-p} = \varepsilon_n \qquad n = 0, 1, 2, \ldots \tag{1}$$

where the a_i-s are unknown deterministic values, which depend on the geological structure of the observed area.

In the case of a noise appearing, the reflection response has the form

$$y_n = x_n + v_n \tag{2}$$

where v_n is a noise. Here we suppose the noise is eliminated. The predictive deconvolution problem is to compute the ε_n-s from the x_n-s. However, (1) implies a system of equations having more unknown variables than the number of equations, so it is impossible to find the ε_n-s.

Robinson proposed the statistical method as follows:

Although the model (1) is deterministic, i.e. there is no random variable in it, but the sequence ε_n may be considered as a realization of a random white noise, i.e.

$$E\varepsilon_n = 0 \qquad E\varepsilon_n \varepsilon_s = \begin{cases} \sigma^2 & n = s \\ 0 & n \neq s \end{cases} \tag{3}$$

If we suppose further that

$$1 + a_1 z + a_2 z^2 + \ldots + a_p z^p \neq 0 \qquad \text{for} \quad |z| \leq 1 \tag{4}$$

then x_n is a stationary autoregressive process. The coefficients $a_1, a_2, \ldots, a_p$ then can be estimated from the observations x_n-s, and the ε_n-s are estimated by

$$\hat{\varepsilon}_n = x_n + \hat{a}_1 x_{n-1} + \cdots + \hat{a}_p x_{n-p} \tag{5}$$

§.2. SOME REMARKS ON ROBINSON'S MODEL

In our opinion, Robinson's model is not clear in some respects:
/a/ In the equation (1) the ε_n-s are deterministic values, so we can consider them as special random variables such that

$$E\varepsilon_n = \varepsilon_n \quad , \qquad var\,\varepsilon_n = 0$$

which contradicts the assumption (3) .
/b/ In practice the interfaces are not so arranged as in our assumptions, i.e. the travel time of the input signal in going to some interface is not always equal to $\frac{n}{2}$, where n is some positive integer, but may be an arbitrary real value t. Thus we can not consider the ε_n-s as exact reflection coefficients.
/c/ We recourse to irregularity of the sequence ε_n to make information for the earth's sedimentary structure at depth. For example, if $\varepsilon_n = 0$, we think that perhaps there is no interface at depth n, if $\varepsilon_n \approx 1$, we think it may be an interface between oil and gas strata... The assumption that the ε_n-s have the same mean and variance seems not always suitable to the practice.
/d/ In his model Robinson supposed only that the sequence ε_n is uncorrelated. / see [6] /. However, by the following example we want to show that in practice , when we have to estimate the correlation function from the sample time series, this assumption is not always sufficient for getting good estimates.

<u>Example 1</u>: Let our model be

$$x_n + a x_{n-1} = u_n \qquad , n = 1,2,3,\dots$$

where $|a|<1$, $u_n = \cos nW$ and W is an uniformly distributed random variable on $[0,2\pi]$. Then

$$Eu_n = 0 \qquad Eu_n u_s = \begin{cases} \frac{1}{2} & n = s \\ 0 & n \neq s \end{cases}$$

Thus the sequence u_n is a white noise in wide sense.
As usual, then a is estimated by

$$\hat{a} = \frac{-\frac{1}{N}\sum_{n=1}^{N} x_{n+1}x_n}{\frac{1}{N}\sum_{n=1}^{N} x_n^2}$$

Now suppose $a = 0$ then

$$\hat{a} = -\cos W - \frac{[\cos(2N+1)W - \cos W]\sin W}{\sin(2N+1)W + (2N-1)\sin W}$$

from which we can see that

$$\lim_{N\to\infty} \hat{a} = -\cos W \qquad \text{a.s.}$$

Thus the estimate $\hat{a}$ is always a random variable, its limit is also such a random variable, which has not any connection with the true value a . Hence we can not say that $\hat{a}$ is a good estimate of a.

§.3. THE MODIFIED MODEL

In order to modify Robinson's model so that it be more suitable to practice, let us firstly consider the simplest case:

Suppose after explosion the input signal $f(t)$ propagates to the earth's crust. When acting an interface having reflection coefficient ε it reflects to the surface with reflected wave $g(t) = \varepsilon f(t)$. Since the elastic wave $f(t)$ represents the motion of particle about its equilibrium point, $f(t)$ always has a damped sinusoidal form / in the case of explosion it is relatively narrow with great frequency /.Now let us consider some observed value u on $g(t)$. In geophysics the arrival time of a reflected wave is usually considered as a uniformly distributed random variable / i.e. we do not know exactly when the reflected wave appears /. Thus the observed value u can also be considered as a random variable $u = g(\tau)$ where τ is a uniformly distributed random variable on some interval $[a, b]$. We have

$$Eu = \frac{1}{b-a}\int_a^b \varepsilon f(t)\,dt \approx 0 \qquad \text{/ cf. Riemann lemma /}$$

$$Eu^2 = \frac{\varepsilon^2}{b-a}\int_a^b f^2(t)\,dt = C\varepsilon^2 \tag{6}$$

Or more exactly speaking, Eu is negligible compared with Eu^2 . although ε is some fixed value / even in the case $\varepsilon > 0$ /, the measured value u may be arbitrary value in $[-\varepsilon, \varepsilon]$. Hence u can not be considered as an approximation of ε .

By the above reason, we propose to modify (1) by introducing the new model:

$$x_n + a_1 x_{n-1} + \cdots + a_p x_{n-p} = u_n \quad n = 0,1,2,\cdots \tag{7}$$

Where $Eu_n = 0$, $Eu_n^2 = C\varepsilon_n^2$

For the reflection coefficients are different, Eu_n^2 can not be constant. However, by estimating the reflection coefficients from oil wells, White and Obriend /1974/ of British Petroleum, Schoenberger and Levin /1974/ of Exxon saw that the reflection coefficient sequence ε_n is like the realization of a white noise, i.e. for N large enough

$$\frac{1}{N}\sum_{n=0}^{N-1} \varepsilon_n^2 = \gamma^2 > 0 \tag{8}$$

$$\frac{1}{N}\sum_{n=0}^{N-1} \varepsilon_{n+s}\varepsilon_n \approx 0 \qquad n = 1,2,\ldots$$

/ see [8] , p.490 /. By (6) and (8) we have

$$\frac{1}{N}\sum_{n=0}^{N-1} Eu_n^2 = C\frac{1}{N}\sum_{n=0}^{N-1}\varepsilon_n^2 \approx C\gamma^2 = \sigma^2 > 0 \tag{9}$$

We suppose that in the case of a complicated geological phenomenon the variables u_n-s are independent.

Briefly speaking, we suppose that the reflection response satisfies (7) and the following three conditions:

/a/ The variables $u_0, u_1, u_2, \ldots$ are independent with mean 0 and

$$E|u_n|^{2+\varepsilon_0} < K < \infty \qquad \text{for some K and} \quad \varepsilon_0 > 0 \tag{10}$$

/b/ $$\lim_{N\to\infty} \frac{1}{N}\sum_{n=0}^{N-1} Eu_n^2 = \sigma^2 > 0 \tag{11}$$

/c/ $$1 + a_1 z + \cdots + a_p z^p \neq 0 \qquad \text{for} \quad |z| \leq 1 \tag{12}$$

We would like to remark that the condition (10) is obvious because the variables u_n-s are uniformly bounded. For the validity of the condition (12) the reader can see in [5] and [6] .

§.4. PREDICTIVE DECONVOLUTION OF LONG-RUN STATIONARY AUTOREGRESSIVE PROCESS

Definition 1: We call the process x_n satisfying (7) , (10) , (11) and (12) a long-run stationary autoregressive process.

This notion does not occure in the standard literature on time series analysis.

Definition 2: We call the process y_n a stationary autoregressive process corresponding to the above defined long-run stationary process x_n if y_n satisfies

$$y_n + a_1 y_{n-1} + \dots + a_p y_{n-p} = v_n \qquad n = \dots -1, 0, 1, 2, \dots \tag{13}$$

where v_n is a white noise with variance σ^2.

Theorem: Let x_n be a long-run stationary process satisfying (7), (10), (11), (12) then for $s = \dots -1, 0, 1, 2, \dots$ there exist the limits

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x_{n+s} x_n = \varphi_s \qquad \text{a.s.} \tag{14}$$

where φ_s is the correlation function of the corresponding stationary process y_n.

Proof: By (12) we can take the reciprocal $B(z)$ of the Z-transform A(

$$B(z) = \frac{1}{1 + a_1 z + \dots + a_p z^p} = 1 + b_1 z + b_2 z^2 + \dots$$

and the process x_n can be written in the form

$$x_n = \sum_{s=0}^{\infty} b_s u_{n-s} \qquad \text{where } u_n = 0 \text{ for } n < 0$$

By (10) $Eu_n^2 < d$ for some $d > 0$.

Let

$$\xi_n = u_n^2 - Eu_n^2 \quad , \quad \delta_0 = \frac{\varepsilon_0}{2}$$

Using Minkowski's inequality, we have

$$\left(E|\xi_n|^{1+\delta_0}\right)^{\frac{1}{1+\delta_0}} = \left(E|u_n^2 - Eu_n^2|^{1+\delta_0}\right)^{\frac{1}{1+\delta_0}} \leq$$

$$\leq \left(E|u_n|^{2+\varepsilon_0}\right)^{\frac{1}{1+\delta_0}} + Eu_n^2 < K^{\frac{1}{1+\delta_0}} + d$$

from which

$$E|\xi_n|^{1+\delta_0} < \left(K^{\frac{1}{1+\delta_0}} + d\right)^{1+\delta_0} < \infty$$

The sequence ξ_n satisfies the conditions of Markov's theorem

/ see [2] , p.287 / therefore

$$\underset{N\to\infty}{\mathrm{Plim}} \frac{1}{N} \sum_{n=0}^{N-1} \xi_n = 0$$

Thus we have

$$\underset{N\to\infty}{\mathrm{Plim}} \frac{1}{N} \sum_{n=0}^{N-1} (u_n^2 - Eu_n^2) = \underset{N\to\infty}{\mathrm{Plim}} \left(\frac{1}{N} \sum_{n=0}^{N-1} u_n^2 - \frac{1}{N} \sum_{n=0}^{N-1} Eu_n^2 \right) = 0 \tag{15}$$

By (11) and (15)

$$\underset{N\to\infty}{\mathrm{Plim}} \frac{1}{N} \sum_{n=0}^{N-1} u_n^2 = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} Eu_n^2 = \sigma^2$$

Since the variables u_n^2-s are independent, by the equivalence theorem / see [3] , p.263 / we have

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} u_n^2 = \sigma^2 \qquad \text{a.s.} \tag{16}$$

For $s = 1, 2, 3, \ldots$ we can write

$$\frac{1}{N} \sum_{n=0}^{N-1} u_n u_{n+s} = \sum_{\ell=0}^{s} \frac{1}{N} \sum_{\tau=0}^{\left[\frac{N-\ell-1}{s+1}\right]} u_{\tau s+\ell+\tau} u_{(\tau+1)s+\ell+\tau} \tag{17}$$

Now consider

$$E\left(\frac{1}{N} \sum_{\tau=0}^{M} \eta_\tau \right)^2 = \frac{1}{N^2} \sum_{\tau=0}^{M} Eu_{\tau s+\ell+\tau}^2 \, Eu_{(\tau+1)s+\ell+\tau}^2 \leq \frac{Nd^2}{N^2} = \frac{d^2}{N} \to 0$$

where

$$M = \left[\frac{N-\ell-1}{s+1} \right] , \qquad \eta_\tau = u_{\tau s+\ell+\tau} u_{(\tau+1)s+\ell+\tau}$$

Here we have used the independence of the variables u_n-s. Using Tchebychef's inequality we get

$$\underset{N\to\infty}{\mathrm{Plim}} \frac{1}{N} \sum_{\tau=0}^{M} \eta_\tau = 0$$

We can see the variables η_τ-s are independent, using the equivalence theorem again we get

$$\lim_{N\to\infty} \frac{1}{N} \sum_{\tau=0}^{M} \eta_\tau = 0 \qquad \text{a.s.}$$

From which and (17) we have

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} u_n u_{n+s} = 0 \qquad \text{a.s.} \tag{18}$$

From (17) and (18) we have

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x_{n+s} x_n = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \sum_{l=0}^{\infty} \sum_{r=0}^{\infty} b_l b_r u_{n+s-l} u_{n-r} =$$

$$= \sum_{l=0}^{\infty} \sum_{r=0}^{\infty} b_l b_r \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} u_{n+s-l} u_{n-r} = \sigma^2 \sum_{r=0}^{\infty} b_{r+s} b_r = E y_{n+s} y_n = \varphi_s \quad \text{a.s.}$$

Thus the proof is complete.

Remark: If we know φ_s, $s=0,1,2,\dots,p$ then $a_1, a_2, \dots, a_p$ are determined by Yule-Walker-type equations

$$\begin{pmatrix} \varphi_0 & \varphi_1 & \cdots & \varphi_{p-1} \\ \varphi_1 & \varphi_0 & \cdots & \varphi_{p-2} \\ \vdots & & & \\ \varphi_{p-1} & \varphi_{p-2} & \cdots & \varphi_0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = - \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_p \end{pmatrix}$$

Therefore the u_n-s are determined by

$$u_n = x_n + a_1 x_{n-1} + \cdots + a_p x_{n-p} \tag{19}$$

In practice we estimate φ_s by

$$r_s = \frac{1}{N} \sum_{n=0}^{N-1} x_{n+s} x_n$$

and then $a_1, a_2, \dots, a_p$ are estimated by

$$R\hat{a} = -r \tag{20}$$

where

$$\hat{a} = (\hat{a}_1, \hat{a}_2, \dots, \hat{a}_p)'$$

$$r = (r_1, r_2, \dots, r_p)'$$

$$R = \begin{pmatrix} r_0 & r_1 & \cdots & r_{p-1} \\ r_1 & r_0 & \cdots & r_{p-2} \\ \vdots & & & \\ r_{p-1} & r_{p-2} & \cdots & r_0 \end{pmatrix}$$

Therefore the estimate $\hat{u}_n$ of u_n is obtained by

$$\hat{u}_n = x_n + \hat{a}_1 x_{n-1} + \cdots + \hat{a}_p x_{n-p}$$

and we have

$$\lim_{N \to \infty} \hat{u}_n = u_n \qquad \text{a.s.}$$

§.5. ON THE LIMITING DISTRIBUTION OF THE ESTIMATES

It is well known, that if x_n is a stationary autoregressive process then $a_\tau, \tau=1,\dots p$ has a limiting normal distribution. However this is not always true for a long-run stationary process. For the sake of simplicity here we illustrate this by the following example:

Example 2: Let x_n satisfy

$$x_n + a x_{n-1} = u_n \qquad |a| < 1 \quad , n = 0, 1, 2, \dots$$

where the u_n-s are independent with $Eu_n = 0$ and

$$Eu_n^2 = \begin{cases} \sigma^2 & n \text{ even} \\ 0 & n \text{ odd} \end{cases}$$

Then by (20) a is estimated by

$$\hat{a} = \left(\frac{1}{N} \sum_{n=0}^{N-1} x_n^2 \right)^{-1} \left(-\frac{1}{N} \sum_{n=0}^{N-1} x_n x_{n-1} \right)$$

Suppose $a = 0$, then $\hat{a} \equiv 0$, i.e. the distribution of $\hat{a}$ is degenerated.

Thus, although the estimation problem is the same for both processes, but we must be careful when we want to test hypotheses for a long-run stationary process.

§.6. SOME NUMERICAL EXAMPLES

Example 3: Let x_n be a long-run stationary process defined by

$$x_n + a x_{n-1} = u_n \qquad a = 0.5,\ n = 0,1,2,\dots 499 \tag{21}$$

$$u_n = \begin{cases} 4W_n & n = 10k+l,\ l = 0,1,2,3 \\ W_n & n = 10k+l,\ l = 4,5,6,7 \\ 3W_n \quad \text{if } v_n \leq 0.5 & n = 10k+l,\ l = 8,9 \\ 0 \quad \text{if } v_n > 0.5 & \end{cases} \qquad k = 0,1,2,\dots,49 \tag{22}$$

where the W_n-s are independently, normally distributed random variables with mean 0 and variance 1, the v_n-s are independent uniformly distributed r.v.-s on $[0,1]$ and independent of W_n. Thus

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} E u_n^2 = 8.6$$

From randomly generated u_n-s we can compute the x_n-s / see Table 1, here only the first 60 observations are printed /

Table 1: Long-run stationary process defined by (21) and (22)

n	x_n	n	x_n	n	x_n	n	x_n
0	2.9643	15	-1.2439	30	4.9835	45	1.1672
1	-1.6459	16	0.4143	31	0.3325	46	-0.5887
2	-1.2496	17	1.9409	32	0.2347	47	-0.9002
3	-6.9688	18	-0.9705	33	-0.0831	48	0.4501
4	3.6511	19	0.4852	34	0.3186	49	-0.2251
5	-2.1529	20	-1.2999	35	0.1936	50	-0.5336
6	1.0097	21	1.6766	36	-1.2565	51	2.0615
7	2.0500	22	-1.6783	37	1.1738	52	-1.8660
8	-1.0250	23	-2.9980	38	-0.5869	53	-10.3786
9	1.0432	24	1.4378	39	2.3421	54	3.6968
10	-4.2701	25	-1.4391	40	-0.5952	55	-3.5001
11	-0.1656	26	1.7844	41	-0.0013	56	2.3429
12	3.4760	27	-1.6509	42	-4.9674	57	-1.7414
13	-5.7481	28	0.8254	43	7.0509	58	2.7707
14	3.8189	29	-1.7480	44	-3.6282	59	-4.9872

By (20) we get the estimate $\hat{a} = 0.4680$.
Hence the estimates $\hat{u}_n$-s are also obtained / In table 2 only the first 30 values are printed /

Table 2: Long-run white noise defined by (22) and its estimate

n	long-run white noise	estimated long-run white noise	n	long-run white noise	estimated long-run white noise
0	2.9643	2.9643	75	0.6655	0.5432
1	-0.1637	-0.2587	16	-0.2077	-0.1679
2	-2.0725	-2.0198	17	2.1481	2.1348
3	-7.5936	-7.5536	18	0.0	-0.0622
4	0.1667	0.3900	19	0.0	0.0311
5	-0.3274	-0.4444	20	-1.0573	-1.0729
6	-0.0668	0.0022	21	1.0267	1.0683
7	2.5548	2.5224	22	-0.8399	-0.8937
8	0.0	-0.0657	23	-3.8372	-3.7834
9	0.5307	0.5636	24	-0.0612	0.0348
10	-3.7485	-3.7819	25	-0.7202	-0.7662
11	-2.3006	-2.1638	26	1.0648	1.1109
12	3.3933	3.3986	27	-0.7587	-0.8159
13	-4.0101	-4.1215	28	0.0	0.0529
14	0.9449	1.1291	29	-1.3358	-1.3623

From table 1 and table 2 we can see that the condition $u_8 = 0$ / in practice it means that $\varepsilon_8 = 0$, i.e. there is no interface at $n = 8$ / can not be recognized on the reflection response x_n , because $x_8 = -1.025$. However after the predictive deconvolution we get $\hat{u}_8 = -0.0657$ which is approximately 0.

Example 4: Let x_n be a long-run stationary process defined by

$$x_n + a_1 x_{n-1} + a_2 x_{n-2} = u_n \qquad n = 0, 1, 2, \dots, 499 \qquad (23)$$

where $a_1 = -0.7$, $a_2 = 0.49$ and

$$u_n = \begin{cases} 4\vartheta_n & n = 6k+\ell, \ell=0,1 \\ \vartheta_n & n = 6k+\ell, \ell=2,3 \quad k = 0,1,\dots,83 \\ 0 & n = 6k+\ell, \ell=4,5 \end{cases} \tag{24}$$

The ϑ_n -s are independently , uniformly distributed random variables on $[-0.5,0.5]$. Thus

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} Eu_n^2 = 0.47$$

By (20) we get the estimates $\hat{a}_1=-0.6268$, $\hat{a}_2= 0.4475$. In table 3 the first 30 values of u_n and its estimate are printed.

Table 3: Long-run white noise defined by (24) and its estimate

n	long-run white noise	estimated long-run white noise	n	long-run white noise	estimated long-run white noise
0	-1.0830	-1.0830	15	-0.2498	-0.2138
1	1.0675	0.9882	16	0.0	0.0467
2	0.1832	0.2518	17	0.0	0.0257
3	0.0092	0.0642	18	0.0507	0.0458
4	0.0	-0.0023	19	1.7568	1.7445
5	0.0	-0.0289	20	0.1240	0.2443
6	1.3899	1.3707	21	-0.0776	-0.0529
7	-0.5841	-0.4816	22	0.0	-0.0526
8	-0.1011	-0.1218	23	0.0	-0.0456
9	-0.0429	-0.0902	24	-0.1231	-0.1293
10	0.0	-0.0218	25	-1.7805	-1.7715
11	0.0	0.0097	26	0.0198	-0.0960
12	-1.3358	-1.3183	27	-0.1032	-0.1116
13	-1.0959	-1.1862	28	0.0	0.0425
14	0.1625	0.0672	29	0.0	0.0382

Acknowledgement: The author is very grateful to Prof.Dr. Nguyen xuan Loc, Dr. Nguy . Ho and Dr. András Krámli for their help, valuable comments and suggestions.

The author would like to take this opportunity to express deep gratitude to his teacher, Dr. József Tomkó, for his guidance and encouragement during the preparation of this paper.

REFERENCES

/1/ Anderson,T.W /1972/ Statistical Analysis of Time series John Wiley and sons.

/2/ Arató,M., Benczur,A., Krámli,A., Pergel,J.,/1974/ Statistical problems of elementary Gaussian process,I. Stochastic process / MTA SZTAKI Tanulmányok 22/1974 /

/3/ Loève,Michael/1977/ Probability. D.van Nostrand Co.,Inc.,New York.

/4/ Meskó,A.,/1984/ Digital filtering applications in Geophysical Exploration for oil. Akadémiai Kiadó, Budapest.

/5/ Rényi Alfréd /1973/ Valószinüségszámitás. Tankönyvkiadó,Budapest.

/6/ Robinson,E.A /1967/ Predictive deconvolution of time series with application to seismic exploration, Geophysics, Vol.XXXII, N -3/ June,1967,pp.418-84/

/7/ Robinson,E.A and Treitel,Seven /1980/ Seismic signal processing Englewood cliffs,N.J: Prentice-Hall

/8/ Robinson,E.A /1981/ Time series analysis and applications, Houston, Goose Pond Press.

Об обратном фильтре асимптотически стационарных временных рядов

Фан Данг Кау

Резюме

Статистическая модель Робинзона была эффективно использована в области сейсмического исследования ресурсов нефти и газа. По нашему мнению эта модель в некотором смысле не ясна.

В этой работе обобщая условия оригинальной модели дается новый подход к проблеме. Мы показываем, что ход численного счета, который требуется более общим условиям совпадает с оригинальными формулами Робинзона. В конце работы результаты иллюстрируются численными экспериментами.

ASZIMPTOTIKUS STATICONÁRIUS IDŐSOROK PREDIKTIV DEKONVOLUCIÓJÁRÓL

Phan Dang Cau

Összefoglaló

Robinson statisztikai modelljét hatásosan használják a gáz és olaj-kutatásokban. Azonban, véleményünk szerint ez a modell néhány szempontból nem világos. A jelen dolgozatban a Robinson modell egy uj értelmezését adjuk. Bebizonyitjuk, hogy az uj feltételek mellett a megfigyeléseken alapuló számitási folyamat ugyanugy történik, mint a Robinson modell esetén. Illusztrációként adunk néhány numerikus példát.

MTA SZTAKI Közlemények 36/1987 pp. 93-97

ON THE DENSITY OF TRANSLATES OF A DOMAIN

A. HEPPES

Computer and Automation Institute
Hungarian Academy of Sciences
Budapest, Hungary

Let w be a domain in the Euclidean plane. We shall denote the density of the densest packing of translates of w by d(w) and the density of the densest lattice packing of translates of w by d'(w).

It has been shown ([1], [2], [3]) that if w is convex then

$$d(w) = d'(w). \qquad (1)$$

In a recent paper L. Fejes Tóth [4] started an interesting field of research trying to extend the validity of (1) to more general domains. He has proved that if w is the union of two intersecting equal circles than equation (1) is still valid. The range of validity of this property has been curbed by a construction of A. Bezdek and G. Kertész [5]. They have constructed a domain consisting of 5 convex domains that can be arranged to have higher density if you do not require the packing to be latticelike. Fejes Tóth's conjecture [1] is that this can not be done with a domain that is the union of two convex domains with a point in commmon. The analoguous question has been raised for starlike domains as well.

In the present paper we are going to give a construction for a domain u with the following properties:

(i) u is the union of three convex domains
(ii) u is starlike,
(iii) d(u) > d'(u), i.e. the densest packing of u is not latticelike.

To describe the domain u and to show its properties we shall use the 2-dimensional coordinate system. In what follows A(x) will denote the area of x, and the sum of a domain and a vector denotes the translate of the domain by that vector. The three conponents that we use to construct our domain u are two rhombs R1 and R2 and a hexagon H. We define them by listing their corners as follows (Fig.1) :

R1 (a, -a), (a, 1-a), (-1+a, 2-a), (-1+a, 1-a)
R2 (a, a), (a,-1+a), (-1+a,-2+a), (-1+a,-1+a)
H (0, 0), (1, 1), (1+L, 1), (2+L, 0), (1+L,-1), (1,-1)

Here L denotes a sufficiently large and a > 0 denotes a sufficiently small number.

This work was supported by the National Scientific Research Fund No. 1066/1985

The union u of R1, R2 and H is clearly starlike with respect to any point of the triangle (0,0), (a,a), (-a,-a) thus u shares properties (i) and (ii).

Consider now the translate u1 = u + (1,2) and the union v of u and u1. On the one hand u and u1 have no interior point in common, on the other hand the vectors (0, 4) and (L+3, 2) define a latticelike arrangement of v that is a packing (Fig.2). Thus we have a packing of translates of u of density

A(u)/(2*L+6).

Although we are convinced that the best lattice packing is generated by the vectors (1, 2) and (L+3-a,-1+a) (Fig.3) we need not prove that to reach our goal.All we are left to show is that any lattice-packing of u has a smaller density than A(u)/(2*L+6).

Let us consider a lattice-packing of u. First we define the 'side strip' and the 'neck' of u. The side strip of u is a rhomb of area b*(L+1) given by its corners: (0, 1), (L+1, 1), (L+1-b, 1+b), (-b, 1+b), and the neck is a triangle given again by its corners (a, a), (a+b,a+b), (a, a+b); where b is a sufficiently small but positive number.

We distinguish two cases. First we assume that in the lattice-packing the hexagonal parts of the neighbouring domains are not close to each other, more precisely, we assume that the side strip of u does not contain a point of the hexagonal part of a translate. Then - considering that no more than a single rhombic part of the whole packing can have a point in common with the side strip of u, and that the area of that common part is certainly smaller than b, to each translate there belongs an uncovered part of area > b*L. Since A(u) = 2*L+4-2*a*a, to any prefixed a and b L can be chosen so that A(u)+b*L > 2*L+6.

In the other case the sides of certain pairs of hexagons are close than b. Of the logically symmetric two subcases we assume that the rhombic part of a translate u2 uf u enters the neck triangle of u Then u2 = u + (1+t1, 2+t2), where 0 <= t1 <= t2 < b. The domains u and u2 define a stripe of the lattice-packing, and the whole lattice is defined by two neighbouring stripes. Since the closest position of two such stripes is defined by the translation (L+3-a-t2, -1+t2), the area of the fundamental domain of the lattice of the densest such lattice packing is 2*L+7 +(L+3)*t2 -2*(a+t2) -(t2-t1) -t1*t2. For sutably chosen a and b this area is > 2*L+6.5, and this is what we wanted to show.

References:

[1] C. A. Rogers, The closest packing of convex two-dimensional domains. Acta Math. 86 (1951), 309-321.

[2] L.Fejes Tóth, Some packing and covering theorems. Acta Sci. Math. (Szeged) 12/A (1950), 62-67.

[3] L.Fejes Tóth, On the densest packing of convex discs. Mathematika 30 (1983), 1-3.

[4] L.Fejes Tóth, Densest packing of translates of the union of two circles. Discrete Comput. Geom. 1 (1986), 307-314.

[5] A. Bezdek and G. Kertész, Counter-examples to a packing problem of L. Fejes Tóth, Colloquia Mathematica Soc. János Bolyai 48. Intuitive Geometry, Siófok, 1985. 29-36

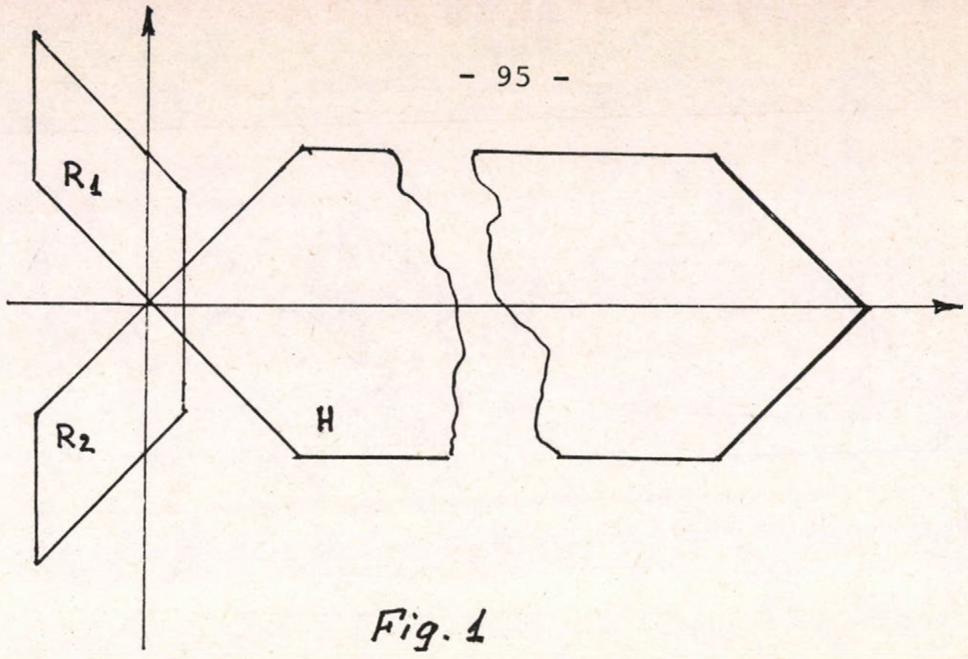


Fig. 1

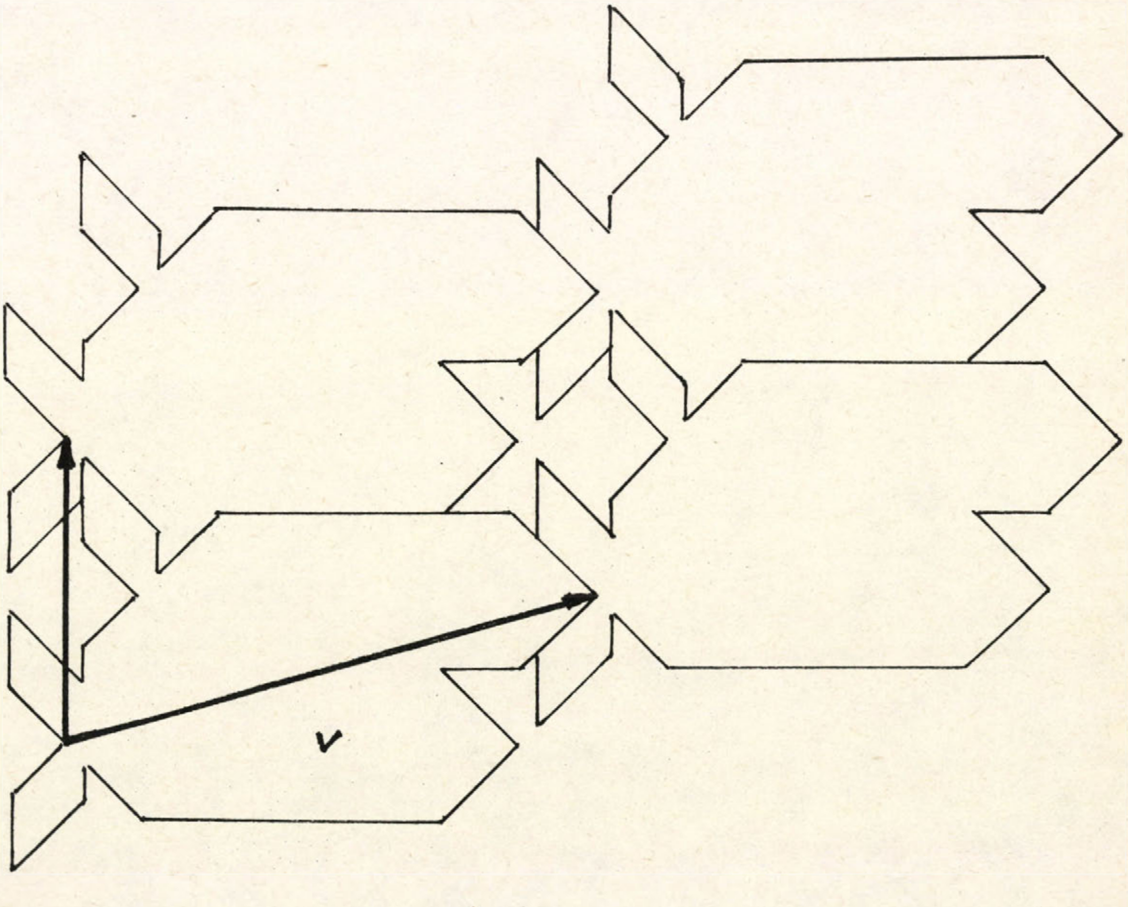


Fig. 2

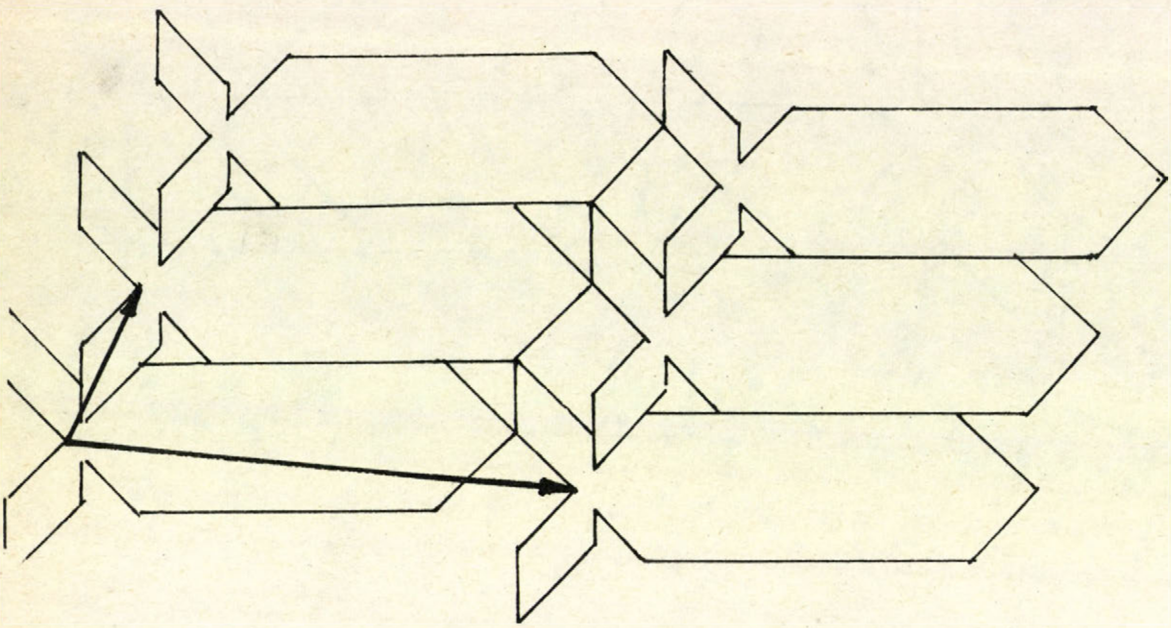


Fig. 3

Плотность трансляций одной области

А. Хеппеш

Резюме

Статья изучает выполнения плоскости трансляциями данной облисти, так что две области не имеют совместные внутренние точки. Исследуя одну проблему Л. Фейеш Тот-а, автор конструктивным образом доказывает, что: а/ существует область /связное объединение трех выпуклых областей/ такая, что трансляция дающая максимальную плотность не решеточная, и б/ существует звездочная область с такими же свойствами.

EGY TARTOMÁNY ELTOLÁSAINAK SÜRÜSÉGÉRŐL

Heppes A.

Összefoglaló

A szerző Fejes Tóth László által felvetett problémákat vizsgálva konstrukció utján bizonyitja, hogy a/ létezik olyan tartomány a sikon, amelyen 3 konvex tartomány /összefüggő/ egyesitése és amelynek legsürübb átfedés nélküli elrendezése nem rácsszerü, b/ létezik olyan csillagszerü tartomány a sikon, amelynek legsürübb átfedés nélküli elrendezése nem rácsszerü.

MTA SZTAKI Közlemények 36/1987 pp.99-103

LAN BASED INTEGRATED HOSPITAL INFORMATION SYSTEM

P. KERÉKFY, M. RUDA, Gy. ALEXICS, A. GÁL
K. KOVÁCS, T. LENGYEL, F. RÁKOCZI

Computer and Automation Institute
Hungarian Academy of Sciences
Budapest, Hungary

This lecture presents a complex hospital information system which is based on a distributed data management method. The hardware architecture is a local area network which is composed from simple 8-bit microcomputers. In the information system there are three main components: the patient admission subsystem, a complete clinical (chemical) laboratory workstation and patient treatment activities in the children's department. This latest subsystem has three typical workstations. One of these is an administrative workstation, connected to the integrated patient registration subsystem of the whole hospital. Another subsystem in the children's department contains properly the nursing administration activities: e.g. describing the parameters of health of the patients, storing and computing the parameters of examinations, etc. And finally a particular network program provides the data transfer between the chemical laboratory and the children's department.

The basic software tool applied in this hospital information system is the micro-SHIVA form and data manager. The micro-SHIVA form manager is an extendeed full-screen editor providing interface to the data manager. The end-user simply fill in fields on forms displayed on the screen to activate the data query and data input functions. The user-friendliness of the system is based on the visual tools of the micro-SHIVA and on the transparency of the local network.

In our hospital information system the microcomputers are connected to LANPBOX (local area network preocessor) cluster controllers. This local network system has the following characteristics:

- the topology of the system can be modified easily,

This work was supported by the National Scientific Research Fund No. 1066/1985

- the system is transparent, i.e. the user need not know anything about the topology and functions of the network,
- it allows interprocess communications,
- each remotely connected device (e.g. printer, disk) can be accessed as if it were locally connected
- it includes a gateway function which allows accessing public data networks or IBM compatible host computers from all stations within the network system,
- the data transportation speed is 1 megabit/sec,
- microcomputers and network processors are based on Z80 microprocessor, the operative storage capacity is 64/128 kbyte RAM,
- floppy disk storage (2 * 700 kbyte at each workstation),
- CP/M compatible operating system.

The nursing system

The Institute of the Cardiology not having utilized computers before, will be equipped with our local computer network system. Consequently, this system should be easy-to-use, effective and fast. It must not consume more time and labour than the traditional methods or disturb the normal way of life in hospital, and must be introduced gradually. For this reason, first a relatively small department (the children department) was be partially supported by the computer system.

The user friendliness is based not only on the visual tools of micro-SHIVA and the transparency of the network system, but the homogenity of the network is a source of the easy usage, too. The system should be homogeneous in the sense that the workstations are on equal rank i.e. any workstation can be a controlled element as well as a controlling element in the distributed information system. Transactions initiated at any of the workstations should bring about changes at other nodes, too. For example, discharge of a patient from the hospital results in a final report at the department, while in the central register and in the statistical subsystem it should produce a new record.

Another important matter in health-care systems comes from the private character of medical data. On personal computers professional secrecy can be assured by simple means. Micro-SHIVA assures both comfortable data processing and data security through utilization of forms. Any user can perform operations only on data that appear on forms that he has access to. While defining new forms he can use names of data that are known to him. Since names of data fields are never displayed for general users, this method assures data security. Only the system manager disposes of subsystems for database definition and modification.

In the nursing system there is a tool to handle coded and natural language information together. This is a special interpreter language (VOCAB - vocabulary manager). The "program" interpreted by VOCAB contains rules and decoding informations. It gets its coded and free text input from the user and performs checking procedures described by the rules. When the information is complete it can be transformed into natural language sentences. From the point of view of computer technics it is a coding/decoding system. From the medical point of view this is a system where coding needed for computer processing is based upon medical problems and "way of thinking" and from the other hand it provides a "human", natural language output. The VOCAB is based on the micro-SHIVA form and data manager system.

An application of the VOCAB is the medical state definition of the inpatient. At the time of filling the status data, the system controls the way of coding analising some logical connections and in case of mistakes it gives a feedback for the user. The next step is the forming of the coded information as a normal language text. The novelty of the VOCAB is mostly that the sentences are formed by sentence-panels on the basis of the joint analysis of certain codes being logically connected with one another, and so the laboured, "not human" character of the texts can be meaningfully eliminated.

This system was also capable to support the task of making medical diagnosis, simply by writing other control forms. The process of definition can be put by arranging the possible codes in a net structure. The nodes of the net can be given diagnosis names and it is possible to enter in the net at any node. In this way the user may find the code of the diagnosis step-by-step.

The laboratory system

This subsystem can work either as an independent workstation or as a subsystem in the complex hospital information system. The laboratory subsystem handles the measurment data of ambulatory clients and patients under clinical treatment. Special effort was made to support data access and information retrieval in a great variety. The data (measured and to be measured) are divided into working lists on the basis of laboratory experience. The system automatically compiles the lists of the required tests and takes care on abnormal values signaling the too low / too high masurment data. There are two ways to fill out the laboratory forms: can be filling a single patient form which involves only data of one patient or a labour resort list which involves groupped data. In addition to typing data from keyboard a special feature helps collecting data arriving from measuring automates. The diagnoses are printed in laboratory or sent by the system to the involved clinical department (the children department) via network. Custom statistics, quality control programs make the system complete.

Inpatient admission

The inpatient admission handles administrative data of patients. It shedules patients waiting for treatment, maintaines hospital bed registry and produces statistical data.

This system releases the departments from a significant percent of administrative duties since data collected by the admission can be transmitted to any department.

Beside treatment of the patients, it is necessary to supply the local, regional and national managing bodies with statistical data. These data are useful in scientific studies, e.g. morbidity statistics, study of effectivity of treatments. Statistical data are needed for planning food, medicine and drogs supplies, too. We can show that, for example a nation-wide complex data system is useful and fit for life only if the data to be processed are recorded and checked at their very source, i.e. in the hospitals (e.g. at the inpatient admission). Otherwise, a rather worthless mass of data is obtained only. And we know that management based on inadequate data is useless, even destructive.

Интегрированная информационная система для больниц, основанная на локальной сетевой системе

П. Керекфи, М. Руда, Дь. Алексич, А. Гал,
К. Ковач, Т. Лендел, Ф. Ракоци

Резюме

В статье описаны главные черты системы. Система состоит из сети простых 8 битовых микрокомпьютеров. В информационной системе три базисные компоненты: субсистема приема больных, субсистема лабораторных работ и субсистема в отделе госпиталя. Главным средством мат. обеспечения есть система управления базой данных micro-SHIVA.

LOKÁLIS HÁLÓZATON ALAPULÓ INTEGRÁLT KÓRHÁZI INFORMÁCIÓS RENDSZER

Kerékfy P., Ruda M., Alexics Gy., Gál A., Kovács K.,
Lengyel T., Rákóczi F.

Összefoglaló

A szerzők a rendszer alapvető tulajdonságait ismertetik. A rendszer hardverje egyszerü 8-bites mikro-számitógépeken alapuló hálózat. Az információs rendszernek három komponense van: beteg-felvételi alrendszer, klinikai laboratóriumi alrendszer, alrendszer az osztályon. A fő szoftver eszköz a micro-SHIVA ürlap-és adat-kezelő rendszer.

MTA SZTAKI Közlemények 36/1987 pp. 105-121

THE RELATION BETWEEN ANTIKEYS AND M-MINIMAL COVERS IN THE RELATION SCHEMES

PHAM THE QUE

Computer and Automation Institute
Hungarian Academy of Sciences
Budapest, Hungary

A b s t r a c t

In this paper, we introduce the notion of so-called M-minimal covers for relation scheme and prove some of its properties. Basing upon these properties, a necessary and sufficient condition under which a subset X of Ω is an antikey for a relation scheme is established when the set of all keys for the relation scheme was known.

This work was supported by the National Scientific Research Fund No. 1066/1985

§1. Definitions:

In this section we present some necessary definitions. Let $S = \langle \Omega , F \rangle$ be a relation shheme and

$\mathcal{K}_s = \{ K_1, K_2, K_3, \ldots, K_m \}$ be the set of all keys

for S

Let us denote:

$$H = \bigcup_{i=1}^{m} K_i = \{ a_1, a_2, \ldots, a_p \} \subseteq \Omega$$

$M = \{ 1,2,3,\ldots,m \}$ is set of all indexes for keys.

Recall that $K \subseteq \Omega$ is a key for S if:

a) $K^+ = \Omega$

b) $\overline{\exists} K' \subset K$ such that $(K')^+ = \Omega$.

The subset $K^{-1} \subset \Omega$ is called an antikey for S if:

a) $K \nsubseteq K^{-1} \quad \forall K \in \mathcal{K}_s$

b) $\forall X : (X \subseteq \Omega \,\&\, K^{-1} \subset X) \Rightarrow \exists K \in \mathcal{K}_s :$

$K \subseteq X$.

Let $\mathcal{K}_s^{-1}$ be the set of all antikeys for S .

<u>1.1</u> We construct the set I_j as follows:

$$\forall a_j \in H : \; I_j = \{ i \mid a_j \in K_i \, , \; i \leq m \}, \; j \leq p .$$

It is obvious that:

a) $I_j \subseteq M$ and $I_j \neq o$, $\forall j \leq p$

b) $M = \bigcup_{j=1}^{p} I_j = \{ 1,2,\ldots,m \}$.

Thus I_j is the set of all indexes for keys containing a_j . For any given $a_j \in H$, the set I_j is completely determined by a_j .

Let $\mathcal{I}_M = \{ I_1, I_2, \ldots, I_p \}$. Let $\mathcal{N} \subseteq \mathcal{I}_M$. The set $\mathcal{N}$ is said to be a M-minimal cover if $\mathcal{N}$

satisfies the following conditions:

a) $M = \bigcup_{I_j \in \mathcal{N}} I_j$

b) $\overline{\exists} \mathcal{N}' \subset \mathcal{N} : \bigcup_{I_j \in \mathcal{N}'} I_j = M$.

That means, if $\mathcal{N} \subseteq \mathcal{J}_M$ is a M-minimal cover then for all $\mathcal{N}' \subset \mathcal{N}$, we have $\bigcup_{I_j \in \mathcal{N}'} I_j \subset M$.

If $\mathcal{N} \subseteq \mathcal{J}_M$ only satisfies condition (a) , we say that $\mathcal{N}$ is a M-cover.

It is easy to see that $\mathcal{J}_M$ is a M-cover and contains at least one M-minimal cover.

<u>1.2</u> We can define the notion of M-minimal cover in an another way:

Given the set $\mathcal{K}_S$ for a relation scheme $S = \langle \Omega , F \rangle$, we can determine a matrix $\mathcal{M}(\mathcal{K}_S) = (\alpha_{ij})$ having p rows and m columns as follows:

$$\alpha_{ij} = \begin{cases} 1 & \text{if } a_i \in K_j \\ 0 & \text{otherwise} . \end{cases}$$

We call r_i , the i-th row of matrix $\mathcal{M}(\mathcal{K}_S)$ for every $i \leqslant p$, and then

$$\mathcal{M}(\mathcal{K}_S) = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_p \end{pmatrix}$$

Let us define:

$$r_i \leqslant r_j \iff \alpha_{il} \leqslant \alpha_{jl} \quad \text{for all } l \leq m.$$

It is obvious that

$$r_i \leqslant r_j \iff I_i \subseteq I_j .$$

We say that

$$\mathfrak{M}^{(k)} = \begin{pmatrix} r_{i_1} \\ r_{i_2} \\ \vdots \\ r_{i_k} \end{pmatrix}$$

is a submatrix of $\mathfrak{M}(\mathcal{K}_s)$ and the meaning of the following notations are obvious:

$$\mathfrak{M}^{(k)} \subseteq \mathfrak{M}(\mathcal{K}_s)\ ,\ r_{i_j} \in \mathfrak{M}^{(k)} \quad \forall i_j \leq i_k .$$

The row vector $C\left[\mathfrak{M}^{(k)}\right] = (l_1, l_2, \ldots, l_m)$ is called the characteristic vector of the submatrix $\mathfrak{M}^{(k)} \subseteq \mathfrak{M}(\mathcal{K}_s)$

if $l_j \in \{0,1\}$ and $l_j = 0 \iff \sum_{i=i_1}^{i_k} \alpha_{ij} = 0$, $1 \leq j \leq m$.

If we remove any row r_j from the matrix $\mathfrak{M}^{(k)}$, then the remaining part is denoted by $\mathfrak{M}^{(k)} - \{r_j\}$.

The submatrix $\mathfrak{M}^{(k)} \subseteq \mathfrak{M}(\mathcal{K}_s)$ is called a M-minimal cover if $\mathfrak{M}^{(k)}$ satisfies the following conditions:

a) $C\left[\mathfrak{M}^{(k)}\right] = (1, 1, \ldots, 1)$

b) $\bar{\exists}\, \mathfrak{M}^{(k')} \subseteq \mathfrak{M}^{(k)} : C\left[\mathfrak{M}^{(k')}\right] = (1, 1, \ldots, 1)$.

If $\mathfrak{M}^{(k)}$ only satisfies condition (a) then it is called a M-cover.

Let be given a relation scheme $S = \langle \Omega, F \rangle$ and the set of all its keys $\mathcal{K}_s$. Then the matrix $\mathfrak{M}(\mathcal{K}_s)$ is completely determined, $\mathfrak{M}(\mathcal{K}_s)$ is a M-cover, $r_i \neq 0$ and $r_i \leq C\left[\mathfrak{M}(\mathcal{K}_s)\right] = (1, 1, \ldots, 1)$, $\forall i \leq m$, $\mathfrak{M}(\mathcal{K}_s)$ contains at least one M-minimal cover submatrix.

In the following, we will show that $S = \langle \Omega, F \rangle$ has at least one antikey, *assuming that* $(\emptyset \to \Omega) \notin F$.

1.3 Example:

Let $S = \langle \Omega, F \rangle$ be a relation scheme

and $\mathcal{K}_s = \{K_1, K_2, K_3, K_4\}$,

where $K_1 = \{a_1, a_2\}$, $K_2 = \{a_2, a_3, a_4\}$,

$K_3 = \{a_2, a_4, a_5\}$, $K_4 = \{a_4, a_6\}$.

a) By definition 1.1 :

$$M = \{1,2,3,4\}$$

$$H = \bigcup_{i=1}^{4} K_i = \{a_1, a_2, a_3, a_4, a_5, a_6\}$$

$I_1 = \{1\}$ $I_2 = \{1,2,3\}$ $I_3 = \{2\}$

$I_4 = \{2,3,4\}$ $I_5 = \{3\}$ $I_6 = \{4\}$

$\mathcal{J}_M = \{I_1, I_2, I_3, I_4, I_5, I_6\}$ is an M-cover:

$$M = \bigcup_{i=1}^{6} I_i = \{1,2,3,4\} \quad .$$

And $\mathcal{N}_1 = \{I_1, I_4\}$ $\mathcal{N}_2 = \{I_2, I_4\}$

$\mathcal{N}_3 = \{I_2, I_6\}$ $\mathcal{N}_4 = \{I_1, I_3, I_5, I_6\}$

are M-minimal covers .

b) By definition 1.2 :

$r_1 = (1,0,0,0)$ $r_2 = (1,1,1,0)$ $r_3 = (0,1,0,0)$

$r_4 = (0,1,1,1)$ $r_5 = (0,0,1,0)$ $r_6 = (0,0,0,1)$

$$\mathcal{M}(\mathcal{K}_s) = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$c[\mathfrak{M}(\mathcal{K}_s)] = (1,1,1,1)$$

and

$$\mathfrak{M}^{(1)} = \begin{pmatrix} r_1 \\ r_4 \end{pmatrix} = \begin{pmatrix} 1 & o & o & o \\ o & 1 & 1 & 1 \end{pmatrix} \qquad c[\mathfrak{M}^{(1)}] = (1,1,1,1)$$

$$\mathfrak{M}^{(2)} = \begin{pmatrix} r_2 \\ r_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & o \\ o & 1 & 1 & 1 \end{pmatrix} \qquad c[\mathfrak{M}^{(2)}] = (1,1,1,1)$$

$$\mathfrak{M}^{(3)} = \begin{pmatrix} r_2 \\ r_6 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & o \\ o & o & o & 1 \end{pmatrix} \qquad c[\mathfrak{M}^{(3)}] = (1,1,1,1)$$

$$\mathfrak{M}^{(4)} = \begin{pmatrix} r_1 \\ r_3 \\ r_5 \\ r_6 \end{pmatrix} = \begin{pmatrix} 1 & o & o & o \\ o & 1 & o & o \\ o & o & 1 & o \\ o & o & o & 1 \end{pmatrix} \qquad c[\mathfrak{M}^{(4)}] = (1,1,1,1)$$

are M-minimal covers in matrix representation.

§2. Let

$\mathfrak{F} = \{\mathfrak{M}^{(k)} \mid \mathfrak{M}^{(k)}$ is a M-minimal cover, $k \le \mu\}$ be the set of all M-minimal covers.

Theorem 2.1

Let $r_i \in \mathfrak{M}(\mathcal{K}_s)$ be any row. Then there exists a M-minimal cover $\mathfrak{M}^{(k)} \subseteq \mathfrak{M}(\mathcal{K}_s)$ such that $r_i \in \mathfrak{M}^{(k)}$.

Proof:

Let be given any row $r_i \in \mathfrak{M}(\mathcal{K}_s)$.

1) The case: $c[r_i] = (1,1,1,\dots,1)$ then $r_i \in \{r_i\}$.

2) The case: $c[r_i] \neq (1,1,1,\dots,1)$

i) If $\mathfrak{M}(\mathcal{K}_s) \in \mathfrak{F}$ then $r_i \in \mathfrak{M}(\mathcal{K}_s)$.

ii) If $\mathfrak{M}(\mathcal{K}_s) \notin \mathfrak{F}$. From $c[\mathfrak{M}(\mathcal{K}_s)] = (1,1,\dots,1)$ there exists $j \neq i$ such that $c[\mathfrak{M}(\mathcal{K}_s) - \{r_j\}] = (1,1,1,\dots,1)$. In fact, suppose the contrary, that

$\forall j \neq i : \ c[\mathcal{M}(\mathcal{K}_s) - \{r_j\}] \neq (1,1,\ldots,1)$.

On the other hand: $\mathcal{M}(\mathcal{K}_s) \notin \exists$, $c[\mathcal{M}(\mathcal{K}_s)] =$

$(1,1,\ldots,1)$, showing that $c \ [\mathcal{M}(\mathcal{K}_s) - \{r_i\}] = (1,1,\ldots 1)$.

Because $\forall j \neq i$, $c[\mathcal{M}(\mathcal{K}_s) - \{r_j\}] = (\eta_1, \eta_2, \ldots, \eta_m)$

$\neq (1,1,\ldots,1)$ there exists a column q_j such that $\eta_{q_j} = o$,

showing that $\alpha_{jq_j} = 1$ and $\alpha_{tq_j} = o$ for every $t \neq j$.

Let $j_1 \neq j_2$, $j_k \neq i$, $k = 1,2$ then $q_{j_1} \neq q_{j_2}$.

Were this false, and we have $q_{j_1} = q_{j_2}$.

Consequently $\alpha_{j_1 q_{j_1}} = \alpha_{j_2 q_{j_1}} = 1$ i.e in the

q_{j_1} - th column there are two elements equal to 1.

Hence for the submatrix

$\mathcal{M}(\mathcal{K}_s) - \{r_{j_1}\}$ we have $\eta_{q_{j_1}} = 1$, a contradiction.

Thus, for all $j \in \{1,2,\ldots,i-1,i+1,\ldots,p\}$ we have different columns $q_1, q_2, \ldots, q_{i-1}, q_{i+1}, \ldots, q_p$ such that in each column there is only one element equal to 1.

It follows that the vector r_i has the q_i-th component equal to 1. Since $c\ [\mathcal{M}(\mathcal{K}_s) - \{r_i\}] = (1,1,\ldots,1)$ then in the q_i-th column there are at least two elements equal to 1. Suppose $\alpha_{iq_i} = \alpha_{i'q_i} = 1$, $i \neq i'$.

It follows that $K_{q_i'} \subset K_{q_i}$. We arrive to a contradiction, (by the definition of a key.).

We have proved that there exists $j \neq i$ such that

$c[\mathcal{M}(\mathcal{K}_s) - \{r_j\}] = (1,1,\ldots,1)$.

Now, let us consider the submatrix $\mathcal{M}^{(1)} = \mathcal{M}(\mathcal{K}_s) - \{r_j\}$ $\subset \mathcal{M}(\mathcal{K}_s)$

a) If $\mathcal{M}^{(1)} \in \exists$ then $r_i \in \mathcal{M}^{(1)}$

b) If $\mathcal{M}^{(1)} \notin \mathfrak{I}$ and because $C[\mathcal{M}^{(1)}] = (1,1,\ldots,1)$ then there exists $j_1 \neq i$ such that $C[\mathcal{M}^{(1)} - \{r_{j_1}\}] = (1,1,\ldots,1)$. Since the matrix $\mathcal{M}(\mathcal{K}_s)$ has p rows and $p < +\infty$, it follows that there exists $k > o$ such that

$$\mathcal{M}(\mathcal{K}_s) \supset \mathcal{M}^{(1)} \supset \ldots \supset \mathcal{M}^{(k)} \supset \phi$$

and $\mathcal{M}^{(l)} \notin \mathfrak{I}$, $C[\mathcal{M}^{(l)}] = (1,1,\ldots,1)$ $o \leq l \leq k-1$,

$$\mathcal{M}^{(k)} \in \mathfrak{I} \Rightarrow r_i \in \mathcal{M}^{(k)}$$

The theorem 2.1 is completely proved.

From Theorem 2.1 , we have the following corollary.

Corollary 2.1:

Any M-cover has a M-minimal cover.

Definition:

Let $\mathcal{M}^{(k)} = \begin{pmatrix} r_{i_1} \\ r_{i_2} \\ \vdots \\ r_{i_k} \end{pmatrix}$ be a M-minimal cover

(or $\mathcal{N} = \{I_{i_1}, I_{i_2}, \ldots, I_{i_k}\}$).

Then the set $Q = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\} \subseteq H$ determined by the matrix $\mathcal{M}^{(k)}$ (or $\mathcal{N}$) is called a representative set of the set of all keys.

Theorem 2.2:

Let $S = \langle \Omega , F \rangle$ be a relation scheme and $\mathcal{K}_s$ be the set of all its keys. Let $Q \subseteq H$, and let $\mathcal{N} \subseteq \mathfrak{I}_M$ be the set determined by Q (or the matrix $\mathcal{M}^{(k)} \subseteq \mathcal{M}(\mathcal{K}_s)$). Then the set $\mathcal{N}$ is a M-minimal cover if and only if the set Q satisfies following conditions:

a) $\forall K \in \mathcal{K}_s \Rightarrow \exists a \in Q$ such that $a \in K$

b) $\forall Q' \subset Q \Rightarrow \exists K \in \mathcal{K}_s$ such that $\forall a \in Q'$ $\Rightarrow a \notin K$

Proof:

Suppose that $\mathcal{N} \leq \mathcal{J}_M$ is a M-minimal cover. We need prove that the set Q satisfies both conditions (a) and (b) .

a) Since $M = \bigcup_{I_i \in \mathcal{N}} I_i$ then for all $i \in M$ there exists $I_j \in \mathcal{N}$ such that $i \in I_j \iff$ for all $K \in \mathcal{K}_s$ there exists $a \in Q$ such that $a \in K$.

b) Let Q' be any proper subset of Q . The set Q' determines $\mathcal{N}' \subset \mathcal{N}$. Then there exists $j \in M$ and $j \notin \bigcup_{I_i \in \mathcal{N}'} I_i$. Equivalently, there exists $K \in \mathcal{K}_s$ such that for every $a \in Q'$: $a \notin K$.

Conversely , let Q be a set that satisfies both conditions (a) and (b) . We have to prove that the set $\mathcal{N} \leq \mathcal{J}_M$ determined by the set Q is a M-minimal cover.

i) It is clear that $M \supseteq \bigcup_{I_i \in \mathcal{N}} I_i$. We must prove that $M \subseteq \bigcup_{I_j \in \mathcal{N}} I_j$ i.e for all $i \in M$ then $i \in \bigcup_{I_j \in \mathcal{N}} I_j$.

Since for all $K \in \mathcal{K}_s$, there exists $a \in Q$ such that $a \in K \iff$ for all $i \in M$, there exists $I_j \in \mathcal{N}$ such that $i \in I_j \iff$ for all $i \in M$, there exists $I_j \in \mathcal{N}$ such that $i \in \bigcup_{I_j \in \mathcal{N}} I_j$.

ii) Let $\mathcal{N}'$ be any proper subset of $\mathcal{N}$. The set $\mathcal{N}'$ is determined by the set Q'. It is obvious that Q' is a proper subset of Q . Hence by condition (b) there exists $K \in \mathcal{K}_s$ such that for all $a \in Q'$, $a \notin K \iff$ there exists $i \in M$ such that for all $I_j \in \mathcal{N}'$, $i \notin I_j$.

This shows that $M \supset \bigcup_{I_j \in \mathcal{N}'} I_j$.

Theorem 2.2 is completely proved.

Theorem 2.3:

Let $S = \langle \Omega, F \rangle$ be a relation scheme and $\mathcal{K}_s$ be the set of all of its keys. Let Q be any subset of H. Then the set $K^{-1} = \Omega - Q$ is an antikey for S if and only if the set $\mathcal{N}$, determined by the set Q, is a M-minimal cover .

Proof:

The only if part: Let K^{-1} be any antikey for S . We show that the set $\mathcal{N}$ determined by the set $Q = \Omega - K^{-1}$ is a M-minimal cover, i.e we must prove that the set Q satisfies the following conditions:

a) For all $K \in \mathcal{K}_s$ there exists $a \in Q$ such that $a \in K$

b) For any $Q' \subset Q$, there exists $K \in \mathcal{K}_s$ such that for all $a \in Q'$ then $a \notin K$.

Now let us show the condition (a) :
Since K^{-1} is an antikey for S , for every $K \in \mathcal{K}_s$, $K \nsubseteq K^{-1} = \Omega - Q$. Then there exists $a \in Q$ such that $a \in K$.

We remain to prove the condition (b). Let Q' be any proper subset of Q . Since $Q' \subset Q$ then $\Omega - Q \subset \Omega - Q'$, i.e $X = \Omega - Q'$ is an extension for $K^{-1} = \Omega - Q$. By the definition for antikey, there exists $K \in \mathcal{K}_s$ such that $K \subseteq X = \Omega - Q'$, i.e for all $a \in Q'$, $a \notin K$.

The if part: Suppose $Q \subseteq H$ satisfies both condition (a) and (b). Let us show $K^{-1} = \Omega - Q$ is an antikey for S, i.e we must prove that

α) For all $K \in \mathcal{K}_s$, $K \nsubseteq K^{-1}$

β) For any $X \subseteq \Omega$ is an extension+/ of K^{-1} $(K^{-1} \subset X)$, there exists $K \in \mathcal{K}_s$ such that $K \subseteq X$.

Now we prove

α) Assume the contrary that there exists $K \in \mathcal{K}_s$ such that $K \subseteq K^{-1} = \Omega - Q$, i.e for all $a \in Q$ then $a \notin K$. This contradicts to the condition (a). Thus for all $K \in \mathcal{K}_s$ then $K \nsubseteq K^{-1}$ holds.

β) Let X be a subset of Ω such that $K^{-1} \subset X$, i.e $K^{-1} = \Omega - Q \subset X$. It follows that there exists $a \in X$ and $a \in Q$. Consider $Q' = Q - \{a\} \subset Q$. By the condition (b) there exists $K \in \mathcal{K}_s$ such that for all $a' \in Q'$ then $a' \notin K$. That means

$$K \subseteq \Omega - Q' = (\Omega - Q) \cup \{a\} \subset X \cup \{a\} = X \text{, i.e}$$

$K \subseteq X$. Thus, for all extension X of K^{-1}, there exists $K \in \mathcal{K}_s$ such that $K \subseteq X$.

The proof is complete.

From Theorem 2.3 the following corollaries are obvious.

Corollary 2.2:

Let $S = \langle \Omega, F \rangle$ be a relation scheme and $\mathcal{K}_s$ be the set of all of its keys. Then any antikey K for S has the following form:

$$K^{-1} = \Omega - \{a_{i_1}, a_{i_2}, a_{i_3}, \dots, a_{i_k}\} .$$

Where $Q = \{a_{i_1}, a_{i_2}, a_{i_3}, \dots, a_{i_k}\} \subseteq H$ is a representative set of the set $\mathcal{K}_s$.

Corollary 2.3:

Let $S = \langle \Omega, F \rangle$ be a relation scheme, $\mathcal{K}_s$ be

+/ i.e. a superset of K^{-1}

the set of all keys for S , $\mathcal{K}_S^{-1}$ be the set of all of its antikeys, and $\mathfrak{I}$ be the set of all M-minimal covers.

Then $$|\mathfrak{I}| = |\mathcal{K}_S^{-1}|.$$

Where $|\mathfrak{I}|$ is the cardinality of $\mathfrak{I}$ and $|\mathcal{K}_S^{-1}|$ is the cardinality of $\mathcal{K}_S^{-1}$.

Theorem 2.4:

Let $S = \langle \Omega, F \rangle$ be a relation scheme and $\mathcal{K}_S$ be the set of all keys for S . The set $\{Q_i\}$ of all representative sets for $\mathcal{K}_S$ will be denoted by $\mathcal{Q}$.

Then $$H = \bigcup_{K_i \in \mathcal{K}_S} K_i = \bigcup_{Q_i \in \mathcal{Q}} Q_i .$$

Proof:

It is obvious that $\bigcup_{Q_i \in \mathcal{Q}} Q_i \subseteq H$.

We have to prove that

$$H \subseteq \bigcup_{Q_i \in \mathcal{Q}} Q_i .$$

By Theorem 2.1 , for any row $r_i \in \mathcal{M}(\mathcal{K}_S)$, there exists a M-minimal cover $\mathcal{M}^{(k)} \subseteq \mathcal{M}(\mathcal{K}_S)$ such that $r_i \in \mathcal{M}^{(k)}$. This is equivalent to say that, for all $a_j \in H$ there exists a M-minimal cover $\mathcal{N} \subseteq \mathfrak{I}_M$ such that $I_j \in \mathcal{N}$. Let Q_t be the representative set which determines $\mathcal{N}$. Obviously $a_j \in Q_t$, i.e $a_j \in \bigcup_{Q_i \in \mathcal{Q}} Q_i$, showing that $H \subseteq \bigcup_{Q_i \in \mathcal{Q}} Q_i$. The proof is complete.

Definition:

Let $S = \langle \Omega, F \rangle$ be a relation scheme. Let us denote

$$G^* = \Omega - H$$

$$\Omega_1 = \Omega - G^*$$
$$F_1 = F - G^*$$
and $$S_1 = \langle \Omega_1, F_1 \rangle .$$

In [3] HO THUAN and LE VAN BAO proved that the set of all keys for $S = \langle \Omega, F \rangle$ is the same as the set of all keys for $S_1 = \langle \Omega_1, F_1 \rangle$, i.e $\mathcal{K} = \mathcal{K}_S = \mathcal{K}_{S_1}$.

Thus the set of all M-minimal covers on Ω is the same as the set of all M-minimal covers on Ω_1.

Let us investigate this problem in more detail.

We have the following theorem:

Theorem 2.5:

a) If K_S^{-1} is an antikey for $S = \langle \Omega, F \rangle$ then $K_{S_1}^{-1} = K_S^{-1} - G^*$ is an antikey for $S_1 = \langle \Omega_1, F_1 \rangle$.

b) If $K_{S_1}^{-1}$ is an antikey for $S_1 = \langle \Omega_1, F_1 \rangle$ then $K_S^{-1} = K_{S_1}^{-1} \cup G^*$ is an antikey for $S = \langle \Omega, F \rangle$.

Proof:

A). Let be given K_S^{-1} an antikey for $S = \langle \Omega, F \rangle$. We must show that $K_S^{-1} - G^*$ is an antikey for $S_1 = \langle \Omega_1, F_1 \rangle$.

i) Since $K_S^{-1} \in \mathcal{K}_S^{-1}$, for all $K \in \mathcal{K}$ then $K \nsubseteq K_S^{-1}$. It follows that $K \nsubseteq K_S^{-1} - G^*$.

ii) Let X be a subset of Ω_1 such that $K_S^{-1} - G^* \subset X$. Since $\Omega_1 = \Omega - G^* = H$ and

$K \subseteq \Omega_1$, $\forall K \in \mathcal{K}_S$, we have $K \cap G^* = \emptyset$.

It is easy to see that $K_S^{-1} \subset X \cup G^*$ and $\exists K \in \mathcal{K}_S$: $K \subseteq X \cup G^*$. Consequently, $K \subseteq X$.

Thus, for $X \subseteq \Omega_1$ which is an extension of $K_s^{-1} - G^*$, there exists $K \in \mathcal{K}$ such that $K \subseteq X$.

Combined (i) with (ii) we concluded that $K_{s_1}^{-1} = K_s^{-1} - G^*$ is an antikey for $S_1 = \langle \Omega_1, F_1 \rangle$.

b) Suppose that $K_{s_1}^{-1}$ is an antikey for $S_1 = \langle \Omega_1, F_1 \rangle$. We prove that $K_s^{-1} = K_{s_1}^{-1} \cup G^*$ is an antikey for $S = \langle \Omega, F \rangle$.

i) Since $K_{s_1}^{-1} \in \mathcal{K}_{s_1}^{-1}$ ($\mathcal{K}_{s_1}^{-1}$ is the set of all antikeys for S_1), we have $K \nsubseteq K_{s_1}^{-1}$, $\forall K \in \mathcal{K}$.

Since $K \subseteq \bigcup_{K_i \in \mathcal{K}} K_i = H$, we have $K \cap G^* = \phi$.

It follows that $K \nsubseteq K_{s_1}^{-1} \cup G^*$, $\forall K \in \mathcal{K}$.

ii) Let be given any $X \subseteq \Omega$ such that $K_{s_1}^{-1} \cup G^* \subset X$. It follows that $K_{s_1}^{-1} \subset X$. Since $K_{s_1}^{-1} \in \mathcal{K}_{s_1}^{-1}$, there exists $K \in \mathcal{K}$ such that $K \subseteq X$.

Thus for any $X \subseteq \Omega$ which is an extension of K_s^{-1}, there exists $K \in \mathcal{K}$ such that $K \subseteq X$.

The Theorem is completely proved.

In this paper we do not present the algorithm to determine the set of all M-minimal covers for any relation scheme $S = \langle \Omega, F \rangle$ also as the algorithm to recognize whether a given set $X \subseteq \Omega$ is or is not a representative set of $\mathcal{K}$.

In other words, we have not proposed an algorithm to find all the antikeys of a relation scheme.

They will be presented in a subsequent paper.

We close our paper with an example.

2.1 Example:

Let $S = \langle \Omega, F \rangle$ be a relation scheme and $\mathcal{K}_s$ be the set of all keys for S.

Where $\Omega = \{a_o, a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$

$$\mathcal{K}_s = \{K_1, K_2, K_3, K_4\}$$

$$K_1 = \{a_1, a_2\} \qquad K_2 = \{a_2, a_3, a_4\}$$

$$K_3 = \{a_2, a_4, a_5\} \qquad K_4 = \{a_4, a_6\} \quad .$$

then $\Omega_1 = H = \bigcup_{i=1}^{4} K_i = \{a_1, a_2, a_3, a_4, a_5, a_6\}$

$$G^* = \Omega - H = \{a_o, a_7\} .$$

The Q_i sets of $\mathcal{K}$	The antikeys for S_1	The antikeys for S
$Q_1 = \{a_1, a_4\}$	$K^{-1} = \{a_2, a_3, a_5, a_6\}$	$K^{-1} = \{a_o, a_2, a_3, a_5, a_6, a.$
$Q_2 = \{a_2, a_6\}$	$K^{-1} = \{a_1, a_3, a_4, a_5\}$	$K^{-1} = \{a_o, a_1, a_3, a_4, a_5, a.$
$Q_3 = \{a_2, a_4\}$	$K^{-1} = \{a_1, a_3, a_5, a_6\}$	$K^{-1} = \{a_o, a_1, a_3, a_5, a_6, a.$
$Q_4 = \{a_1, a_3, a_5, a_6\}$	$K^{-1} = \{a_2, a_4\}$	$K^{-1} = \{a_o, a_2, a_4, a_7\}$

Acknowledgement:

The author would like to thank Prof. Dr. J, Demetrovics for his help and encouragement.

The author is also very grateful to Dr. Ho Thuan for his valuable remarks and suggestions during the the preparation of this paper.

References:

[1] Demetrovics, J., On the equivalence of candidate keys with Sperner sys., Acta Cybernetica 4/1979/,247-252.

[2] Demetrovics, J. and Ho Thuan, Keys and superkeys for relation scheme. J. of Computers and Artificial Intelligence,Bratislava,N^{o} 6, 1986.

[3] Ho Thuan and Le van Bao, Translation of relation scheme, Serdica 1986.

[4] Ho Thuan and Le van Bao, Some results about keys of relational schemas. Acta Cybernetica, Tom. 7. Facc. I,Szeged, 1985, pp. 99-113.

[5] Demetrovics, J. and Ho Thuan, Some additional properties of keys for relation schemes, Serdica, Sofia /To appear/.

Связь между М-минимальными покрытиями и анти-ключами в реляционных схемах

Пхам Тхе Куе

Резюме

В статье определяется понятие М-минимального покрытия реляционной схемы и доказаны его основные свойства. На основе этих свойств доказаны необходимые и достаточные условия того, чтобы $X \subset \Omega$ было множество анти-ключей /множество ключей предполагается знакомым/.

KAPCSOLAT AZ M-MINIMÁLIS LEFEDÉSEK ÉS AZ ANTI-KULCSOK KÖZÖTT A RELÁCIÓS SÉMÁBAN

Pham The Que

Összefoglaló

A szerző bevezeti a relációs séma M-minimális lefedésének fogalmát és megvizsgálja néhány tulajdonságát. Vizsgálatainak eredményeképpen szükséges és elégséges feltételt ad arra, hogy az $X \subset \Omega$ az anti-kulcsok halmaza legyen (amennyiben a kulcsok halmazát ismerjük).

MTA SZTAKI Közlemények 36/1987 pp. 123-144

MATHEMATICAL MODELS OF DATA SECURITY PROCESSES IN CENTRALIZED AND DISTRIBUTED DATA BASE SYSTEMS

B. SZAFRANSKI

Zawadzkiego str. 18/38
Warsaw, Poland

Abstract

Program data security in currently exploited data base systems in general is based upon access control mechanism. The detailed analysis of data processing from data base leads to conclusion, that mechanism of this type do not guarantee secure data processing because of keeping secrecy. Therefore in the systems of especially rigorous requirements on efficiency data security it is nenecessary to inculcate besides access control mechanism also data flow control mechanism. The paper includes main elements of formal models of such mechanism. Subsequently basing upon properties of the models and conclusions from analysis of semantics of data processing from data base it has been demonstrated, that some models elements generate universally bounded lattices, that have been defined as data flow lattice and operation scope lattice. Continuing considerations we have defined the algebraic structure as a composition of above mentioned lattices. This structure fulfils all properties of the lattice and it is called data security lattice. Next we assumed that in each node of Distibuted Data Base System may exist local data base system in which data security policy is based on the local data security lattice. The formal model of determining the data security for the whole system is presented and a method of deriving a common lattice, which is called superlattice from local lattices is demonstrated. The consideration are illustrated with a number of examples.

1. Introduction

My paper deals with data security in database. It is sure, that user who gives away his data under control of data base system requires the guarantee from the system, that other users can use his data according to his demands. Program data security mechanism in currently exploited data base system are based upon either data access control conception or data flow control conception.

The data access control

The mechanism of data access control /Fig.1.1/ base on conception of access privileges matrix. This matrix determines the access privileges of active objects (users, programs) to passive objects (data units). On the Fig.1.1 there is showed an example of access privileges matrix. You can see that in such systems we differ a number of active objects (in this case JONES,...,SMITH) and a number of passive objects (data units A, B, C). Moreover, there exist a number of operations / in our example GET,PUT/. These operations can be used by active object to process passive objects. The policy of data access control rely on checking if the operations issued in this process are admissible for this active object /in contex of his access privilege/. Looking on the picture you can notice, for example, that GET operation issued by JONES to process A will be legal, but PUT will not.

1. The base elements:
   - active objects /like users, programs/,
   - passive objects /like data units/,
   - data processing operations /like GET, PUT/,
   - access privileges matrix, which determines access privileges of active objects to passive objects
2. Interpretation of access privileges matrix

Example:

	data unit A	data unit B	data unit C	...
JONES	GET	GET, PUT	X	
. . .				
SMITH	GET	GET	GET	

JONES can read data from data units A and B, write to data unit B, but he can't process data from data unit C. SMITH can read data from data units A, B, C.

3. The rule of acting:

   Data access control mechanism compares the operation specified in the program with the privilege in the matrix ,to reject unlegal operations.

Fig.1.1. The idea of data access control.

The data flow control

The mechanism of data flow control /Fig.1.2/ checks the correctness of data flows between objects. Data flow between objects A and B appeares then, when data in any way are transfered from object A to B. Copying data from file A to B is a simple example of data flow. Data flow control conception requires creating of set of secrecy classes. For example such set can contain public, confidential, secret classes. The objects are given the secrecy class from this set. The policy of flow control rely upon flow relation, which determines ordering of secrecy classes. For example, according to flow relation data can flow from confidential file to secret file, but can't flow in opposite direction.

1. The base elements:
   - objects /like users, data units/,
   - secrecy classes /like secret, public, top secret/,
   - flow relation,
   - flow operations /operations, which transfer data between objects; operations like PUT, GET, COPY, UPDATE/,
   - secrecy attributes matrix.
2. Interpretation of flow relation and secrecy attr.matrix:

Example:

Objects	Secrecy class
JONES	Secret
SMITH	Top secret
Data unit A	Public
Data unit B	Secret
Data unit C	Top secret

Flow relation determines ordering of secrecy classes set, Example:

(public, confidential, secret, top secret)

If ordering of above set is from left to right we can say:

"Data can flow from confidential data unit to secret, but those can not flow in opposite direction"

3. The rule of acting:

The data flow control mechanism examines each data processing operation, to reject these operations, which cause unlegal flow.

Fig.1.2. The idea of data flow control

2. The need of data security processes integration.

To clarify this problem we will consider very simple example, showed on Fig.2.1. You can see, that there exist two users U1 and U2 and two data units (files A and B). The user U1 can read the data from the file A and write to file B, but the user U2 can read from the file B only. The data access control mechanism can examine legality of operations using by the users. But it doesn't prevent the unlegal cooperating of users. For example, the user U1 according to his privileges reads up data from file A and he writes up these data to the file B. And now, user U2 can read these data from the file B because he has privilege of reading this file. It is easy to notice, the unlegal flow was realized, despite existing of data access control mechanism. Next we assume, the file A is secret and the file B is public. Then applying of data flow control mechanism could eliminate this unlegal flow, because it wouldn't permit data flow from the secret to public file.
From the other side , the data are transfered between objects as an result performing of different operations. Some of them are presented on Fig.2.2. It is obvious that they cause different effects in data base. For example, effects of doing UPDATE and ERASE operations can be potential uncomparable. The UPDATE operation is usually used to update one or more fields in record occurences, while the single performing of ERASE operation could cancel a big part of network data base, but unfortunatly they are treated by data flow control mechanism in the same way, as the flow operations. Therefore the users can require to differ these operations. From this point of view the data flow control mechanism don't satisfy such demand of users.

At the end we came to conclusion:

It is necessary to build integrated data security mechanism, which includes possibilities of data access control and flow control.

3. The simplified models of data security processes

The whole models of data access control and data flow control are in [1]. Below I citized only such elements of them, which are important from the aim of this paper.

3.1. Data access control

Def.1
Data access control model is e three:

$$AM = \langle Z, T, \Psi \rangle$$

where:
Z – objects names set,
T – operation names set,
Ψ – operation scope relation, $\Psi \subset T \times T$.

To clarify this relation, let us introduce formally the definition of operation scope:

Def.2

An operation scope $t_i \in T$ is an operation set $\{ t_j \} \in 2^T$ such, that the ability of doing the operation t_i implicates the ability of doing each operation $t_i \in \{ t_j \}$.

Def.3

The operation scope $\{ t_j \} \in 2^T$ of operation t_1 is smaller/equal/ than the operation scope $\{ t_k \}$ if only if $\{ t_j \} \subset \{ t_k \}$.

The operation scope relation is defined on the pairs of operation names.

Def.4

For $t_1, t_2 \in T$ we can say that $(t_1, t_2) \in \Psi$ if and only if the operation scope t_1 is smaller/equal/ than operation scope t_2.

Def.5

Data security mechanism basing upon data access control works correctly if it ensures that no data access is realized, which is not in accordance with determined operation scope relation.

3.2. Data flow control

Def.6

A data flow control model is a three:

$$FM = \langle Z, K, \varphi \rangle$$

where:

Z - objects names set,

K - secrecy classes set,

φ - flow relation,

$\varphi \subset K \times K$.

Def.7

For $k_1, k_2 \in K$ we say that $(k_1, k_2) \in \varphi$ if and only if data from secrecy class object k_1 can flow to object of secrecy class k_2.

Def.8

Data security mechanism basing upon data flow works correctly if it ensures, that no data flow is realized, which is not in accordance with defined flow relation.

3.3. Data security control

Def.9

A data security model is a three:

$$SM=\langle Z, A, \lambda \rangle$$

where:

Z - object names set,
A - A=KxT
λ - security relation,
AxA.

$$((a_i, a_j), (a_l, a_k)) \in \lambda$$

$$\text{when } (a_i, a_l) \in \varphi \wedge (a_j, a_k) \in \Psi$$

Def.10

Data security mechanism based on model SM works correctly if it ensures after performing any operation set, that no access and flow is realized, which is not in accordance with defined security relation.

According to the above definition data security model have the possibility of data access control anf flow control. Therefore it can be a base of integrated data security mechanism.

4. The lattice models of data security processes

4.1. The lattice structure

Def.11

The algebraic structure < X, ,+, > is a lattice if:
1/ X is a finite set,
2/ $\langle X, \leq \rangle$ is a partially ordered set,
3/ $\oplus$ is a binary operator on X such that $x \oplus y$ is the least upper bound for any $x, y \in X$,
4/ $\odot$ is a binary operator on X such that $x \odot y$ is the greatest lower bound for any $x, y \in X$.

Operators $\oplus$ and $\odot$ we called the least upper and greatest lower bounds operators, respectively.

Def.12

A lattice is an universally bounded lattice if there exist elements 1 and 0 such that $x \leq 1$ and $0 \leq x$ for all $x \in X$.

We call the elements 1 and 0 the universal upper and universal lower bounds, respectively.

4.2. Data flow lattice

D. Denning in her work entittled A Lattice Model of Secure Information Flow proved on the base of data processing semantics analysis, that secrecy classes set K and flow relation create universally bounded lattice. The lattice will be called a flow lattice and noted as:

$< K, \mathcal{f}, \square, \Delta >$ with universal bounds k_{max} and k_{min} in which $\square$ and Δ are the least upper and greatest lower bounds operators, respectively.

4.3. Operation scope lattice

After analysis of relations between operations from the set T you can prove, that model elements AM, that is: operation names set T and scope relation create a universally bounded lattice $<T, \Psi, +, \bullet>$. For this reason it must be proved, that the following properties of the lattice are satisfied:

1/ T is a finite set,
2/ $< T, >$ is a partially ordered set,
3/ + and • are the least upper and greatest lower bounds operators, respectively,
4/ there exist universal lower and upper bounds t_{max}, t_{min}, repectively.

Now we will point out the satisfying some of these conditions. The satisfying of condition (1) issues from practical features of real systems. Subsequently, the relation must be reflexive, transitive and antisymmetric; it results from condition (2). It is obvious, that is reflexive, because operation scopes of the same opeartions are equal. The fact, that is antisymmetric results from the real assumption, that operation scopes are not redundant. You can prove for most of data processing operations from data base, that Ψ istransitive. Let us consider as an instance the set T={FIND, READ, UPDATE}. It is obvious, that before data reading you should first find them and before updating you should first read them. Therefore practical sense of the data processing requires besides the permission of executing the operation READ also the permission of executing the operation FIND. The permission of executing the operation UPDATE should contain the permission of executing the operation READ and through transitiveness the permission of executing the operation FIND.
The above considerations let us come to conclusion, that $<T, \Psi>$ is a partially ordered set.

Now we will consider a fragment of typical program demonstrated by means of data flow diagram of Fig.4.1. To check the correctness of such a program it would be required to examine the legality of each issued operation. It is worth to emphasize, that such and similar program sequences may occure many times in the program. Therefore it is important to search the ways of examining legality of the access which would decrease the number of necessary checks. For this reason it would be enough to check only the legality of operation having the greatest scope of activity /for example operation UPDATE/ to decide, whether the fragment of the prugram is correct. Moreover the way of checking correctness requires existence of an operator defined on T, which guarantees, that the way of defining the "resultant" operation does not reject the realizing of the legal access. It is the least upper bound operator, which satisfies the over mentioned conditions.

Continuing considerations you can prove the practical sense of remaining elements of the earlier defined lattice.

For instance, the sence of existence the lower bound operator results from the necessity of defining the data processing competence on the grounds of user's competence and the competence of program and terminal.

Summarising you can state, that if you introduce the lattice of operation scope into basic model AM, it is not only formal manipulation, but it results from the semantics of data base processing.

4.4. Data security lattice

Up to now we have discussed two independent lattices for data access control model and data flow control model. Basing upon above considerations we will define a composition of these lattices in this way:

$$\langle A, \lambda, \boxplus, \boxtimes \rangle = \langle K, \wp, \Box, \Delta \rangle \blacksquare \langle T, \Psi, +, \cdot \rangle$$

where:

$A = K \times T$ - is a set of all pairs, where the first element is from secrecy class set and the second from the operation names set.

$\lambda \subset A \times A$ - is a relation defined as follows:
$((a_i, a_j), (a_l, a_k)) \in \lambda$ when
$(a_i, a_l) \in \wp \wedge (a_j, a_k) \in \Psi$

$\boxplus$ - is a binary operator defined as follows:
for any $(a_i, a_j), (a_l, a_k) \in A$
$(a_i, a_j) \boxplus (a_l, a_k) = (a_i \Box a_l, a_j + a_k)$

$\boxtimes$ - is a binary operator defined as follows:
for any $(a_i, a_j), (a_l, a_k) \in A$
$(a_i, a_j) \boxtimes (a_l, a_k) = (a_i \Delta a_l, a_j \cdot a_k)$

a_{max}, a_{min} - are the elements of such, that:

$$a_{max} = (k_{max}, t_{max})$$
$$a_{min} = (k_{min}, t_{min})$$

On the way of formal provement you can show, that such defined composition satisfies all properties of a lattice. Tetrad $\langle A, \lambda, \boxplus, \boxtimes \rangle$ is said to be a data security lattices.

4.5. Examples

Figure 4.3 demonstrates a simple linear ordering of secrecy classes set, which satisfies the properties of data flow lattice. Graphic demonstration of the ordering is the preceding graph. Fig.4.4 shows linear ordering of operation set, which satisfies the properties of operation scope lattice. On the other hand, the Fig.4.5 shows graphic interpretation of data security lattice derived from lattices of Fig.4.3 and Fig.4.4. It is required from data security model not to allow data of higher secrecy class flow to lower secrecy class with simulataneous legality examining of performed operations. From the security class demonstrated as an instance on Fig.4.5 results, that object permitted to update public file can read and write to the public file, but it can not update the private file, becouse in that case flow relation is not satisfied.

5. Data security in Distributed Data Base System

I think that you can agree with opinion that modern data base systems are frequently distributed. Usually it means that the data are kept at widely dispersed locations. In other words, data base may be distributed in phisicaly separated nodes. To manage such systems we create the distributed data base systems, shortly called DDB Systems. Each node of DDB system may contains of local data base management system DBMS. Each DBMS has some capabilities of data security and therefore it is reasonable idea to design common data security policy for any DDB system. Quality of target policy strictly depends on correctness and completness of the designing process. Therefore this process should be supported by mathematical models and methods. DDB system with data security possibilities may be represented by generic architecture of Fig.5.1. From this figure you see that local possibilities of data security are reflected by local data security lattices and the global policy of data security in whole DDB system by global data security lattice so called superlattice. It is very important for such a system that all local DBMS implement the same kind of data security, for example only data access control or only data flow control conception. Formally it means for example, that sets X can differ, but X must have the same nature of elements (for example security levels or categories or data processing operations). The draft of methodology, which supports the process of determing the common data security policy in DDB system shown on Fig.5.2. This methodology consists of some theoretical foundations and auxilliary algorithms. Theoretical foundations include the formal conditions for consistency diagnostics of local lattices set, the way of superlattice construction and the set of operations to transform the superlattice according to changes in configuration of DDB system (including or excluding the nodes). The algorithms are necessary to make the process of superlattice construction easier and faster. In this paper I'm not going to present the whole methodology, but I'll explain the practical interpratation two following topics:

- consistency conditions of local lattices set,
- construction of superlattice.

The formal specifications of those problems are described in details in [2].

5.1. Consistency and strict consistency of local lattices set

When we want to integrate local DBMS in one DDB system we can expect two situations:

- the data security policies in nodes are consistent
- and the other, when are not consistent.

For example, in two nodes the data processing operations could be ordered, from data security point of view, in different way and therefore we must check this situation, to answer the question:

- can we or not integrate such local DBMSs in DDB system?

For this aim, I have introduced formal conditions of consistency or strict consistency of local lattices set.

Satisfying of consistency condition means that we can not create two (or more) such chains of elements from union of local sets, which order any two elements, in opposite direction (in sense of lattice relations). Please now look at the Fig.5.3. There is a set of lattices in graph form, which is inconsistent, because there exist two chains (C,F,B) and (B,E,C), which order elements C and B in opposite directions. Therefore we can state that integrating the nodes which refere to local lattices set showed on this figure is not possible from data security point of view.

In the case of strict consistency condition we have other situation. This condition prevents introducing additional ordering between elements of local sets, which can arise as a result of combination of local relations in global lattice. An example of such situation is shown on Fig.5.4.

You can see that this set of local lattices set is not strictly consistent because lattices L2 and L3 additionally ordered elements B and C in lattice L1.

However, such set of lattice can be base of superlattice constructing, but the designers of data security mechanism must remember that global data security policy can change the local policies in some nodes.

5.3. Construction of superlattice

When checking of consistency conditions is finished we can begin to construct the superlattice, obviously only for consistent or strictly consistent local lattices set.

This process consists of two stages.

The first stage includes the construction of global partially ordered set

antisymetric. Suitable proof id presented in the paper [2].
In the second stage we must check whether received partially ordered set $\langle X, \delta \rangle$ generates the lattice. It results from this, that in general even strict consistent local lattices set must not generate lattice. Let we consider example of Fig.5.6.
You can see, that strict cinsistency local lattices set of figures a and b generates the partially ordered set, but not the lattice, because elements E,M and A,G have not the least upper and greatest lower bounds. For example you see that elements E,M have three upper bounds (F,K,1), but none of them have properties of the least upper bound.

6. Conclusions

Properties of data security lattice model can be utilized to construction of more effective access control mechanisms, data flow mechanisms or data security mechanisms. The above effectiveness results from the following conditions:

- the storage size necessary for writing privileges decreases evident through defining the ordering relations of secrecy classes and operation sets;
- introducing of lower and upper operators decreases the number of accesses to operation scope, data flow and data security relation, respectively. It activates considerably the process of legality checking;
- the proof of the property should be one of the elements of any mechanism design. Such requirement is especially referred to data security mechanisms in data base systems. In case, where a lattice for a real data base system can be derived, compactness and completness are defined univocally. In this case the lattice is supposed to be a base of data security mechanism;
- the lattice has the form of dynamic structure, which can be easily copied in computer algorithms.

$\langle X, \delta \rangle$ from local partially sets $\langle X_i, \delta_i \rangle$ and the second includes checking if received pair $\langle X, \delta \rangle$ generates the lattice.
In the first stage we create the global set X as an union of local sets and additional sets D and G(see Fig. 5.5).
The sets D and G will be empty, when in union of local sets exist the elements, which will be the greatest lower and least upper bounds in future, constructing superlattice. If such elements do not exist, then we must add elements named here x_d and x_g.
Next, we create the global relation δ . To do it, we include to δ all pairs, which belong to local relations and its combinations. To prove, that relation received in this way partially ordered set X we should prove that it is reflexive, transitive and

Bibliography

1. SZAFRAŃSKI B. A Data Security Model in Data Base, ICS PAS Reports, Warsaw, 1979, No 352.

2. SZAFRAŃSKI B. The Model of Extended Data Macroflows Control in Distributed Data Base Activity, Proc. of the Conf. on Systems Sciences, Wrocław, 1986.

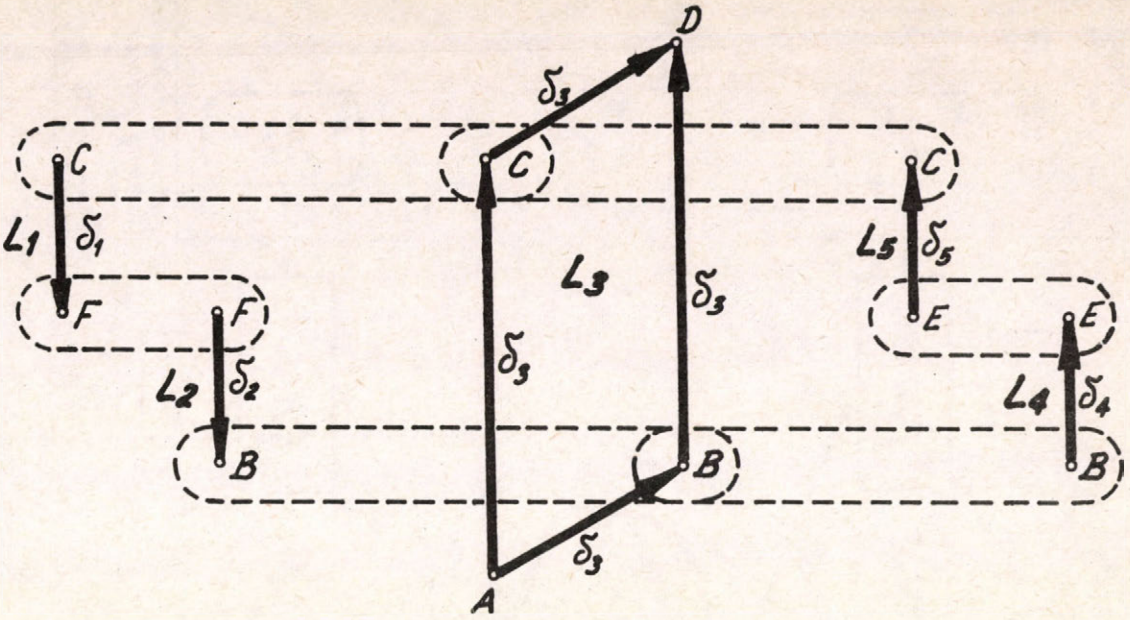

D
δ_3
C
C
C
L_1
δ_1
L_3
L_5
δ_5
F
F
δ_3
E
E
δ_3
L_2
δ_2
L_4
δ_4
B
B
B
δ_3
A

Fig. 2.1

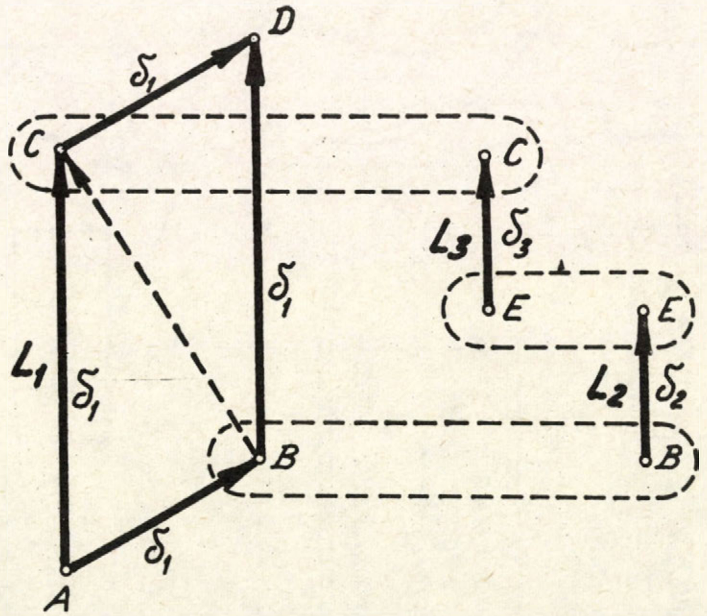

D
δ_1
C
C
L_3
δ_3
δ_1
E
E
L_1
δ_1
L_2
δ_2
B
B
δ_1
A

Fig. 2.2

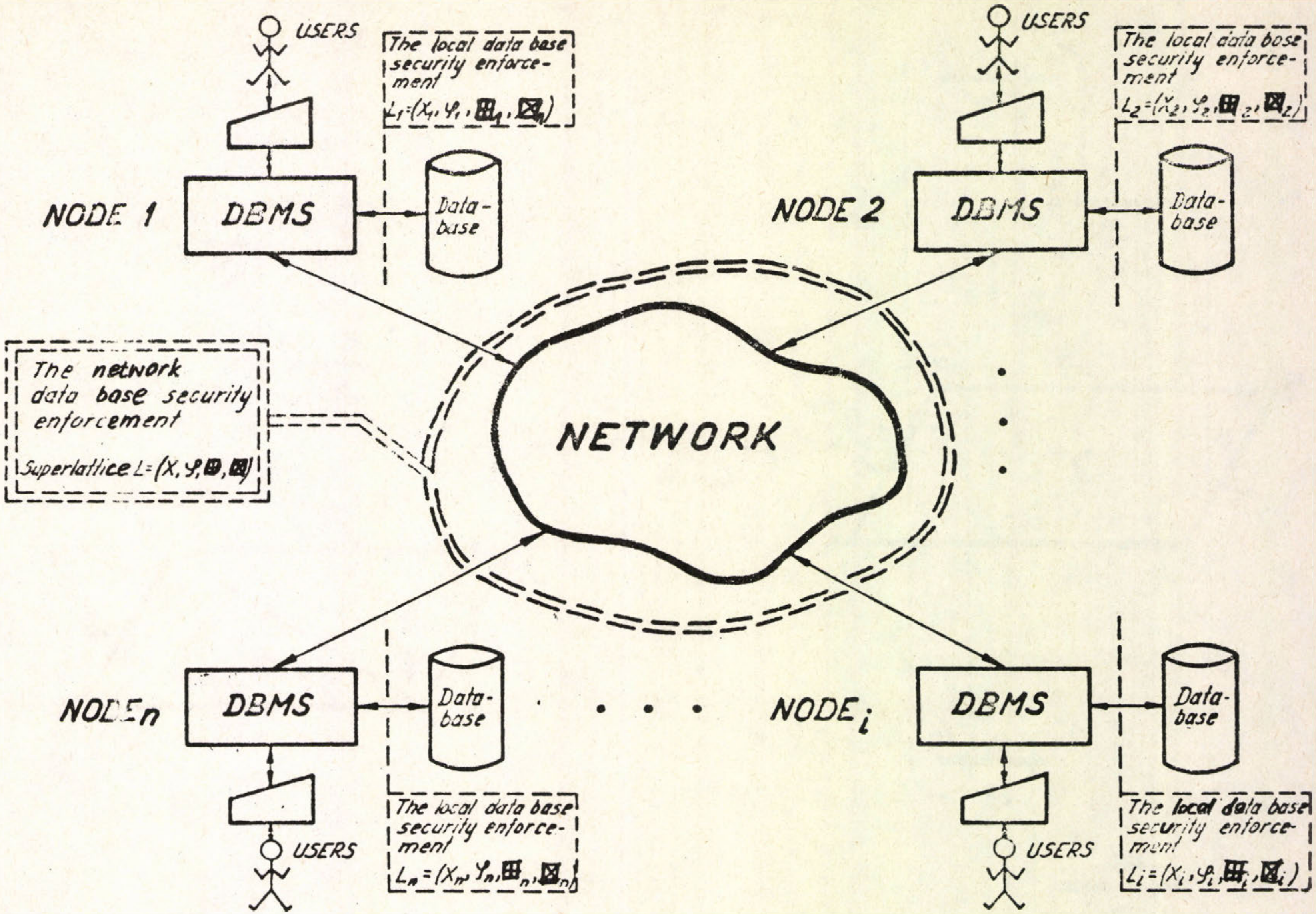

USERS
The local data base security enforcement
$L_1 = (X_1, \varphi_1, ⊞_1, ⊠_1)$
NODE 1
DBMS
Data-base
USERS
The local data base security enforcement
$L_2 = (X_2, \varphi_2, ⊞_2, ⊠_2)$
NODE 2
DBMS
Data-base
The network data base security enforcement
Superlattice $L = (X, \varphi, ⊞, ⊠)$
NETWORK
NODEn
DBMS
Data-base
The local data base security enforcement
$L_n = (X_n, \varphi_n, ⊞_n, ⊠_n)$
USERS
NODE$_i$
DBMS
Data-base
The local data base security enforcement
$L_i = (X_i, \varphi_i, ⊞_i, ⊠_i)$
USERS

Fig. 4.1

$K = \{1, 2, 3\}$

$1 \equiv$ IN PUBLIC

$2 \equiv$ IN PRIVATE

$3 \equiv$ IN SECRET

$(k_i, k_j) \in \varphi$ IF $(k_i \leq k_j)$.

$k_i \sqcap k_j = max(k_i, k_j)$.

$k_i \Delta k_j = min(k_i, k_j)$.

$k_{min} = 1,\ k_{max} = 3$

FIG. 4.2. DATA FLOW LATTICE.

$T = \{1, 2, 3\}$

$1 \equiv$ FIND

$2 \equiv$ READ

$3 \equiv$ UPDATE

$(t_i, t_j) \in \Psi$ IF $(t_i \leq t_j)$.

$t_i + t_j = max(t_i, t_j)$.

$t_i \cdot t_j = min(t_i, t_j)$.

$t_{min} = 1,\ t_{max} = 3$

FIG. 4.3. OPERATION SCOPE LATTICE

$A = \{(k_i, t_j) : k_i \in K, t_j \in T\}$.

$((k_i, t_j), (k_l, t_k)) \in \lambda$ IF

$(k_i, k_l) \in \varphi \wedge (t_j, t_k) \in \Psi$

$(k_i, t_j) \boxplus (k_l, t_k) =$

$= (max(k_i, k_l), max(t_j, t_k))$

$(k_i, t_j) \boxtimes (k_l, t_k) =$

$= (min(k_i, k_l), min(t_j, t_k))$.

$a_{min} = (1, 1),\ a_{max} = (3, 3)$.

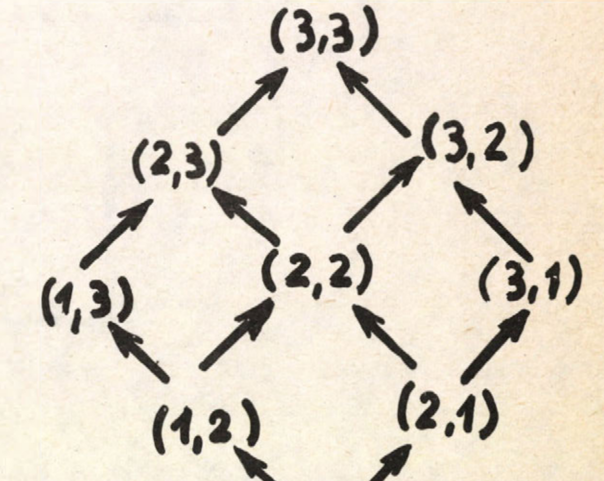


FIG. 4.4. DATA SECURITY LATTICE

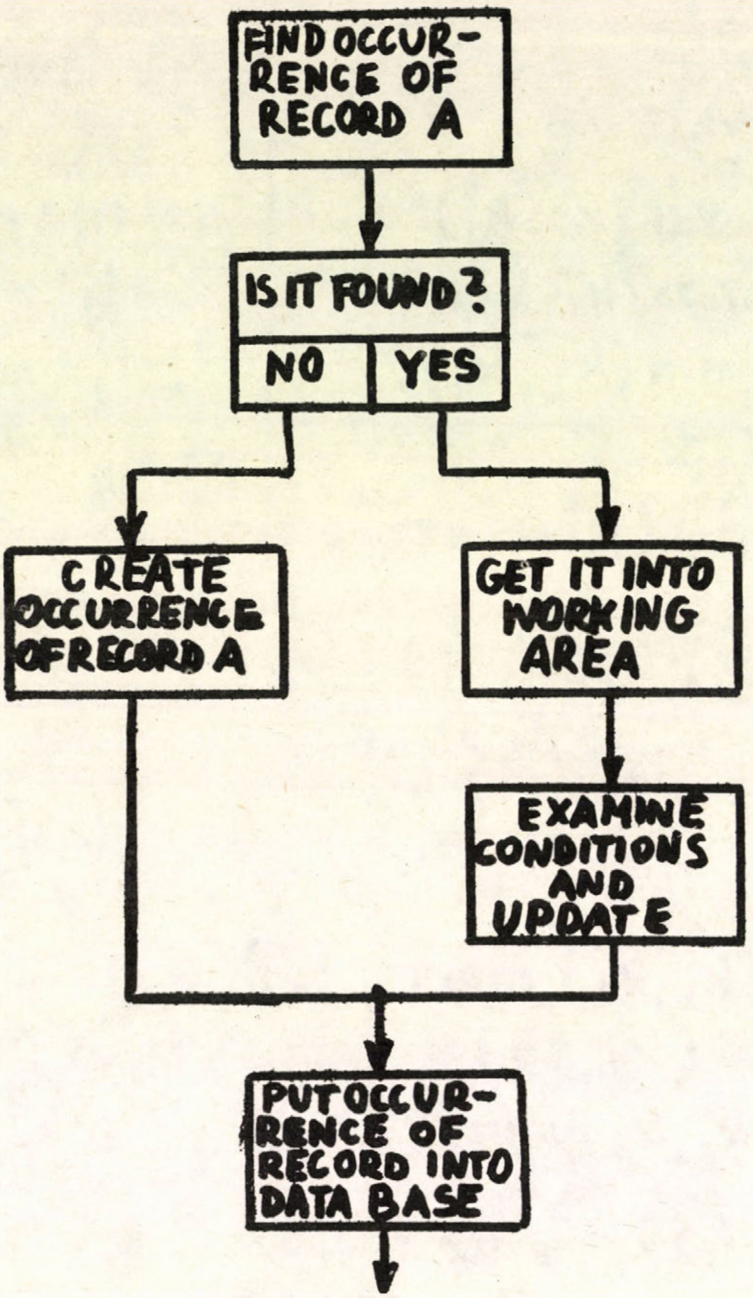

FIND OCCUR-
RENCE OF
RECORD A
IS IT FOUND?
NO
YES
CREATE
OCCURRENCE
OF RECORD A
GET IT INTO
WORKING
AREA
EXAMINE
CONDITIONS
AND
UPDATE
PUT OCCUR-
RENCE OF
RECORD INTO
DATA BASE

Fig. 5.1

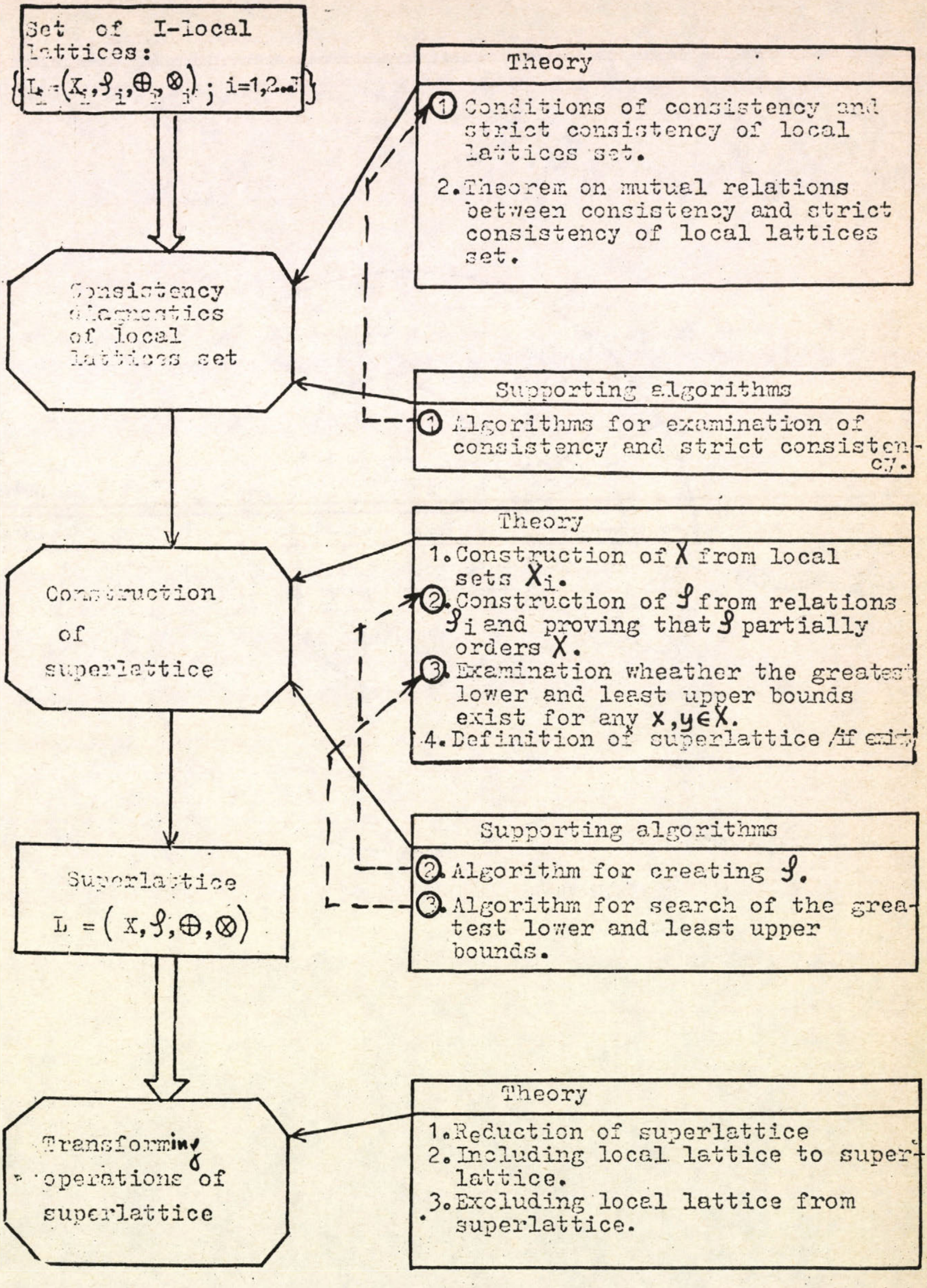

Set of I-local lattices: {L_i=(X_i, ℘_i, ⊕_i, ⊗_i); i=1,2..I}
Theory
1 Conditions of consistency and strict consistency of local lattices set.
2. Theorem on mutual relations between consistency and strict consistency of local lattices set.
Consistency diagnostics of local lattices set
Supporting algorithms
1 Algorithms for examination of consistency and strict consistency.
Theory
1. Construction of X from local sets X_i.
2. Construction of ℘ from relations ℘_i and proving that ℘ partially orders X.
3. Examination wheather the greatest lower and least upper bounds exist for any x,y∈X.
4. Definition of superlattice /if exist/
Construction of superlattice
Supporting algorithms
2. Algorithm for creating ℘.
3. Algorithm for search of the greatest lower and least upper bounds.
Superlattice L = (X, ℘, ⊕, ⊗)
Theory
1. Reduction of superlattice
2. Including local lattice to superlattice.
3. Excluding local lattice from superlattice.
Transforming operations of superlattice

Fig. 5.2

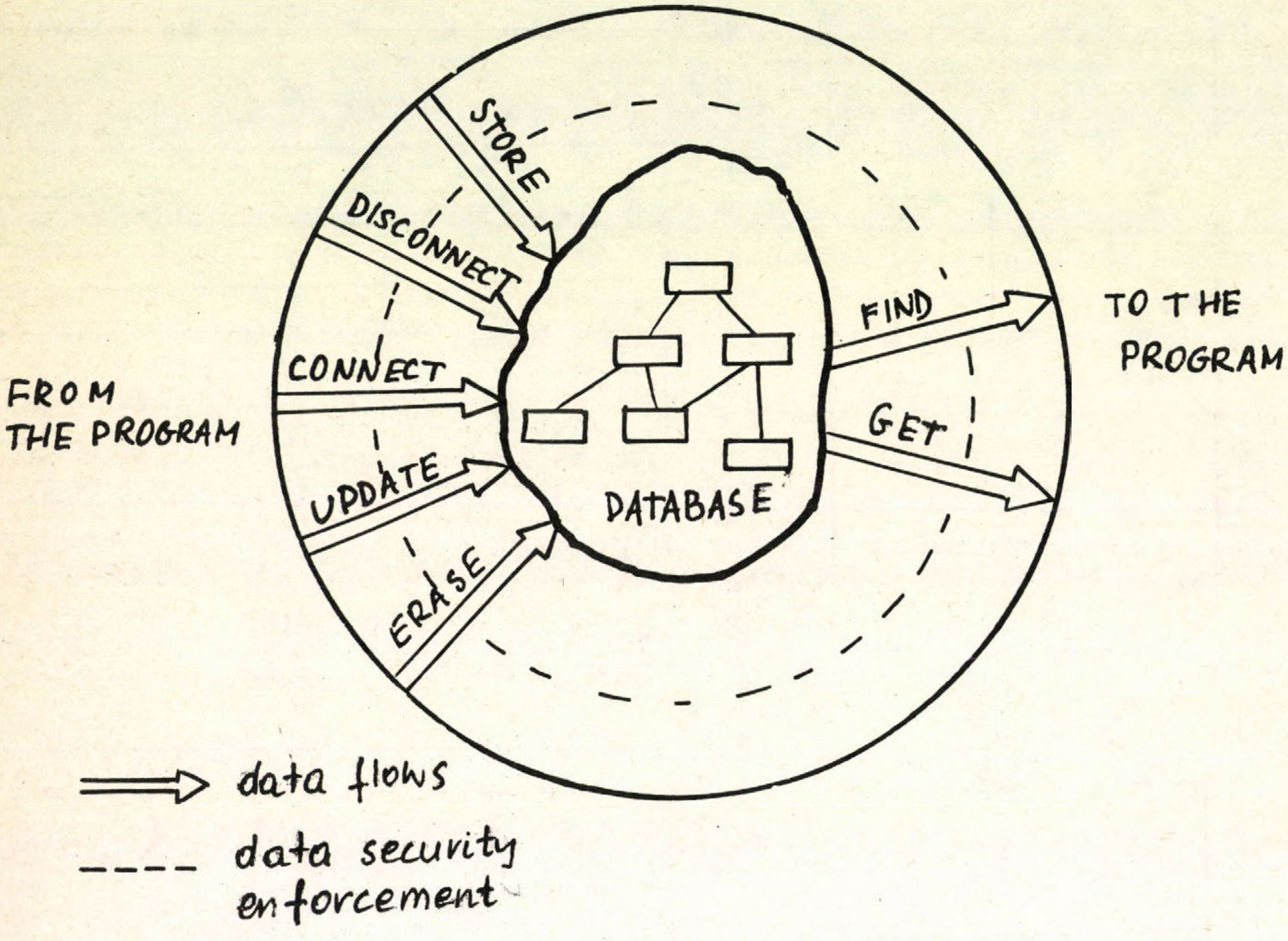

STORE
DISCONNECT
CONNECT
UPDATE
ERASE
FROM
THE PROGRAM
FIND
GET
TO THE
PROGRAM
DATABASE
data flows
data security
enforcement

Fig. 5.3

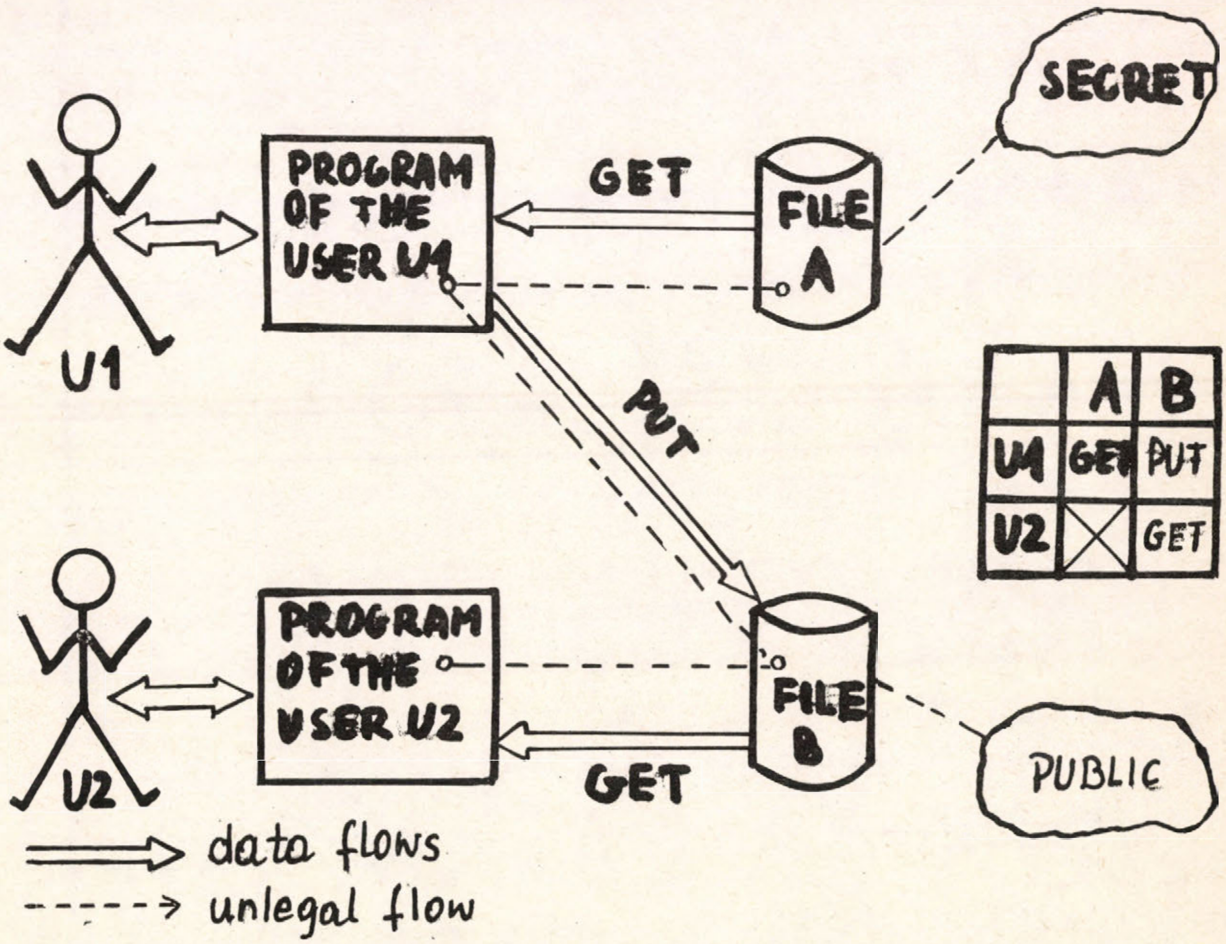

SECRET
PROGRAM OF THE USER U1
GET
FILE A
U1
PUT
| | A | B |
U1 GET PUT
U2 GET
PROGRAM OF THE USER U2
FILE B
PUBLIC
U2
GET
data flows
unlegal flow

Fig. 5.4

I. Construction of global set

$$X = \left(\bigcup_{i=1}^{I} X_i\right) \cup D \cup G$$

where:

$$G = \begin{cases} \emptyset & \text{- if in } \bigcup_{i=1}^{I} X_i \text{ exists the element which satisfy condition of universal upper bound in creating relation.} \\ x_g & \text{- in other case, } x_g \notin \bigcup_{i=1}^{I} X_i. \end{cases}$$

$$D = \begin{cases} \emptyset & \\ & \text{as above} \\ x_d & \end{cases}$$

II. Construction of global relation

$$\bigwedge_{x,y \in X} (x \delta y) \Longleftrightarrow \Big(\bigvee_{\substack{\{z_n\} \subset X \\ n=1,2,\ldots,N}} \; \bigwedge_{n=2,3,\ldots,N} \; \bigvee_{i \in \{1,2,\ldots,I\}} (z_{n-1} \delta_i z_n) \wedge$$

$$\wedge (z_1 = x) \wedge (z_N = y) \vee (x = x_d) \vee (y = x_g) \Big).$$

Fig. 5.5

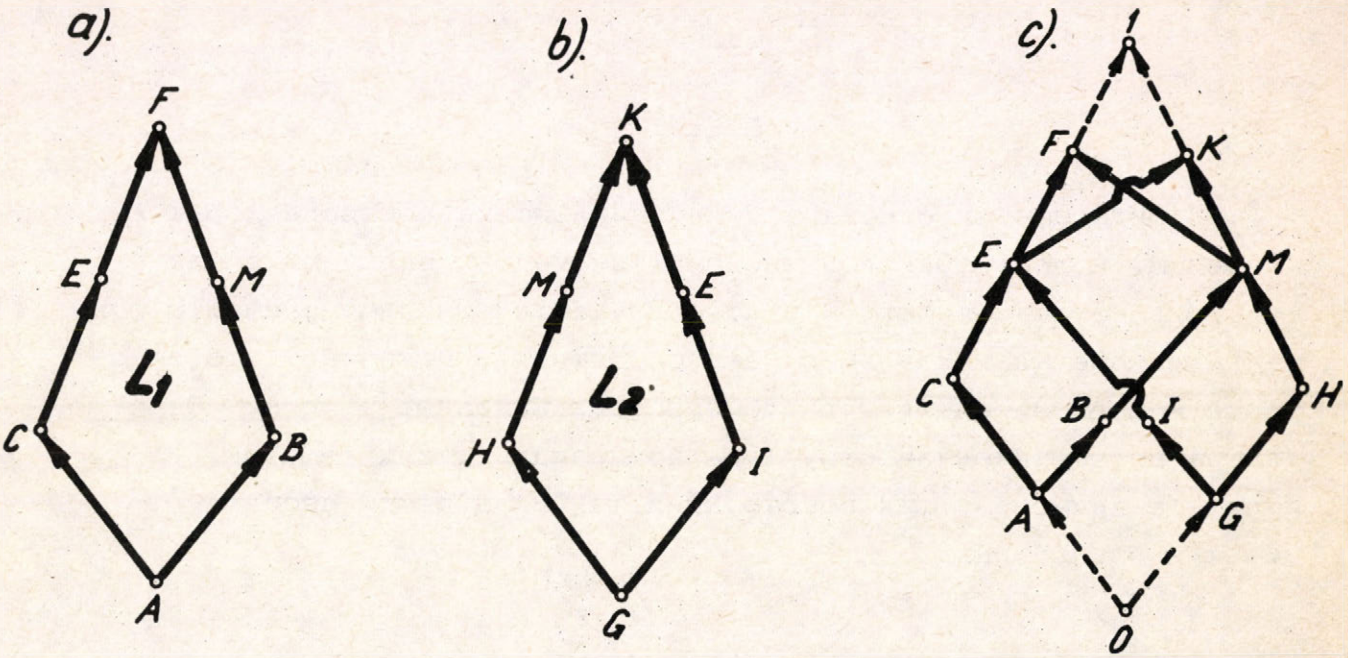


Fig. 5.6

Математические модели процессов защиты данных в централизованных и разделенных системах баз данных

Б. Шафрански

Резюме

Защита данных основана принципиально на контроле добытия к данным. Более эффективными являются методы включающие также контроль потока данных. В статье изучаются главные элементы формальных моделей механизмов такого контроля. Показано, что некоторые модели определяют равномерно ограниченные решетки. Определена алгебраическая структура композиций таких решеток. Результаты применены для конструкции так называемой "security" решетки баз данных.

KÖZPONTOSITOTT ÉS ELOSZTOTT ADATBÁZIS RENDSZEREKRE VONATKOZÓ ADAT-BIZTONSÁGI ELJÁRÁSOK MATEMATIKAI MODELLJEI

B. Szafranski

Összefoglaló

Az adat-védelem általában az adat-elérés ellenőrzésén alapszik. Biztonságosabbak azok a módszerek, amelyek az adat-mozgás folyamatait is ellenőrzik. A cikkben az utóbbi ellenőrzéseket is magába foglaló mechanizmusok formális modelljeinek elemei találhatók. A szerző megmutatja, hogy bizonyos esetekben a modellek egy egyenletesen korlátos hálót alkotnak, és megvizsgálja az igy kapott háló algebrai strukturáját. Az eredményeket az adat-bázisok u.n. biztonsági hálójának konstruálásához használja.

MTA SZTAKI Közlemények 36/1987 pp.145-172

О СЛУЧАЙНЫХ ЗАВИСИМОСТЯХ ДАННЫХ В БАЗАХ ДАННЫХ РЕЛЯЦИОННОГО ТИПА

Тронг Н.
Институт Кибернетики
Ханой- Вьетнам

В этой статье мы попытаемся поставить задачу изучать явленеие часто возникающее в технике данных. Это явление мы здесь назовем "явлением взаимовлияний данных" в базах данных. С некоторой точки зрения, можно считать, что все изученные зависимости данных /функциональные, многозначные, сильные, слабые, дуальные .../ есть разновидные взаимовлияния данных. Но важно заметить, что все эти зависимости отражают логические связи в базах данных. В практике же встречается очень много нелогических связей данных. Ниже будет выяснено, что такое "логическая" и "нелогическая" связь.

Статья разделена на две части:

- В первой части мы бегло выясним логичность изученных зависимостей, и поставим общую задачу изучения нелогических связей /взаимовлияний/ данных в базах данных.
- Во второй части мы приведем полное решение поставленной задачи с точки зрения теории информации.

ЧАСТЬ I.

§1. НЕКОТОРЫЕ ФАКТЫ ИЗ ПРАКТИКИ ОБРАБОТКИ ДАННЫХ

Рассмотрим базу данных с множеством атрибутов Ω. Пусть x, y, з, т некоторые элементы множества Ω. Допустим, что семантика их такова:

x : возраст со значениями x^1: молодой
x^2: средний
x^3: старый

y : семейное положение с y^1: никогда не был в браке
y^2: хоть раз был в венчании

з : число детей с	$з^1$: нет детей
	$з^2$: один
	$з^3$: больше одного
т : пол с	$т^1$: мужской
	$т^2$: женский

Очевидно, что в реальном мире вообще не существует ни каких функциональных зависимостей между этими атрибутами. Но вполне реально такое утверждение:

ЭТИ АТРИБУТЫ "КАК-ТО" И "КАКИМ-ТО ОБРАЗОМ" ОКАЗЫВАЮТ ВЛИЯНИЕ ДРУГ НА ДРУГА.

Что такое это взаимовлияние?

Каковы закономерности этих взаимовлияний?

Изученные знакомые зависимости в некоторой мере уже отвечают на эти вопросы.

Например, теория функциональных зависимовтей дает нам такой закон:

Если х "очень сильно повлияет" на у /так что х функционально определяет у!/
и у "очень сильно повлияет" на з,
то х "очень сильно повлияет" на з!

Но если влияния х на у и у на з не так "сильны", то что можно сказать о влиянии х на з?

В практике обработки данных еще не редко поступают так: Если семантически утверждают, что: $х \not\rightarrow у$ и $у \not\rightarrow х$ /1.1/, то работают с х пренебрегая у и наоборот! В конце концов получаются многочисленные неприятности. Глубокая причина этих аномалий заложена в том, что хотя имеется /1.1/, но вообще х и у взаимовлияют между собой. Это взаимовлияние определяется семантикой данных и оно не позволяет работать с одним, забывая другое. В некотором редком случае это допускается, интуитивно видно, что это и есть случай, когда х и у совсем не повлияют друг на друга.

§2. БУЛЕВЫЕ МАТРИЦЫ И ЛОГИЧЕСКИЕ СВЯЗИ ДАННЫХ

Придерживая обозначения в /2/, мы обозначим:

$A \xrightarrow{f} B$: в случае B функционально зависмо от A

$A \xrightarrow{m} B$: в случае B многозначно зависимо от A /вместо $A \twoheadrightarrow B$, как обозначено в /3//

$A \xrightarrow{s} B$: в случае B сильно зависимо от A

$A \xrightarrow{w} B$: в случае B слабо зависимо от A

$A \xrightarrow{d} B$: в случае B дуально зависимо от A

Для каждой из этих зависимостей мы составляем соответственную булевую матрицу. Опуская общее описание, мы рассмотрим все это на некоторых примерах.

Пусть $A, B \subset \Omega$.

ПРИМЕР 1. Случай $A \xrightarrow{f} B$ /2.1/

Допустим $A = \{x\}$, $B = \{y\}$

Конечно практически никогда и нигде семантически не допускается, чтобы имело место /2.1/. Но все-же пусть имеем /2.1/! Это соотношение соответствует семейству булевых матриц, каждая из них дает конкретный вид /или облик/ функциональной зависимости B от A. Пример одной из них

	y^1	y^2
x^1	1	0
x^2	1	0
x^3	0	1

Теперь пусть $B \xrightarrow{f} C$, где $C = \{з\}$

Допустим, что имеем:

$$\begin{array}{c} \\ y^1 \\ y^2 \end{array} \begin{array}{c} \begin{array}{ccc} з^1 & з^2 & з^3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \qquad /2.3/$$

Обозначим соответственно матрицы /2.2/, /2.3/ через P(y/x) и P(з/y). Известно, что:

$$A \xrightarrow{f} B; \quad B \xrightarrow{f} C \Longrightarrow A \xrightarrow{f} C.$$

Соответственно:

$$P(з/x) = P(y/x) \cdot P(з/y) \qquad /2.4/$$

Где в /2.4/ мы понимаем, что правая часть есть обычное умножение матриц с заменением алгебраических сложения и умножения на логические.

ПРИМЕР 2. Случай $A \xrightarrow{m} B$

Сначало приведем пример булевой матрицы, которая отражает зависимость слегка отличающейся от многозначной зависимости в знакомом смысле. Лучше назвал бы такую матрицу матрицей для "потенциально" многозначной зависимости.

$$\begin{array}{c} \\ x^1 \\ x^2 \\ x^3 \end{array} \begin{array}{c} \begin{array}{cc} y^1 & y^2 \end{array} \\ \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array} \qquad /2.5/$$

В этом случае для соответствия зависимости $A \xrightarrow{m} B$ должно было бы рассмотреть какую-то последовательность булевых матриц вместе как одной целой. Например:

$$\begin{array}{c} \\ x^1 \\ x^2 \\ x^3 \end{array} \begin{array}{c} \begin{array}{cc} y^1 & y^2 \end{array} \\ \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array} \qquad \begin{array}{c} \\ x^1 \\ x^2 \\ x^3 \end{array} \begin{array}{c} \begin{array}{cc} y^1 & y^2 \end{array} \\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array} \qquad /2.6/$$

ПРИМЕР 3. Случай $A \xrightarrow{s} B$

Как высшее, мы тоже приведем только пример для s, w, d случаи, но остановимся чуть на d-зависимостях с целью еще раз иллюстрировать логичность этих зависимостей.

Допустим $A = \{x, т\}$ и $B = \{y, з\}$

и $xт \xrightarrow{s} yз$. Одна из множества матриц может быть такой

	$y^1з^1$	$y^1з^2$	$y^1з^3$	$y^2з^1$	$y^2з^2$	$y^2з^3$
$x^1т^1$	0	1	0	0	0	0
$x^1т^2$	0	1	0	0	0	0
$x^2т^1$	0	0	1	0	0	0
$x^2т^2$	0	0	1	0	0	0
$x^3т^1$	1	0	0	0	0	0
$x^3т^2$	1	0	0	0	0	0

ПРИМЕР 4. Случай $A \xrightarrow{w} B$

	$y^1з^1$	$y^1з^2$	$y^1з^3$	$y^2з^1$	$y^2з^2$	$y^2з^3$
$x^1т^1$	1	0	0	1	0	0
$x^1т^2$	0	1	0	0	1	0
$x^2т^1$	0	1	0	0	1	0
$x^2т^2$	0	0	1	0	0	1
$x^3т^1$	0	0	1	0	0	1
$x^3т^2$	0	0	1	0	0	1

ПРИМЕР 5. Случай А $\xrightarrow{d}$ В

	$у^1з^1$	$у^1з^2$	$у^1з^3$	$у^2з^1$	$у^2з^2$	$у^2з^3$
$х^1т^1$	1	1	1	0	0	0
$х^1т^2$	0	0	0	1	1	1
$х^2т^1$	1	1	1	0	0	0
$х^2т^2$	0	0	0	1	1	1
$х^3т^1$	1	1	1	0	0	0
$х^3т^2$	0	0	0	1	1	1

/2.7/

Допустим, что еще имеем В $\xrightarrow{d}$ С, где С = {з, р} $\subset \Omega$ /2.8/. Как в случае функциональных зависимостей было замечено, что /2.8/ соответствует множеству булевых матриц вида /2.7/. Рассмотрим, например, две из них:

	$з^1р^1$	$з^1р^2$	$з^2р^1$	$з^2р^2$	$з^3р^1$	$з^3р^2$
$у^1з^1$	1	0	1	0	1	0
$у^1з^2$	1	0	1	0	1	0
$у^1з^3$	1	0	1	0	1	0
$у^2з^1$	0	1	0	1	0	1
$у^2з^2$	0	1	0	1	0	1
$у^2з^3$	0	1	0	1	0	1

/2.9/

	$з^1р^1$	$з^1р^2$	$з^2р^1$	$з^2р^2$	$з^3р^1$	$з^3р^2$
$у^1з^1$	1	0	1	0	1	0
$у^1з^2$	1	0	1	0	1	0
$у^1з^3$	0	1	0	1	0	1
$у^2з^1$	1	0	1	0	1	0
$у^2з^2$	1	0	1	0	1	0
$у^2з^3$	0	1	0	1	0	1

/2.10/

Как известно, что А $\xrightarrow{d}$ В и В $\xrightarrow{d}$ С $\Longrightarrow$ А $\xrightarrow{d}$ С.

Тонкость в определении дуальных зависимостей показывает, что нужно понимать закон транзитивности в этом случае именно при умножении матриц /логические сложение и умножение вместо алгебраических/ /2.7/ с /2.9/, а не с /2.10/.

На этом мы заканчиваем беглую иллюстрацию логичности некоторых рассматриваемых зависимостей данных в базах данных и переходим к нашей основной цели.

§3. МАТРИЦА ВЛИЯНИЙ /МВ/ АТРИБУТОВ ДАННОГО МНОЖЕСТВА НА АТРИБУТЫ ДРУГОГО МНОЖЕСТВА

Пусть А, В какие то множества атрибутов базы данных *R*, т.е.

$$A,\ B \subset \Omega\ .$$

Из данных в *R* мы можем составить разные таблицы.

Например: в *R* сохраняются данные 100 человек и таблица "бракосочетание по возрасту" такова:

	y^1 не был в браке	y^2 хоть раз был
молодой (x^1)	15	25
средний (x^2)	10	20
старый (x^3)	5	25

Видно, что мы можем говорить о таких вероятностях, например, вероятность того, что человек не был ни разу в браке при условии, что он молодой. Обозначим эту вероятность как обычно через $p(y^1/x^1)$. В нашем примере примерно $p(y^1/x^1) = \frac{3}{8}$.

Обозначим матрицу этих вероятностей /условно/ через $P(y/x)$. Эта матрица называется МВ атрибута x на атрибут y. Легко обобщить на общий случай и мы получаем МВ атрибутов в А на атрибуты в В обозначаемую через

$$P(B/A).$$

Пример: $A = \{x,\ т\}$; $B = \{y,\ з\}$

$$\begin{array}{r|cccccc|} & \text{у}^1\text{з}^1 & \text{у}^1\text{з}^2 & \text{у}^1\text{з}^3 & \text{у}^2\text{з}^1 & \text{у}^2\text{з}^2 & \text{у}^2\text{з}^3 \\ x^1\text{т}^1 & & & \vdots & & & \\ x^1\text{т}^2 & & & \vdots & & & \\ P(B/A)=x^2\text{т}^1 & \cdots & \cdots & p(\text{у}^1\text{з}^3/x^2\text{т}^1) & & & \\ x^2\text{т}^2 & & & & & & \\ x^3\text{т}^1 & & & & & & \\ x^3\text{т}^2 & & & & & & \end{array}$$

Стохастическая матрица P(B/A) изображает полную картину влияний A на B. Важность изучения этих влияний /т.е. этих матриц/ несомнена. Под случайную зависимоть множества атрибутов B от множества A мы понимаем именно МВ P(B/A).

Общая задача изучения связей данных в базах данных будет такой:

Пусть дано семейство МВ

$$\mathcal{P} = \{P(B_i/A_i) \mid A_i,\ B_i \subset \Omega\ ;\quad i = \overline{1, k}\}$$

/или известны какие-то характеристики этих матриц/.

И пусть X, У какие-то множества атрибутов базы данных, т.е X, У $\subset \Omega$.

Судя по структурам X, У, A_i, B_i $(i = \overline{1, k})$

ЧТО МОЖНО СКАЗАТЬ О МАТРИЦЕ P(У/X)?

/Понимается, например, в таких смыслах:

- Вычисляется ли P(У/X) по $P(B_i/A_i)$ $i = \overline{1, k}$?

  Вычисляется ли или оценивается ли какие то характеристики матрицы P(У/X) по характеристикам $P(B_i/A_i)$?
- Да, еще непротиворечива ли сама система $\mathcal{P}$?
  и т.д.

Рассмотрим простой пример: Пусть дана система $\mathcal{P}$:

$$\mathcal{P} = \{\ P(B/A),\ P(C/B)\}$$

/т.е. даны МВ A на B и B на C/. Пусть X = A и У = C.

Что можно сказать о матрице P(У/X) , т.е. о матрице P(C/A)? В этом случае P(C/A) "достаточно определена" по матрицам

P(B/A) и P(C/B)

/в смысле показанном ниже/.

Но врядли что-нибудь стóящее можно сказать о MB P(A/C).

Конечно изучение зависимости /f, m, s, w, d/ дают нам не мало для изучения случайных зависимостей. Об этом здесь мы не будем говорить, а будем рассматривать задачу в целом с точки зрения теории информации, что на наш взгляд имеет глубокое значение в семантической целостности данных в базах данных.

ЧАСТЬ II

Нумерация этой части продолжает нумерацию первой части.

§4. МЕРА СЛУЧАЙНЫХ ЗАВИСИМОСТЕЙ ИЛИ ОТНОСИТЕЛЬНАЯ СВОБОДНОСТЬ /ОС/ НЕКОТОРОГО МНОЖЕСТВА АТРИБУТОВ ОТ ЗАДАННОГО МНОЖЕСТВА АТРИБУТОВ

Пусть A, B некоторые множества атрибутов, т.е. $A, B \subset \Omega$. Вместо P(B/A) мы рассмотрим некоторую характеристику этой матрицы, а именно условную энтропию B при условии A. Обозначим эту величину через [A, B] /обычно в теории информации эту величину обозначают так H(B/A)/. Явная формула вычисления [A, B] по P(B/A) написана в любых учебниках по теории информации.

Мы назовем величину [A, B] мерой случайной зависимости B от A или Относительной Свободностью /ОС/ B относительно A. В дальнейшем предпочтем названию ОС, так как, с одной стороны, оно отражает семантику введенного понятия, с другой стороны, видно, что полезно посмотреть и другие меры случайной зависимости.

Мы чувствуем, что "свободность" B от A уменьшается по мере уменьшения ОС [A, B], и ожидаем, что B становится "полностью зависимым" от A при минимальном значении [A, B] и B будет "полностью свободным" от A при максимальном значении [A, B]. Как известно из теории информации:

$$\text{Min}\,[A,\ B] = 0 \qquad /4.1/$$

$$\text{Max}\,[A,\ B] = [\Lambda,\ B] \qquad /4.2/$$

где обозначение $[\Lambda,\ B]$ выражает безусловную энтропию B, или можно считать это есть условная энтропия B при условии достоверного события Λ /событие с одним исходом/. Соотношение /4.1/ есть реализация следующей простой, но важной теоремы:

ТЕОРЕМА 1.

$$A \xrightarrow{f} B \Longleftrightarrow [A,\ B] = 0 \qquad /4.3/$$

Доказательство этой теоремы имеется в литературах по теории информации с учетом того, что МВ P(B/A) в этом случае является булевой матрицей для функциональных зависимостей.

Соотношение /4.2/ говорит о том, что:

при обработке данных, несмотря на то, что имеется $A \overset{f}{\not\to} B$ и $B \overset{f}{\not\to} A$ /напомним здесь соотношение /1.1/ и, что было сказано про него/ все-таки не допускается относиться к A и B как к независимым существам, при этом не ожидая грубых ошибок! Мы можем "работать" с A без всяких забот о B /и наоборот/, если имеет место по крайней мере одно из двух /выводимых друг из друга/ соотношений:

$$[A,\ B] = [\Lambda,\ B] \quad \text{или}$$

$$[B,\ A] = [\Lambda,\ A] \qquad /4.4/$$

Теорема 1 показывает, что изучение функциональных зависимостей является изучением случайных зависимостей с нулевой мерой, или мы еще говорим, что функциональная зависимость является вырожденным случаем случайной зависимости.

§5. ОСНОВНЫЕ ЗАДАЧИ ДЛЯ ОТНОСИТЕЛЬНЫХ СВОБОДНОСТЕЙ

5.1. ОС вычисляемая и ОС оцениваемая по заданной системе ОС

Для ясного представления всего сказанного здесь мы немного вспомним об аналогичной ситуации в теории функциональных зависимостей.

Пусть дана система F функциональных зависимостей /вместо $A \xrightarrow{f} B$ мы пишем $[A, B]$, временно забывая об истинном значении величины $[A, B]$/

$$F = \{[A_i, B_i] \mid A_i, B_i \subset \Omega; \quad i = \overline{1, k}\} \qquad /5.1/$$

Функциональная зависимость $[X, У]$ /т.е. $X \xrightarrow{f} У$ / называется вытекающей из F , если:

- /По методу выражения в теории функциональных зависимостей/ любая реализация схемы удовлетворяющая F должна удовлетворять и $[X, У]$.
- /На языке случайных зависимостей/ для любой реализации схемы, если $[A_i, B_i] = 0; \quad i = \overline{1, k}$, то и $[X, У] = 0$.

Рассмотрим некоторую систему ОС:

$$E = \{[A_i, B_i] \mid A_i, B_i \subset \Omega; \quad i = \overline{1, k}\} \qquad /5.2/$$

ОПРЕДЕЛЕНИЕ 1.

ОС $[X, У]$ называется вычислимой по E если существует функция f от переменных $[A_i, B_i]$ такая, что любая реализация удовлетворяющая E должна удовлетворить и $[X, У]$, где

$$[X, У] = f([A_i, B_i] \quad i = \overline{1, k}) \qquad /5.3/$$

Обозначим множество всех вычислимых ОС по заданной системе E через Cal(E).

Рассмотрим пример:

Допуская некоторую вольность, вместо $A = \{x, y\}$ мы будем писать здесь $A = xy$. Это обозначение уже знакомо при изучении других зависимостей. Здесь оно еще имеет значение умножения случайных переменных, что будет видно ниже.

И так, пусть $\Omega = \{x_1, x_2, \ldots, x_n\}$ и пусть

$$E = \{[x_1x_2, x_3] \quad , \quad [x_1x_2x_3 \, , x_4x_5]\}$$

Тогда можно доказать, что: $[x_1x_2\ ,\ x_3x_4x_5] \in Cal(E)$, а именно:

$$[x_1x_2\ ,\ x_3x_4x_5] = [x_1x_2\ ,\ x_3] + [x_1x_2x_3\ ,\ x_4x_5] \quad /5.4/$$

в то время $[x_1\ ,\ x_2x_3] \notin Cal(E)$.

ВАЖНОЕ ЗАМЕЧАНИЕ:

Перед тем как проводить дальнейшее рассуждение, мы сделаем важное замечание о равенствах и неравенствах написанных из этого параграфа этой статьи. Для тех, кто понимает глубоко смысл группы слов "свойства справедливы на степени реляционной схемы ..." это замечание не нужно. Все-таки мы сделаем раз на всю статью замечание. Изучая случайные зависимости данных в базах данных мы считаем реляционную схему с множеством атрибутов Ω как схему, где элементы Ω являются случайными переменными. Реализация схемы получается при назначении случайным переменным конкретными случайными величинами. Вычислимость $[X, У]$ по E есть свойство, которое сраведливо на степени схемы. Это означает, что равенства вида /5.3/, /5.4/ /и ниже встречающиеся неравенства/ есть тождественные равенства при любых назначениях случайным переменным присутствующих в этих соотношениях любыми конкретными случайными величинами, иначе они справедливы при любых реализациях схемы. Например, при одной реализации схемы мы имеем: $[x_1x_2\ ,\ x_3] = 2$

$$[x_1x_2x_3\ ,\ x_4x_5] \neq 3$$

то для этой реализации $[x_1x_2\ ,\ x_3x_4x_5]$ не может равняться 5, но если в другой реализации $[x_1x_2x_3\ ,\ x_4x_5] = 3$, то теперь обязательно $[x_1x_2\ ,\ x_3x_4x_5] = 5$!

ОПРЕДЕЛЕНИЕ 2.

ОС $[X, У]$ оценимой по E, если существует функция φ от переменных $[A_i, B_i]$ такая, что любая реализация удовлетворяющая E должна и удовлетворять $[X, У]$, где

$$[X, У] \leq \varphi([A_i, B_i] \quad i = \overline{1, k}) \qquad /5.5/$$

Обозначим множество оценимых ОС по заданной системе E через Est(E).

Пример: пусть $E = \{[x_1, x_2], [x_1x_2, x_3]\}$, тогда $[x_1, x_3] \in Est(E)$, а именно:

$$[x_1, x_3] \leq [x_1, x_2] + [x_1x_2, x_3] \qquad /5.6/$$

в то время $[x_2, x_3] \notin Est(E)$.

5.2. Основные задачи:

В этой статье мы рассмотрим только две основных задач из множества проблем, связанных с рассмотрением любого вида зависимости данных в базах данных.

ЗАДАЧА 1.

Для любой E, найти множество Cal(E). Вместе с этим для любых $X, Y \subset \Omega$ определить $[X, Y] \in Cal(E)$ или нет? В случае "ДА" найти функцию f , чтобы имелось /5.3/.

ЗАДАЧА 2.

Для любой системы E, найти множество Est(E). Вместе с этим для $X, Y \subset \Omega$ определить $[X, Y] \in Est(E)$ или нет? В случае "ДА" найти функцию φ, чтобы имелось /5.5/.

Из сделанного выше замечания, мы сформулируем два следующих преложения:

ПРЕДЛОЖЕНИЕ 1.

Если $[X, Y] \in Cal(E)$, то при любых двух назначениях случайным переменным конкретными случайными величинами, если $[A_i, B_i]$ неменяются, то $[X, Y]$ тоже неизменится. А если $[X, Y] \notin Cal(E)$, то существуют по крайней мере два разных назначения, при которых $[A_i, B_i]$ не меняются, а $[X, Y]$ изменится.

ПРЕДЛОЖЕНИЕ 2.

Если $[X, Y] \in Est(E)$, то при всех назначениях случайным переменным конкретными случайными величинами так чтобы:

$$[A_i, B_i] \le C_i \quad i = \overline{1, k} \qquad /5.7/$$

где C_i заданные константы, существует константа C такая, что

$$[X, У] \le C \qquad /5.8/$$

А если $[X, У] \notin Est(E)$, существует по крайней мере такое назначение, при котором /5.7/ удовлетворено, в то время $[X, У] > C$, где C любая заданная константа.

§6. СИСТЕМА НЕВЫРОЖДЕННЫХ ПРАВИЛ ВЫВОДОВ АРМСТРОНГ И ЗАДАЧА НАХОЖДЕНИЯ Est(E)

Термин "невырожденное" объясняется тем, что знакомая система аксиом Армстронг в теории функциональных зависимостей решила поставленные здесь задачи 1 и 2, когда все ОС в системе E равны нулю. Мы исследуем общий случай допускающий $[A_i, B_i] \in E$ принимать любые значения.

Здесь приведем правила, в следующих пунктах докажем полноту этой системы правил.

6.1. Система невырожденных правил выводов Армстронг:

В системе включаются три группы правил:

Группа 2 основных правил: $\forall$ A, B, C $\subset \Omega$

1. $[A, C] \le [A, B] + [AB, C]$ /6.1/

2. $[A, BC] \ge [A, B] + [AB, BC]$ /6.2/

Группа 3 естественных правил: $\forall$ A, B, C, D

3. $[AB, CD] = [BA, CD] = [AB, DC] = [BA, DC]$ /6.3/

4. $[A\Lambda, B\Lambda] = [A, B]$ /6.4/

5. $[AA, BB] = [A, B]$ /6.5/

Группа алгебраических операций, например, для чисел *A*, *B*, *C*:

6. $A \le B \;\&\; B \le A \Longrightarrow A = B$

7. $A \leq B \ \& \ B \leq C \Longrightarrow A \leq C$

.....

Вообще допускается пользоваться всеми алгебраическими операциями по мере нужности.

6.2. Некоторые производные правила:

Нетрудно доказать многие полезные производные правила выводов, мы перечислим некоторые самые важные из них с продолжающейся нумерацией.

8. $[A, BC] = [A, B] + [AB, BC]$

9. $[A, \Lambda] = 0$

10. $[A, B] \geq 0$

11. $[AB, B] = 0$

12. $[AC, B] \leq [A, B]$

13. $[AC, BC] = [AC, B]$

14. $[A, BC] \geq [A, B]$

15. $[A, BC] = [A, B] + [AB, C]$

16. $[\Lambda, AB] = [\Lambda, A] + [A, B]$

17. $[A, C] \leq [A, B] + [B, C]$

18. $[A, BC] \leq [A, B] + [A, C]$

19. $[\Lambda, AB] \leq [\Lambda, A] + [\Lambda, B]$

20. $|[\Lambda, A] - [\Lambda, B]| \leq \max \{[A, B], [B, A]\}$

. . .

Все выписанные основные и производные правила выводов в сущности есть не что иное, как более или менее знаковые свойства энтропий. Они являются тождественными соотношениями между энтропиями.

6.3. ОС выводимая с системы Е по Армстронг:

Пусть дана система Е /5.2/.

ОПРЕДЕЛЕНИЕ 3.

ОС $[X, Y]$ называется выводимой с E по Армстронг /кратко назовем выводимой/, если после конечных шагов применений невырожденных правил выводов Армстронг /конечно возможно и производных правил/ получается функция Φ от переменных $[A_i, B_i]$ такая, что

$$[X, Y] \leq \Phi \{[A_i, B_i] \ i = \overline{1, k}\} \qquad /6.6/$$

Множество всех выводимых ОС с E обозначим через Ded(E).

ТЕОРЕМА 2.

Если $[X, Y] \in Ded(E)$, то существуют числа $\alpha_1, \alpha_2, \ldots, \alpha_k$ такие, что

$$\Phi = \sum_{i=1}^{k} \alpha_i [A_i, B_i] \qquad /6.7/$$

Доказательство:

Просто, что все правила выводов имеют линейный вид. Если вместо /6.6/ получается равенство, т.е.

$$[X, Y] = \Phi ([A_i, B_i] \ i = \overline{1,k}) \qquad /6.8/$$

то $[X, Y]$ называется точно выводимой. Множество всех точно выводимых ОС с E обозначим через Dedex(E).

6.4. Структура множества Ded(E):

Полученные результаты в этом пункте совпадают с результатами в теории Армстронг для функциональных зависимостей.

Пусть $X \subset \Omega$. Назовем замкнутым замыканием X относительно E множество X_E^+ определяемое следующим образом:

$$X_E^+ = \{x_i \in \Omega : [X, x_i] \in Ded(E)\} \qquad /6.9/$$

ТЕОРЕМА 3.

$$[X, Y] \in Ded(E) \Longleftrightarrow Y \subset X_E^+ \qquad /6.10/$$

Доказательство:

Докажем $[X, У] \in Ded(E) \Longrightarrow Y \subset X_E^+$ /6.11/

Пусть $У = x_{i_1} x_{i_2} \dots x_{i_t}$, где $x_{i_1}, x_{i_2}, \dots, x_{i_t}$ элементы Ω. Так как $[X, У] \in Ded(E)$, следовательно существует процедура конечных шагов применений правил выводов чтобы

$$[X, У] \le \Phi\ ([A_i, B_i]\ i = \overline{1, k}) \qquad /6.12/$$

Применим производный закон 14, получаем

$$[X, x_{i_j}] \le [X, У] \le \Phi([A_i, B_i]\ i=\overline{1, k})\ j=\overline{1, t} \qquad /6.13/$$

Значит $[X, x_{i_j}] \in Ded(E)$ $j = \overline{1, t}$, следовательно $x_{i_j} \in X_E^+$ и получаем /6.11/.

Обратно покажем $У \subset X_E^+ \Longrightarrow [X, У] \in Ded(E)$ /6.14/

$У \subset X_E^+$ следовательно $x_{i_j} \in X_E^+$ $j = \overline{1, t}$.

Это означает, что после конечных шагов /вообще число шагов разное для каждого j/ применений правил выводов получется функция Φ_j, такая, что

$$[X, x_{i_j}] \le \Phi_j([A_i, B_i]\ i = \overline{1, k})\ \ j = \overline{1, t} \qquad /6.15/$$

Применяя правило 15 несколько раз, получим

$$[X, У] \le \sum_{j=1}^{t} [X, x_{i_j}] \le \sum_{j=1}^{t} \Phi_j([A_i, B_i]\ \ i = \overline{1, k})$$

Так что $[X, У] \in Ded(E)$.

ТЕОРЕМА 4. /алгоритм определения X_E^+/

Определим последовательность $X^{(0)}, X^{(1)}, X^{(2)}, \dots$ следующим образом $X^{(0)} = X$. Если $X^{(i)}$ определено, то

$$X^{(i+1)} = X^{(i)} \cup \Big(\bigcup_{j : A_j \subset X^{(i)}} B_j \Big)$$

Мы имеем: $X^{(0)} \subset X^{(1)} \subset \dots \subset \Omega$. Существует самое малое i такое, что $X^{(i)} = X^{(i+1)} = \dots$ Это $X^{(i)}$ и есть X_E^+.

Алгоритм доказывается совершенно аналогично, как в теории функциональных зависимостей.

ТЕОРЕМА 5. /полнота системы невырожденных правил выводов Армстронга для задачи нахождения Est(E)/

$$Ded(E) = Est(E)$$

Доказательство:

Одно включение очевидно, докажем включение

$$Est(E) \subset Ded(E) \qquad /6.16/$$

Пусть $[X, У] \notin Ded(E)$ /6.17/

тогда $У \not\subset X_E^+$ /6.18/

Разделим E на две части E^1 и E^2. Для различения ОС в E^1 и E^2 мы припишем соответственно индексы 1 или 2. Например, $[A_2, B_2] \in E^1$, будем писать так $[A_2^1, B_2^1]$.

Отнесем к E^1 все ОС с левой частью входящейся в X_E^+, т.е.

$$E^1 = \{[A_i^1, B_i^1]\ \forall i : A_i \subset X_E^+\} \qquad /6.19/$$

тогда и все $B_i^1 \subset X_E^+$.

В E^2 входят все остальные ОС в системе E. Это значит, что если существует какая-то ОС $[A_i^2, B_i^2]$, то $A_i^2 \not\subset X_E^+$.

Заметим, что из правила 13 всегда можем считать, что для любой ОС нет общих переменных в правой и в левой части. Из /6.18/ имеем такую структуру $У : У = У'g$, где $g \notin X_E^+$ /6.20/.

Рассмотрим два случая: E^2 пустое и E^2 не пустое.

Первый случай: E^2 пустое. Это означает, что все переменные присутствующие в E принадлежат X_E^+. В этом случае можно поставить значение Λ всем этим случайным переменным и получается, что

$[A_i, B_i] = 0 \quad i = \overline{1, k}$. В то время из-за /6.20/ легко добиться того, что $[X, У]$ превосходит любой заданной константе, что показывает $[X, У] \notin Est(E)$ /предложение 2/.

Второй случай: E^2 не пустое.

Для всех переменных в E^1 поступаем как в первом случае. В каждом A_i^2 /значит возможно и в некоторых B_i^2/ содержатся переменные не входящие в X_E^+ . Обозначим множество таких переменных через M.

Пусть $M = \{m_1, m_2, \ldots, m_\gamma\}$ /конечно $M \subset \Omega$/. E^2 содержит ОС вида:

$$[A_i'^2 M'_{A_i}, B_i'^2 M'_{B_i}] \text{ , где } M'_{A_i}, M'_{B_i} \subset M \text{ и } M'_{A_i} \cap M'_{B_i} = 0 .$$

и $[A_j'^2 M'_{A_j}, B_j^2]$. Напомним, что $X \cap M = \emptyset$.

Если $\varrho \notin M$, то почти возвращается к первому случаю, т.е. назначить всем переменным кроме ϱ значение Λ, и все получается как в первом случае.

Пусть $\varrho \in M$. Всем переменным кроме тех входящих в M назначим Λ. Для любой заданной константы C можно выбрать случайные величины $c_1, c_2, \ldots, c_\gamma$ такие, что $[\Lambda, c_1 c_2 \ldots c_\gamma] > C$.

Назначим всем $m_j \quad j = \overline{1, \gamma}$ величину $c_1 c_2 \ldots c_\gamma$. Тогда все ОС в E^2 также равняются нулю, в то время X, У

$$[X, У] = [\Lambda, c_1 c_2 \ldots c \quad У'] > C.$$

Теорема доказана.

Мы решили задачу 2, т.е. задача с множеством Est(E). Попутно заметили, что результат дедукции иногда дает равенство вида /6.8/. К сожалению, что методом Армстронг не дает нам выяснения структуры множества Dedex(E), с которым мы надеемся получить ответ для задачи нахождения Cal(E). Другим путем мы покажем, что Cal(E) = Dedex(E) и получим алгоритм построения этих множеств.

§7. ЛИНЕЙНЫЕ КОМБИНАЦИИ ОС И ЗАДАЧА НАХОЖДЕНИЯ Cal(E)

7.1. Линейные комбинации ОС:

Пусть дана система E /5.2/. Составим линейную комбинацию вида:

$$\sum_{i=1}^{k} \alpha_i [A_i, B_i], \text{ где } \alpha_i \; i = \overline{1,k} \text{ вещественные числа} \qquad /7.1/$$

Напомним, что $[A_i, B_i]$ есть некоторая величина /вообще разная для разных реализаций/ вычисляемая по явной формуле от MB $P(B_i/A_i)$. Это значит умножение $\alpha_i [A_i, B_i]$ просто умножение реальных чисел.

Составить комбинации вида /7.1/ можно как угодно! Но видно, что не всякая комбинация дает какую-то ОС /при одной и той же реализации/, т.к. число ОС в данной реализации конечно.

7.2. Линейная зависимость и независимость системы ОС:

ОПРЕДЕЛЕНИЕ 4.

Система E ОС называется линейно-независимой, если из равенства /напомним, что в смысле тождественного равенства относительно всех реализаций/

$$\sum_{i=1}^{k} \alpha_i [A_i, B_i] = 0 \qquad /7.2/$$

вытекает, что $\alpha_i = 0 \quad i = \overline{1, k}$. В противном случае E называется линейно-зависимой.

Обозначим $E_\Omega = \{ [\Lambda, X], \; X \neq \emptyset, \; X \subset \Omega \}$ /7.3/

Например: $\Omega = \{x_1, x_2\}$, тогда

$$E_\Omega = \{ [\Lambda, x_1]; \; [\Lambda, x_2]; \; [\Lambda, x_1 x_2] \}$$

ЛЕММА.

Все возможные соотношения связывающие элементы E_Ω есть:

1. $X^1 \subset X^2 \subset \Omega \Longrightarrow [\Lambda, X^1] \le [\Lambda, X^2]$

2. $X = X^1 \cup X^2$ и $X^1 \cap X^2 = \emptyset$

$$[\Lambda, X] \le [\Lambda, X^1] + [\Lambda, X^2]$$

Лемма есть следствие теоремы 5.

Рассмотрим систему неотрицательных чисел индексируемых подмножествами множества $K = \{1, 2, \ldots, n\}$ вида

$$a_1, a_{12}, a_{134}, \ldots, a_{123\ldots n}$$

Обозначим множества индексов /т.е. подмножества множества K/ через K^j. Пусть эти числа удовлетворяют условиям:

1. $K^1 \subset K^2 \subset K \Longrightarrow a_{K^1} \le a_{K^2}$ /7.4/

2. $K^3 = K^1 \cup K^2, \; K^1 \cap K^2 = \emptyset \Longrightarrow a_{K^3} \le a_{K^1} + a_{K^2}$

Лемма дает, что существует реализация реляционной схемы такая, что

$$[\Lambda, x_{i_1} x_{i_2} \ldots x_{i_t}] = a_{i_1 i_2 \ldots i_t} \qquad /7.5/$$

ТЕОРЕМА 6.

E_Ω есть линейно-независимая система ОС.

Доказательство:

Допустим противно и не потеряя общности пусть

$$\sum_{\forall X \subset \Omega} \alpha_X [\Lambda, X] = 0 \qquad (X \ne \emptyset, \; X \subset \Omega) \qquad /7.6/$$

с коэффициентом при $X = \{x_1\}$ отличным от нуля. Тогда перепишем

$$[\Lambda, x_1] = \sum_{\forall X \ne \{x_1\}} \alpha'_X [\Lambda, X] \qquad /7.7/$$

Не трудно выбрать систему чисел удовлетворяющую условиям /7.4/, /7.5/, но не удовлетворяющую /7.7/.

На минутку вернемся к энтропии, сделаем, например, такое следствие:

СЛЕДСТВИЕ.

Представление условной энтропии через бузесловные по формуле

$$H(B/A) = H(AB) - H(A) \qquad /7.8/$$

есть единственное линейное выражение связывающее их.

Ниже мы увидим, что нет и всяких других /не линейных/ связей между ними.

Задача распознавания зависимости или независимости данной системы ОС решается как в линейной алгебре для системы векторов. Мы сформулируем одну из возможных теорем.

Рассмотрим линейную комбинацию

$$\sum_{i=1}^{k} \alpha_i \, [A_i, B_i] \qquad /7.9/$$

Перепишем /7.9/ на основе правила 16 и сгруппируем слагаемые, получаем

$$\sum_{i=1}^{k} \alpha_i([\Lambda, A_i B_i] - [\Lambda, A_i]) = \sum_{j=1}^{m} \beta_i [\Lambda, X^j], \text{ где } X^j \subset \Omega \qquad /7.10/$$

и

$$\beta_1 = \lambda_{11}\alpha_1 + \lambda_{12}\alpha_2 + \dots + \lambda_{1k}\alpha_k$$

$$\beta_2 = \lambda_{21}\alpha_1 + \lambda_{22}\alpha_2 + \dots + \lambda_{2k}\alpha_k \qquad /7.11/$$

$$\dots$$

$$\beta_m = \lambda_{m1}\alpha_1 + \lambda_{m2}\alpha_2 + \dots + \lambda_{mk}\alpha_k$$

Условие линейно-зависимости системы Е приводит к тому, что система /7.11/ становистя однородной системой алгебраических уравнений с неизвестными α_i $i = \overline{1,k}$, и так

ТЕОРЕМА 7.

Для того, чтобы система E была линейно-независимой необходимо и достаточно, чтобы ранг матрицы λ равнялся k.

В /5/ была доказана теорема:

$$\text{а/ } [X, У] = \sum_{i=1}^{k} \alpha_i [A_i, B_i] \qquad /7.12/$$

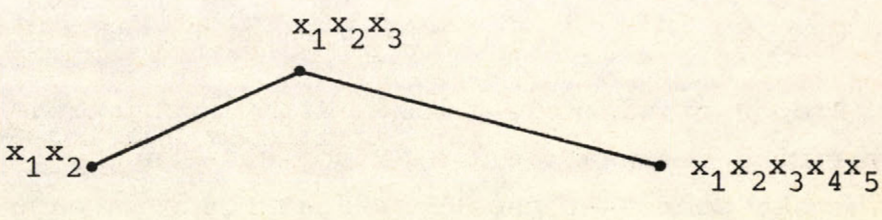
тогда и только тогда, когда система $E \cup |X, У|$ линейно-зависима.

б/ Если имеет место /7.12/, то можно добиться того, чтобы все α_i целые.

в/ Если E независимая система, то α_i принимает только значения -1, 0 или 1.

В /5/ был построен граф соответствующий данной системе E /обозначен через G_E/. Опуская общее описание, рассмотрим один пример.

Пусть $E = \{[x_1x_2, x_3], [x_1x_2x_3, x_4x_5]\}$. Граф G_E будет такой



В /5/ же доказывается такая теорема: Для того, чтобы [X, У] представима линейной комбинацией ОС в E /т.е. имеет место /7.12// необходимо и достаточно, что X и XУ являются связными вершинами графа G_E .

Выше было написано соотношение /5.4/, что видно на нарисованном графе.

ТЕОРЕМА 9.

/Полнота линейных комбинаций для задачи нахождения множества Cal(E)./

$$[X, У] \in Cal(E) \Longleftrightarrow [X, У] = \sum_{i=1}^{k} \alpha_i [A_i, B_i] \qquad /7.13/$$

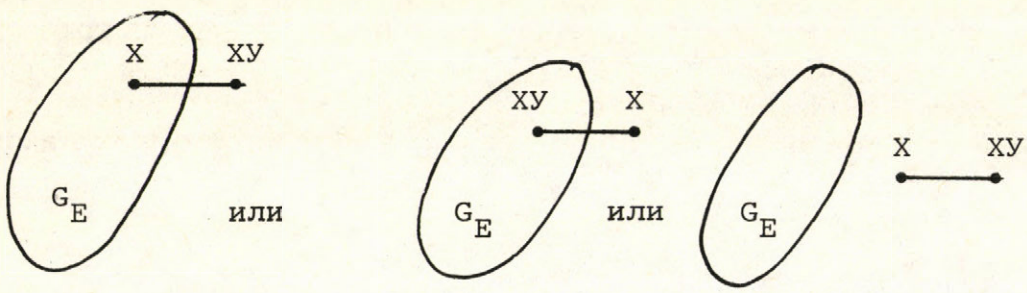
Доказательство:

Мы кратко иллюстрируем доказательство этой теоремы на языке графа. Нужно доказать только

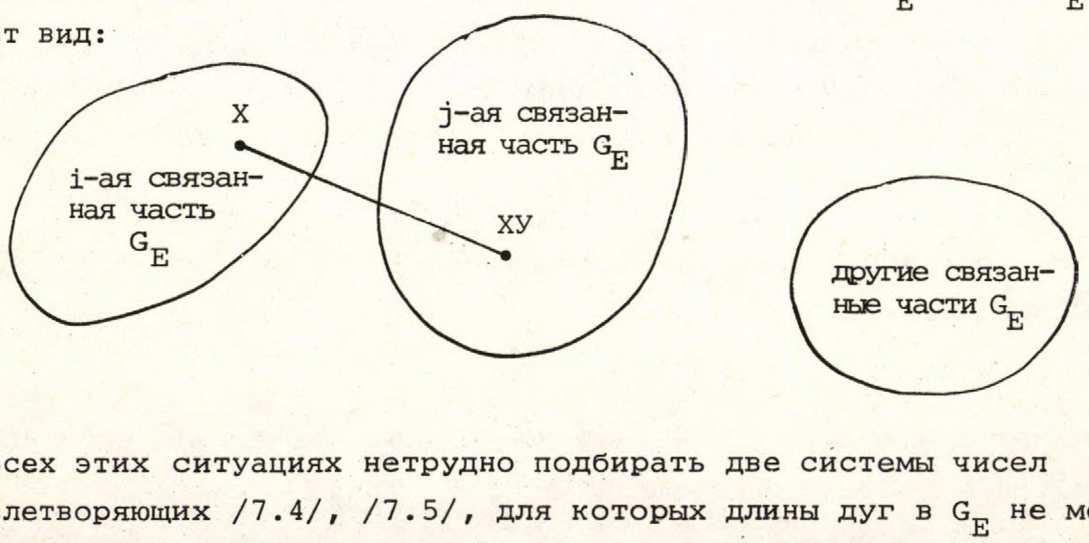
$$[X, У] \in Cal(E) \Longrightarrow [X, У] = \sum_{i=1}^{k} \alpha_i [A_i, B_i] \qquad /7.14/$$

Допустим, что $[X, У] \cup E$ есть линейно-независимая система. Построим граф $G_E^{\times}$ такой: $G_E^{\times}$ есть G_E с добавлением дуги соединяющей вершины X и XУ. Случаются два случая:

Первый случай: по крайней мере одна из двух вершин X, XУ не принадлежит G_E. В этом случае граф $G_E^{\times}$ имеет один из трех следующих видов:



Второй случай: обе вершины X, Ху принадлежат G_E. В этом случае дуга соединяющая X и XУ должна не принадлежать G_E. Это значит она соединяет две не связанные части графа G_E. Граф $G_E^{\times}$ имеет вид:



Во всех этих ситуациях нетрудно подбирать две системы чисел удовлетворяющих /7.4/, /7.5/, для которых длины дуг в G_E не меняются, а $[X, У] = [\Lambda, XУ] - [\Lambda, X]$ /т.е. длина дуги X, XУ/ меняется. Это отрицает существование какой-либо функции f , чтобы

$$[X, У] = f([A_i, B_i] \quad i = \overline{1,k})$$

Что показывает $[X, У] \in cal(E)$.

ТЕОРЕМА 10.

$$Cal(E) = Dedex(E) \qquad /7.15/$$

Доказательство:

$Dedex(E) \subset Cal(E)$ очевидно. Пусть $[X, У] \in Cal(E)$. Тогда $[X, У] \in Est(E)$; $\Longrightarrow [X, У] \in Ded(E)$.

Можем допустить, что E независима, представление $[X, У]$ в виде линейной комбинации ОС в E единственно. Следовательно

$$[X, У] \in Dedex(E).$$

Мы выяснили структуру множеств Cal(E), Est(E), Ded(E) и Dedex(E). Все сказанное дает нам формулу:

$$Cal(E) = Dedex(E) \subset Ded(E) = Est(E) \qquad /7.16/$$

Полученные результаты решили и задачу непротиворечивости в самой заданной системе E, чего вообще нет в логических видах зависимостей данных в базах данных.

ЗАКЛЮЧЕНИЕ

В этой статье мы написали об основных результатах начальных изучений случайных зависимостей данных в базах данных. Видно, что даже с мерой исходящей из МВ еще много нужно делать. Интересно подходить к этой проблеме с экспертными мерами взаимовлияний данных. Все это позволяет нам проникнуть в семантику данных.

Л И Т Е Р А Т У Р А

/1/ Armstrong W.W.: Dependency structures of date base relationships. Information Processing, North Holland Publ. Co. 1974.

/2/ J. Demetrovics and Gy. Gyepesi: Some generalized type functional dependencies formalized as equality set on matrices. Discrete Applied Mathematics 6 /1983/, North Holland Publ. Co.

/3/ Ullman D.: Principles of Database Systems, Computer Science Press - 1980.

/4/ Shannon C.A.: Mathematical Theory of Communication /перевод на русском языке/.

/5/ Trong N. and Chau H.M.: О линейном представлении энтропии по заданной системе энтропий /статья прислана в журнал Вычислительная техники и кибернетика на вьетнамском языке - 1986/.

ON RANDOM DEPENDENCIES OF DATA IN RELATIONAL DATA BASES

Trong N.

Summary

Given a relational data base R and two sets of attributes $A,B \subset \Omega$ of R, a "probability" matrix P(B/A) is defined using some "statistics" among data. This matrix is called the matrix of "influence" of A upon B. Using P(B/A) the entropy [A,B] of B relative to A is defined and [A,B] is called the "freedom" of B relative to A. The number [A,B] can be used to describe the "influence" of A to B (say, B functionally depends on A iff [A,B] = 0). In the paper the following problems are studied: Given a system $E = \{[A_i,B_i] | A_i,B_i \subset \Omega, \ i = 1,\ldots,k\}$ what can be said about the pairs [X,Y] which can be calculated, estimated, derived and exactly derived, respectively, from E. ("Calculated", "estimated", e.t.c. are precisely defined in the paper.)

A RELÁCIÓS ADATBÁZIS ADATAINAK VÉLETLEN FÜGGŐSÉGEIRŐL

Trong N.

Összefoglaló

Ha adva van egy relációs adatbázis(R)és R-nek két attributum halmaza, $A,B \subset \Omega$, akkor az adatok bizonyos "statisztikája" alapján egy P(B/A) "valószinüségi" mátrixot lehet definiálni. Ezt a mátrixot a szerző az A-nak a B-re történő "hatás-mátrixának" ("matrix of influence") nevezi. A P(B/A) felhasználásával ki lehet számitani a B-nek az A-ra vonatkoztatott [A,B] entropiáját, amit a szerző a B-nek az A-tól való "szabadsága" mértékének nevez. (pl. $B \xrightarrow{F} A \iff [A,B] = 0$).

A cikkbena szerző a következő négy problémát tárgyalja:

Ha adott egy $E = \{[A_i,B_i] | A_i,B_i, \subset \Omega, \ i = 1,...,k\}$ rendszer, mit lehet mondani azokról az [X,Y] párokról, amelyeket ki lehet számitani, meg lehet becsülni, le lehet vezetni ill. pontosan le lehet vezetni az E alapján (ezeket a fogalmakat a szerző pontosan definiálja).

MTA SZTAKI Közlemények 36/1987 pp. 173-196

A REDUCTION THEOREM FOR THE MEASURES OF SUM-SETS IN R^n

B. UHRIN

Computer and Automation Institute
Hungarian Academy of Sciences
Budapest, Hungary

1. INTRODUCTION

If we have two compact sets $A, B \subset R^1$ such that $A \cap B = \{0\}$ (the zero) and A is to the left of 0 and B is to the right of 0, then the algebraic (Minkowski) sum A+B contains $A \cup B$, hence

$$(1.1) \qquad \mu_{1*}(A+B) \geq \mu_1(A) + \mu_1(B),$$

where μ_1 and μ_{1*} are Lebesque-measure and inner L-measure in R^1, respectively (A+B is in general not L-measureable). This implies that (1.1) holds for any L-measureable sets $A, B \subset R^1$, because the measures are translation invariant and A,B can be approximated from inside by compact sets.

The inequality (1.1) is the 1-dimensional Brunn-Minkowski -Lusternik (B-M-L)-inequality. We see that the proof of (1.1) is quite simple. We know also that equality occurs in (1.1) if and only if A and B are homothetic intervals. While the "if" part of this statement is trivial, the "only if" part has been rigorously proved first in early fifties by Henstock and Macbeath [1]. They solved the problem via the following integral inequality: Given $\alpha > 0$ and L-integrable functions $f, g: R^1 \to R^1_+$ such that $0 < \gamma := \sup_x f(x) < +\infty$, $0 < \delta := \sup_x g(x) < +\infty$, we have

$$(1.2)\qquad \int_{R^1}^{*} h^{\alpha}(t)\,dt \geq (\gamma+\delta)^{\alpha}\left(\gamma^{-\alpha}\int_{R^1} f^{\alpha}(x)\,dx + \delta^{-\alpha}\int_{R^1} g^{\alpha}(x)\,dx\right),$$

where $\int^{*}$ is the inner L-integral, $h(t) := \sup\{h(x,y): x+y=t\}, t\in R^1$, and $h(x,y)$ equals to $f(x)+g(y)$ when $f(x)g(y) > 0$ and zero otherwise [1].

Henstock and Macbeath used for the proof of (1.2) the following nice idea which goes back to the Bonnesen's proof of (1.1), [2] (see also [3]).

Denote

$$(1.3)\qquad A(\xi) := \{x\in R^1: f^{\alpha}(x) \geq \gamma^{\alpha}\xi\},\; B(\xi) := \{x\in R^1: g^{\alpha}(x) \geq \delta^{\alpha}\xi\},\quad 0 \leq \xi \leq 1,$$

$$(1.4)\qquad C(\xi) := \{t\in R^1: h^{\alpha}(t) \geq (\gamma+\delta)^{\alpha}\xi\},\qquad 0 \leq \xi \leq 1.$$

Then

$$(1.5)\qquad C(\xi) \supseteq A(\xi) + B(\xi),\qquad 0 \leq \xi \leq 1,$$

hence, using (1.1) we get

$$(1.6)\qquad \mu_{1*}(C(\xi)) \geq \mu_1(A(\xi)) + \mu_1(B(\xi)),\qquad 0 \leq \xi \leq 1$$

and integrating (1.6) over $0 \leq \xi \leq 1$, we get (1.2).

In the last step we used the obvious identity

$$(1.7)\qquad \int_{R^1} \varphi(x)\,dx = \int_0^{+\infty} \mu_1(\{x: \varphi(x) \geq \xi\})\,d\xi.$$

It can be seen easily that the inequality (1.2) holds for any $-\infty \leq \alpha \leq +\infty$, $\alpha \neq 0$, as well as (1.2) can be given a more "integral-theoretic" form taking "ess-sup" instead of "sup" in the definitions of $h(t)$, γ and δ. Namely, using the steps (1.3) $\div$ (1.7) we can easily prove

$$(1.8) \qquad \int_{R^1} k(t)dt \geq \frac{\lambda}{\gamma} \int_{R^1} f(x)dx + \frac{1-\lambda}{\delta} \int_{R^1} g(x)dx,$$

where $0 < \gamma := \operatorname{ess-sup}_x f(x) < +\infty$, $0 < \delta := \operatorname{ess-sup}_x g(x) < +\infty$,
and

$$(1.9) \qquad k(t) := \operatorname{ess-sup}_x \min\{\gamma^{-1} f(x/\lambda), \delta^{-1} \cdot g((t-x)/(1-\lambda))\},$$

$$t \in R^1.$$

Define for $a,b \geq 0$, $0 \leq \lambda \leq 1$ and $-\infty < \alpha < +\infty$, $\alpha \neq 0$ the "extended" means as follows

$$(1.10) \qquad M_\alpha^{(\lambda)}(a,b) := \begin{cases} 0 & \text{if } a \cdot b = 0 \\ (\lambda a^\alpha + (1-\lambda) b^\alpha)^{1/\alpha} & \text{if } a \cdot b > 0, \end{cases}$$

$$(1.11) \qquad M_0^{(\lambda)}(a,b) := \lim_{\alpha \to 0} M_\alpha^{(\lambda)}(a,b) = a^\lambda b^{(1-\lambda)}$$

$$(1.12) \qquad M_{-\infty}^{(\lambda)}(a,b) := \lim_{\alpha \to -\infty} M_\alpha^{(\lambda)}(a,b) = \min\{a,b\},$$

$$(1.13) \qquad M(a,b) := M_{+\infty}^{(\lambda)}(a,b) := \lim_{\alpha \to +\infty} M_\alpha^{(\lambda)}(a,b) = \begin{cases} 0 & \text{if } a\, b = 0 \\ \max\{a,b\} & \text{if } a \cdot b > 0. \end{cases}$$

It is clear that for any $a,b,c,d \geqslant 0$ and $-\infty \leqslant \alpha \leqslant +\infty$ we have

(1.14) $$M_{\alpha}^{(\lambda)}(a,b) \cdot M_{-\alpha}^{(\lambda)}(c,d) \geqslant \min\{ac,bd\}.$$

Hence we get from (1.8) immediately

(1.15) $$\int_{R^1} h_{\alpha}^{(\lambda)}(t)\,dt \geqslant M_{\alpha}^{(\lambda)}(\gamma,\delta)\left(\frac{\lambda}{\gamma}\int_{R^1} f(x)\,dx + \frac{1-\lambda}{\delta}\int_{R^1} g(x)\,dx\right),$$

where

(1.16) $$h_{\alpha}^{(\lambda)}(t) := \operatorname*{ess-sup}_{x \in R^1} M_{\alpha}^{(\lambda)}\left(f(x/\lambda),\ g\left(\frac{t-x}{1-\lambda}\right)\right), \quad t \in R^1.$$

(The function $h_{\alpha}^{(\lambda)}(t)$ is already measurable [4], while $h(t)$ is in general not.)
The "ess-sup" definition of $h_{\alpha}^{(\lambda)}(t)$ has an interesting auxiliary effect: taking characteristic functions of two sets χ_A and χ_B instead of f and g, the function $h_{\alpha}^{(\lambda)}(t)$ does not depend on α and it is the characteristic function of the set

(1.17) $$\lambda A \boxplus (1-\lambda)B := \{x \in R^1 : \mu_1(\lambda A \cap (x-(1-\lambda)B)) > 0\}.$$

The set (1.17) is empty if one of the sets has the measure zero. This set has been later called in [5] the "essential sum" of the sets λA and $(1-\lambda)B$. This sum is already measureable ([4]) and

(1.18) $\lambda A \boxplus (1-\lambda)B \subseteq \lambda A+(1-\lambda)B := \{ x\in R^1 : \lambda A \cap (x-(1-\lambda)B) \neq \emptyset \}$.

One can see easily that for compact sets A,B

(1.19) $\lambda A \boxplus (1-\lambda)B \supseteq \lambda A^* + (1-\lambda)B^*$,

where A^*, B^* are the sets of density points of the sets A,B. We recall that $x\in A^*$ iff (see, e.g. [1])

$$\lim_{\delta\to 0+} \frac{\mu_1(A \cap [x-\delta, x+\delta])}{\delta} = 2.$$

After that using the inequality (1.1) and the facts $\mu_1(A) = \mu_1(A^*)$, $\mu_1(B) = \mu_1(B^*)$, we see that

(1.20) $\mu_1(\lambda A \boxplus (1-\lambda)B) \geq \lambda\mu_1(A) + (1-\lambda)\mu_1(B)$

holds for any measureable $A,B \subset R^1$.
The later inequality is a slight sharpening of the B-M-L inequality (1.1) (see [6] for details).

The first multidimensional extension of (1.2) is due to Dancs and Uhrin [7] (see the case $k = n-1$ of Theorem 3.1 below and remarks in Section 4). The main problem in extending (1.2) (or (1.15)) to many dimensions is the presence of γ and δ. Applying to the right hand side of (1.2) the Hölder inequality, we get a weakening of (1.2) where γ and δ are already not involved. After that taking an induction on dimension, one can easily prove:

If $\alpha > 0$ then, denoting $\omega := \frac{\alpha}{1+n\alpha}$ we have

$$(1.21)\qquad \int^{*}_{R^n} h^{\alpha}(t)\,dt \geqslant \Big(\big(\int_{R^n} f^{\alpha}(x)\,dx\big)^{\omega} + \big(\int_{R^n} g^{\alpha}(x)\,dx\big)^{\omega}\Big)^{1/\omega}.$$

This inequality is due to Dinghas [8].

As to a weakening of (1.15), one first prove that for $a,b,c,d \geqslant 0$ and $\alpha+\beta \geqslant 0$ (see, [9],[10] for details):

$$(1.22)\qquad M^{(\lambda)}_{\alpha}(a,b)\cdot M^{(\lambda)}_{\beta}(c,d) \geqslant M^{(\lambda)}_{\frac{\alpha\beta}{\alpha+\beta}}(ac,bd).$$

Now applying this inequality (with $\alpha \geqslant -1$, $\beta = 1$) to the right hand side of (1.15) and performing an induction on the dimension n, we get for $\alpha \geqslant -1/n$ the inequality

$$(1.23)\qquad \int_{R^n} h^{(\lambda)}_{\alpha}(t)\,dt \geqslant M^{(\lambda)}_{\frac{\alpha}{1+n\alpha}}\Big(\int_{R^n} f(x)\,dx, \int_{R^n} g(x)\,dx\Big).$$

(1.23) or a weaker form of it (when "sup" is used instead of "ess-sup" in the definition of $h^{(\lambda)}_{\alpha}(t)$) has been proved and studied by many authors ([5],[7],[11],[12]). Taking in (1.23) $\alpha = +\infty$ and $f := \chi_A$, $g := \chi_B$ we get a following strengthened form of B-M-L inequality ([5])

$$(1.24)\qquad \mu_n(\lambda A \boxplus (1-\lambda)B) \geqslant \big(\lambda\mu_n(A)^{1/n} + (1-\lambda)\mu_n(B)^{1/n}\big)^n,$$

where μ_n is the L-measure in R^n and the essential sum is defined analogously to (1.17). In what follows we shall refer to the weaker form of (1.23) ("sup " instead of "ess-sup" and $\int^{*}_{R^n}$ instead of $\int_{R^n}$) as (1.23').

In this paper we prove some multidimensional extensions of (1.15) that sharpen and extend the result in [7]. Similar but weaker results have been proved also in [10]. However, our main aim is to prove an extension of the reduction theorem in [6] proved for the Lebesque measure μ_n. The measures involved will be generated by so called essentially α-concave functions. We note that many density functions of mathematical statistics belong to this class (see Section 4 for details).

2. PRELIMINARY REMARKS

Before turning to multidimensional extensions of (1.15) we shall write an elementary lower estimation for $\nu(\lambda A \boxplus (1-\lambda)B)$, where ν is a measure defined on L-measureable sets $\mathcal{L}_n \subset R^n$. In general, one can expect only that

$$(2.1) \qquad \nu(\lambda A \boxplus (1-\lambda)B) \geq \nu(\lambda A)+\nu((1-\lambda)B).$$

(The inequalities of type $m(A+B) \geq m(A)+m(B)$ in the more general setting in locally compact Abelian groups are studied in [13]).
We shall see that for well defined classes of measures estimations much better than (2.1) can be proved.
In what follows, we shall frequently use the following identity (an extension of (1.7) to higher dimensions):

$$(2.2) \qquad \int_{R^n} \varphi(x)\,dx = \int_0^{+\infty} \mu_n(\{x \in R^n : \varphi(x) \geq \xi\})\,d\xi,$$

where φ is any non-negative L-integrable function (here and everywhere below $\int \cdot\, dx$ means L-integral).

If $f: R^n \to R^1_+$ is an L-integrable function, then it generates a measure

$$(2.3) \qquad \nu(A) := \int_A f(t)\,dt, \quad A \in \mathcal{L}_n.$$

The function $f \equiv 1$ is the generator of the L-measure μ_n (the integrable here means that $\int_{R^n} f(x)\,dx$ is also meaningful but it can have the value $+\infty$). Let $A \in \mathcal{L}_n$ be an essentially bounded set (i.e. $\mathrm{ess}_x\text{-}\sup \chi_A(x) < +\infty$) such that

$$(2.4) \qquad 0 < m_f(A) := \operatorname*{ess\text{-}sup}_{x \in R^n} \chi_A(x) f(x) < +\infty.$$

Then denoting

$$(2.5) \qquad A_f(\xi) := \{x \in A : f(x) \geq m_f(A)\,\xi\},$$

we have

$$(2.6) \qquad \nu(A) = m_f(A) \int_0^1 \mu_n(A_f(\xi))\,d\xi.$$

Now, assume that for some $0 \leq \lambda \leq 1$ and $-\infty \leq \alpha \leq +\infty$ the function f is such that

$$(2.7) \qquad f(t) \geq \operatorname*{ess-sup}_{x \in R^n} M_\alpha^{(\lambda)}(f(x/\lambda), f((t-x)/(1-\lambda)))$$

for a.e. $t \in R^n$.

Denote the function on the right side of (2.7) by $p_{\alpha}^{(\lambda)}(t)$, $t \in R^n$ (this is defined for all $t \in R^n$).
Let $A, B \in \mathcal{L}_n$ be two essentially bounded sets and let $t \in R^n$ be such that

$$(2.8) \qquad \mu_n(\lambda A_f(\xi) \cap (t-(1-\lambda)B_f(\xi))) > 0.$$

For any $x \in R^n$ belonging to the intersection in (2.8) we have

$$(2.9) \qquad \min\left\{ \frac{f(x/\lambda)}{m_f(A)}, \frac{f((t-x)/(1-\lambda))}{m_f(B)} \right\} \geqslant \xi ,$$

i.e. (2.9) holds for all x from a subset of positive μ_n-measure.

This implies, denoting

$$(2.10) \qquad p(t) := \operatorname*{ess-sup}_{x \in R^n} \min\left\{ \frac{f(x/\lambda)}{m_f(A)}, \frac{f((t-x)/(1-\lambda))}{m_f(B)} \right\} ,$$

that

$$(2.11) \qquad (\lambda A \boxplus (1-\lambda)B)_p(\xi) \supseteq \lambda A_f(\xi) \boxplus (1-\lambda)B_f(\xi),$$

hence using the trivial inequality (1.14) we see that

$$(2.12) \qquad \nu(\lambda A \boxplus (1-\lambda)B) \geqslant M_{\alpha}^{(\lambda)}(m_f(A), m_f(B)) \cdot \int_0^1 \mu_n(\lambda A_f(\xi) \boxplus (1-\lambda)B_f(\xi))\, d\xi .$$

If (2.7) is satisfied for all $0 \leq \lambda \leq 1$, then we call the function f essentially α-concave.
Using (1.24) and the inequality

$$(2.13) \qquad M_{\alpha}^{(\lambda)}(a,b) \cdot M_{\beta}^{(\lambda)}(c,d) \geqslant \min\{\lambda^{\frac{\alpha+\beta}{\alpha\beta}} ac, (1-\lambda)^{\frac{\alpha+\beta}{\alpha\beta}} bd\},$$

which holds for $a,b,c,d \geqslant 0$ and $\alpha+\beta \leq 0$, $\alpha \cdot \beta < 0$ (see [10]), we get immediately from (2.12)

$$(2.14) \qquad \nu(\lambda A \boxplus (1-\lambda)B) \geqslant \min\{\lambda^{n+(1/\alpha)}\nu(A), (1-\lambda)^{n+(1/\alpha)}\nu(B)\},$$

where $-\infty \leq \alpha \leq -1$.

This inequality has been proved for $-\infty \leq \alpha \leq -1/n$ in [7] (see Section 4 for details). We see that already the trivial reduction inequality (2.12) is sharper than a known result.

3. AN INTEGRAL INEQUALITY AND REDUCTION THEOREM

Here we use the definitions and notations of the previous sections. First, some new notations.
Let $S \subset R^n$ be a k-dimensional linear subspace, $0 \leq k \leq n$, and $T \subset R^n$ be an (n-k)-dimensional linear subspace such that $S \oplus T = R^n$ (the direct sum). The L-measures in S and T will be denoted by μ_k and μ_{n-k}, respectively.
We shall denote the L-integrals both in S and T by $\int \cdot \, dx$ (the meaning will be clear from the context). By definition $\mu_0(\theta) = 1$, $\mu_0(\emptyset) = 0$.

Given an L-integrable non-negative function $f:R^n \to R^1_+$, we shall denote

(3.1) $$i(f,u) := \int_S f(x+u)\,dx, \quad u \in T,$$

(3.2) $$m_k(f) := \operatorname*{ess-sup}_{u \in T} i(f,u),$$

in particular

(3.3) $$m_o(f) := \operatorname*{ess-sup}_{x \in R} f(x), \; m_n(f) := \int_{R^n} f(x)\,dx,$$

and

(3.4) $$H_f(\xi) := \{ x \in R^n : f(x) \geq m_o(f)\xi \}, \quad 0 \leq \xi \leq 1.$$

Given two L-integrable functions $f,g:R^n \to R^1_+$, denote for $-\infty \leq \alpha \leq +\infty$ and $0 \leq \lambda \leq 1$

(3.5) $$h_\alpha^{(\lambda)}(t) := \operatorname*{ess-sup}_{x \in R^n} M_\alpha^{(\lambda)}(f(x/\lambda), g((t-x)/(1-\lambda))), \; t \in R^n,$$

(3.6) $$k_\alpha^{(\lambda)}(\tau) := \operatorname*{ess-sup}_{u \in T} M_\alpha^{(\lambda)}\left(\frac{i(f,\frac{u}{\lambda})}{m_k(f)}, \frac{i(g,\frac{\tau-u}{1-\lambda})}{m_k(g)}\right), \; \tau \in T,$$

(in particular if $k = n$ then $k_\alpha^{(\lambda)}(\theta) = 1$).

Now, we have

<u>Theorem 3.1.</u> The following two inequalities hold

$$(3.7)\qquad \int_{R^n} \operatorname*{ess-sup}_{x\in R^n} \min\left\{\frac{f(x/\lambda)}{m_o(f)}, \frac{g((t-x)/(1-\lambda))}{m_o(g)}\right\} dt \geq$$

$$\geq \int_0^1 \mu_n(\lambda H_f(\xi) \boxplus (1-\lambda) H_g(\xi))\, d\xi ;$$

if $0 < k \leq n$, $\alpha+\beta \geq 0$, $(\alpha\beta/(\alpha+\beta)) \geq -1/k$, then

$$(3.8)\qquad \int_{R^n} h_\alpha^{(\lambda)}(t)\,dt \geq M_{-\beta}^{(\lambda)}(m_k(f), m_k(g)) . \int_T k_\omega^{(\lambda)}(\tau)\, d\tau ,$$

where $\omega := ((1/\alpha) + (1/\beta) + k)^{-1}$. □

Proof. The proof of (3.7) is quite simple.
Denoting by $q(t)$ the integrand in the left hand side of (3.7), we can easily see (analogously to (2.11)) that

$$(3.9)\qquad H_q(\xi) \geq \lambda H_f(\xi) \boxplus (1-\lambda) H_g(\xi), \qquad 0 \leq \xi \leq 1,$$

which gives (3.7).

The case $k = n = 1$ of the inequality (3.8) is a simple consequence of (3.7): apply first (1.20) to the integrand in the right hand side of (3.7), integrate over $0 \leq \xi \leq 1$, take into account (1.7) and finally use (1.14) to get (1.15) (in fact, we have proved (1.15) along these lines also in Section 1). Now, applying (1.22), $\alpha \geq -1$, $\beta = 1$, we get from (1.15) the case $k = n = 1$ of (3.8).

Assume for the moment that (3.8) is already proved for $k = n-1$ $(n > 1)$. Then the case $k = n$ can be derived in the following way.
Let α and β be such that

$$\alpha+\beta \geq 0, \quad \alpha\beta/(\alpha+\beta) \geq \frac{-1}{n-1}. \tag{3.10}$$

Then, we know that

$$\int_{R^n} h_{\alpha}^{(\lambda)}(t)\,dt \geq M_{-\beta}^{(\lambda)}(m_{n-1}(f), m_{n-1}(g)) \cdot \int_T k_{\omega}^{(\lambda)}(\tau)\,d\tau, \tag{3.11}$$

where $\omega = (1/\alpha+1/\beta+n-1)^{-1}$.

T is now 1-dimensional, hence applying (3.7) for $n = 1$ and after that using (1.20) and (1.7) we get

$$\int_{R^n} h_{\alpha}^{(\lambda)}(t)\,dt \geq M_{-\beta}^{(\lambda)}(m_{n-1}(f), m_{n-1}(g)) \cdot \left(\lambda\frac{m_n(f)}{m_{n-1}(f)} + (1-\lambda)\frac{m_n(g)}{m_{n-1}(g)}\right). \tag{3.12}$$

The conditions (3.10) are equivalent to the conditions

$$\alpha \geq -1/(n-1), \quad \beta \geq \frac{-\alpha}{1+(n-1)\alpha}. \tag{3.13}$$

Let α, β be such that $\alpha+\beta \geq 0$ and $\alpha\beta/(\alpha+\beta) \geq -\frac{1}{n}$, or equivalently

$$\alpha \geq -\frac{1}{n}, \quad \beta \geq \frac{-\alpha}{1+n\alpha}. \tag{3.14}$$

It is clear that for such α and β the conditions (3.13) are also fulfilled, and we can write (3.12) for $-\beta = \frac{\alpha}{1+(n-1)\alpha}$. For this β we have $-\beta \geq -1$ (because $\alpha \geq -\frac{1}{n}$), hence using (1.22) to the right hand side of (3.12) we get

$$(3.15) \qquad \int_{R^n} h_{\alpha}^{(\lambda)}(t)\,dt \geq M^{(\lambda)}_{\frac{\alpha}{1+n\alpha}}(m_n(f),m_n(g)) \geq M^{(\lambda)}_{-\beta}(m_n(f),m_n(g))$$

for any α and β fulfilling (3.14). This proves (3.8) for the pair $(n,k = n)$ (assuming that it holds for $(n, k = n-1)$. Now, take the pairs (n,k), $1 \leq k \leq n$, into lexicographic order, i.e. $(n_1,k_1) \prec (n_2,k_2)$ iff either $n_1 < n_2$ or $\{ n_1 = n_2$ and $k_1 < k_2 \}$.

We get a sequence

$$(3.16) \qquad (1,1) \prec (2,1) \prec (2,2) \prec (3,1) \prec (3,2) \prec (3,3) \prec \ldots$$

We proceed with the proof of (3.8) by induction on this sequence. For $n = k = 1$ (3.8) is true. Assume we have proved (3.8) for all first $N-1$ members of (3.16). Let (n,k) be the N-th member of the sequence. If $k = n$ we are ready by the above reasoning because the case $(n,k = n-1)$ is assumed to be proved by the induction. So assume $1 \leq k \leq n-1$, $n > 1$ and

$$(3.17) \qquad \alpha \geq -\frac{1}{k}, \quad \beta \geq \frac{-\alpha}{1+k\alpha}$$

(these conditions are equivalent to $\beta+\alpha \geq 0$, $\alpha\beta/(\alpha+\beta) \geq -\frac{1}{k}$).

We can write using (1.22)

$$(3.18)\qquad M_{\beta}^{(\lambda)}(1/m_k(f),\ 1/m_k(g))\cdot\int_{R^n} h_{\alpha}^{(\lambda)}(t)\,dt \geq$$

$$\geq \int_T \underset{eT}{\text{ess-sup}}\cdot\Big(\int_S \underset{x\in S}{\text{ess-sup}}\ M^{(\lambda)}_{\frac{\alpha\beta}{\alpha+\beta}}\Big(\frac{1}{m_k(f)}f\Big(\frac{x}{\lambda}+\frac{u}{\lambda}\Big),\frac{1}{m_k(g)}\,g\Big(\frac{\tau-u}{1-\lambda}+\frac{z-x}{1-\lambda}\Big)\,dz\Big)d\tau\ .$$

Applying (3.8) case (k,k) to the inner integral $\int_S\ldots dz$, where now $\alpha\beta/(\alpha+\beta)$ plays the role of α, we get that the right hand side of (3.18) is not less then

$$(3.19)\qquad \int_T \underset{u\in T}{\text{ess-sup}}\ M_{\omega}^{(\lambda)}\Big(\frac{i(f,\frac{u}{\lambda})}{m_k(f)},\ \frac{i(g,\frac{\tau-u}{1-\lambda})}{m_k(g)}\Big)\ d\tau.$$

By this (3.8) and the whole theorem is proved. ■

Let $f:R^n \to R^1_+$ be L-integrable function and $A,B \subset R^n$ be two essentially bounded L-measurable sets.

Denote

$$(3.20)\qquad m_k(A) := m_k(\chi_A\cdot f),\quad m_k(B) := m_k(\chi_B\cdot f)$$

and for $0 \leq \xi \leq 1$

$$(3.21)\qquad \begin{aligned} A(\xi) &:= \{\, u\in T:\ i(\chi_A\cdot f,u) \geq m_k(A)\,\xi \,\} \\ B(\xi) &:= \{\, u\in T:\ i(\chi_B\cdot f,u) \geq m_k(B)\,\xi \,\}. \end{aligned}$$

Now we have

Theorem 3.2. If $-\frac{1}{k} \leq \alpha \leq +\infty$, $0 \leq \lambda \leq 1$, $0 \leq k \leq n$ and A,B and f are such that

(3.22) $$0 < m_k(A),\ m_k(B) < +\infty$$

and

(3.23) $$f(t) \geq \operatorname*{ess\text{-}sup}_{x \in R^n} M_{\alpha}^{(\lambda)}(f(x/\lambda), f(\tfrac{t-x}{1-\lambda})) \quad \text{for a.e. } t \in R^n,$$

then for the measure ν generated by the f we have

(3.24) $$\nu(\lambda A \boxplus (1-\lambda)B) \geq M_{\beta}^{(\lambda)}(m_k(A), m_k(B)),$$

$$\int_0^1 \mu_{n-k}(\lambda A(\xi) \boxplus (1-\lambda) B(\xi))\, d\xi,$$

where

$$\beta = \frac{\alpha}{1+k\alpha}. \quad \square$$

Proof. After some technical observations, (3.24) will follow from the previous theorem.
Denote the right hand side of (3.23) by $r_{\alpha}^{(\lambda)}(t)$ and

(3.25) $$s_{\alpha}^{(\lambda)}(t) := \operatorname*{ess\,sup}_{x \in R^n} M_{\alpha}^{(\lambda)}(\chi_A(\tfrac{x}{\lambda}) \cdot f(\tfrac{x}{\lambda}), \chi_B(\tfrac{t-x}{1-\lambda}) \cdot f(\tfrac{t-x}{1-\lambda})).$$

First we prove that

$$(3.26)\qquad \int_{\lambda A \boxplus (1-\lambda)B} r_{\alpha}^{(\lambda)}(t)\,dt = \int_{R^n} s_{\alpha}^{(\lambda)}(t)\,dt.$$

It is clear that

$$(3.27)\qquad \chi_{\lambda A \boxplus (1-\lambda)B}(t) = \operatorname*{ess-sup}_{x \in R^n} M\left(\chi_A\left(\tfrac{x}{\lambda}\right), \chi_B\left(\tfrac{t-x}{1-\lambda}\right)\right)$$

where M is defined by (1.13).

Now, using the trivial inequalities

$$(3.28)\qquad \operatorname*{ess-sup}_{x} \varphi(x) \cdot \operatorname*{ess-sup}_{x} \psi(x) \geqslant \operatorname*{ess-sup}_{x} (\varphi(x) \cdot \psi(x)),$$

$$(3.29)\qquad M(a,b) M_{\alpha}^{(\lambda)}(c,d) \geqslant M_{\alpha}^{(\lambda)}(ac,bd),$$

we see that the left hand side of (3.26) is not less than the right hand side.
On the other hand, if for given t and x

$$(3.30)\qquad M_{\alpha}^{(\lambda)}\left(\chi_A\left(\tfrac{x}{\lambda}\right) \cdot f\left(\tfrac{x}{\lambda}\right), \ \chi_B\left(\tfrac{t-x}{1-\lambda}\right) \cdot f\left(\tfrac{t-x}{1-\lambda}\right)\right) > 0,$$

then clearly $\chi_A\left(\frac{x}{\lambda}\right) = \chi_B\left(\frac{t-x}{1-\lambda}\right) = 1$ ($M_{\alpha}^{(\lambda)}$ is the "extended" mean), i.e.

(3.31) $$x \in \lambda A \cap (t-(1-\lambda)B).$$

The definition of ess-sup shows that if $s_{\alpha}^{(\lambda)}(t) > 0$ then there is a set E such that $\mu_n(E) > 0$ and for all $x \in E$ (3.30) holds, i.e. $t \in \lambda A \boxplus (1-\lambda)B$. This shows that the right hand side of (3.26) is not less than the left hand side. Apply now Theorem 3.1 to functions $\chi_A \cdot f$ and $\chi_B \cdot f$. Similarly to (2.11), we can easily check that

(3.32) $$C(\xi) \supseteq \lambda A(\xi) \boxplus (1-\lambda)B(\xi), \quad 0 \leq \xi \leq 1,$$

where

(3.33) $$C(\xi) := \{ \tau \in T : k_{\omega}^{(\lambda)}(\tau) \geq \xi \}.$$

Hence

(3.34) $$\int_T k_{\omega}^{(\lambda)}(\tau)d\tau \geq \int_0^1 \mu_{n-k}(A(\xi) \boxplus (1-\lambda)B(\xi))d\xi.$$

By this (3.24) is proved (we apply (3.8) in the sharpest case $\beta = \frac{-\alpha}{1+k\alpha}$). ■

4. CONCLUDING REMARKS

1. The case $k = n-1$ of (3.8) has been principally proved in [7] (more exactly, the "sup" was used instead of "ess-sup"). The "ess-sup"-case of these inequalities needs some additional care.

For any $0 \leq k \leq n$, a weaker form of (3.8) has been first formulated and proved in [10].

For the domain $-\infty \leq \alpha \leq -\frac{1}{n}$ a following inequality has been also proved in [7]:

$$(4.1) \quad \int_{R^n} \sup_{\lambda x+(1-\lambda)y=t} M_{\alpha}^{(\lambda)}(f(x),q(y))dt \geqslant$$

$$\geqslant \min\left\{ \lambda^{n+(1/\alpha)} \int_{R^n} f(x)dx, (1-\lambda)^{n+(1/\alpha)} \int_{R^n} g(x)dx \right\}$$

(under the assumption the f and g are such that the function $\sup M_{\alpha}^{(\lambda)}$ ... is integrable).

Using the inequality (2.13), we can see easily that in each of the domains $-1/k \leqslant \alpha \leqslant -\frac{1}{k-1}$, $k = 0,1,2,...,n-1$, the inequality (3.8) gives results that are "from both sides" sharper than (4.1).

The inequality (1.23') has been successfully applied in many branches of mathematics: stochastic programming (the case $\alpha = 0$, [14],[15]); mathematical statistics ($\alpha \geqslant -\frac{1}{n}$, [16]); theory of probability ($\alpha \geqslant -\frac{1}{n}$, [12]); theory of diffusion equations ($\alpha = 0$; [5]).

Some principally new results concerning the convolution of unimodal functions has been proved using both (1.23') and (4.1) ([9]).

The conditions of equality in (1.23') has been investigated in [11] ($\alpha \geqslant -\frac{1}{n}$) and in [15] ($\alpha = 0$).

In fact, using another method of the paper [1], one can prove sufficient and necessary conditions of equality in (1.23') for $n = 1$, $\alpha \geqslant -1$ (see [17], p. 131). The proof of such conditions for the sharper inequality (1.15) seems to be a more difficult problem and this has been done only for upper semi-continuous functions f and q (see [17]).

2. As to lower estimations for $\nu(\lambda A \boxplus (1-\lambda)B)$ (or for $\nu_*(\lambda A+(1-\lambda)B)$, where ν_* is the inner ν-measure) only the case $k = n$ of (3.24) has been studied ([5],[7],[11],[12],[15]). For $\alpha \leq -\frac{1}{n}$ the inequality (4.1) was used in [7] to prove a lower estimation for $\nu_*(\lambda A+(1-\lambda)B)$. Using (2.13), our inequality (3.24) can be used to write in each of the domains $-\frac{1}{k} \leq \alpha \leq -\frac{1}{k-1}$, $k = 0,1,\ldots,n-1$, inequalities which are "from both sides" sharper than those in [7]. The case $f \equiv 1$ (L-measure) and $k = n-1$ of (3.24) (more exactly taking $\mu_{n*}(\lambda A+(1-\lambda)B)$ instead of $\mu_n(\lambda A \boxplus (1-\lambda)B)$ is essentially due to Bonnesen (this is the classical sharpening of the B-M-L inequality, see [6] for details). The proper geometric content of inequalities is not quite clear yet, it is so even in the case $f \equiv 1$ (L-measure). For L-measure μ_n the quantities $m_k(A)$ are called in the geometry "inner transversal measures" ("innere Quermass", see e.g. [18]). An interesting theme of study would be to compare (3.24) (at least the case $f \equiv 1$) with some other results in geometry that use transversal measures (see [6] for more details).
The study of more general measures seems to be interesting as well. Let us recall that the function f satisfying (3.23) for all $0 \leq \lambda \leq 1$ we called essentially α-concave. If in (3.23) we take "sup" instead of "ess-sup" and $t \in R^n$ instead of a.e. $t \in R^n$ (let us call these functions α-concave), we get a more restricted class of functions.

Many important density functions in statistics are known to be α-concave. For example, the density functions of normal distribution, Wishart distribution, multidimensional β-distribution, Dirichlet-distribution are known to be 0-concave (log-concave, see [14],[15]), while those of the Pareto-distribution, Student-t-distribution, F-distribution are known to be α-concave for some $\alpha < 0$ ([11]).

3. The inequalities (3.7),(3.8) and (3.24) are to be considered as tools for getting new lower estimations for $\int h_{\alpha}^{(\lambda)}(t)dt$ and $\nu(\lambda A \boxplus (1-\lambda)B)$. Say, we can apply them successively for a series of "nested" subspaces S. One can imagine, that we would get a plenty of inequalities of a pretty complicated from (a simple example of this procedure can be found in [6]). Further research will show, whether these complicated (but very sharp) inequalities can be used in solving some interesting problems.

REFERENCES

[1] R. Henstock, A.M. Macbeath, On the measure of sum sets. (I) The theorems of Brunn, Minkowski and Lusternik, Proc. London Math. Soc., Ser. III., 3(1953), 182-194.

[2] T. Bonnesen, "Les problêmes des isoperimétres et des isèpiphanes", Dunod, Paris, 1929.

[3] T. Bonnesen, W. Fenchel, "Theorie der konvexen Körper", Springer, Berlin, 1934.

[4] B. Uhrin, On some inequalities of inverse Hölder type having applications to stochastic programming problems, Seminar Notes, Mathematics No 2, Hungarian Committee for Systems Analysis, Budapest, 1975.

[5] H.J. Brascamp, E.H. Lieb, On extensions of the Brunn-Minkowski and Prékopa-Leindler theorems including inequalities for log-concave functions, and with an application to the diffusion equation, J. Functional Anal., 22(1976), 366-389.

[6] B. Uhrin, Extensions and sharpenings of Brunn--Minkowski and Bonnesen inequalities, Proc. Coll. on Intuitive Geometry (B. széplak, Hungary, 1985), Coll. Math. Soc. J. Bolyai, Vol 48. Horth-Holland, Amsterdam-New-York, 1987 , 551- 571 .

[7] S. Dancs, B. Uhrin, On a class of integral inequalities and their measure-theoretic consequences, J. Math. Anal. Appl., 74(1980), 388-400.

[8] A. Dinghas, Über eine Klasse superaditiver Mengenfunktionale von Brunn-Minkowski-Luste ik-schem Typus, Math. Zeitschr., 66(1957), 111-12.

[9] B. Uhrin, Some remarks about the convolution of unimodal functions, Annals of Prob., 12(1984), 640-645.

[10] B. Uhrin, Sharpenings and extensions of Brunn--Minkowski-Lusternik inequality, Tech. Report No. 203, November 1984, Stanford Univ., Dept. of Statistics, Stanford, CA, U.S.A.

[11] C. Borell, Convex set functions in d-space, Period. Math. Hungar., 6(2), (1975), 111-136.

[12] S. Das Gupta, Brunn-Minkowski inequality and its aftermath, J. Multivar. Anal. 10 (1980), 295-318.

[13] M. Kneser, Summengen in lokalkompakten abelschen Gruppen, Math. Zeitschr., 66(1956), 88-110.

[14] A. Prékopa, Logarithmic concave measures with applications to stochastic programming problems, Acta Sci. Math. (Szeged), 32(1971), 301-316.

[15] A. Prékopa, On logarithmic concave measures and functions, Acta Sci. Math. (Szeged), 34(1973), 335-343.

[16] Y. Rinott, On convexity of measures, Annals of Prob., 4(1976), 1020-1026.

[17] I. Dancs, B. Uhrin, On the conditions of equality in an integral inequality, Publ. Math. (Debrecen), 29(1982), 117-132.

[18] H. Hadwiger, "Vorlesungen über Inhalt, Oberfläche und Isoperimetrie", Springer, Berlin, 1957.

Одна редукционная теорема для мер суммы двух множеств в R^n

Б. Ухрин

Резюме

Пусть μ_k есть Лебеговая мера в R^k, $0 \le k \le n$, и определим для двух измеримых множеств $A, B \subset R^k$ их естественную выпуклую комбинацию как $(\lambda A \oplus (1-\lambda) B) := \{z \in R^k : \mu_k(\lambda A \cap (z-(1-\lambda)B)) > 0\}$, $0 \le \lambda \le 1$. Пусть $S \subset R^n$ и $T \subset R^n$ такие линейные подпространства размерности k u$(n-k)$, что $S \oplus T = R^n$. Автор в предыдущей статье /Coll. Math. Soc. J. Bolyai, Vol 48, North-Holland, 1987, 551-571/ дал нижную оценку для $\mu_n(\lambda A \oplus (1-\lambda)B)$ используя естественные супремумы функций $\varphi(u) := \mu_k(A \cap (S+u))$, $\psi(u) := \mu_k(B \cap (S+u))$, $u \in T$, и μ_{n-k}-мер естественных выпуклых комбинаций верных множеств уровня этих функций. Статья распространяет этот результат на более общие меры ν_n, которые индуцированы неотрицательными функциями /определенными на R^n/ из хорошо определенных субклассов одновершинных /унимодальных/ функций /субклассц т.н. α-вогнутых функций/. Для доказательства результата автор сперва доказывает n-мерное расширение классического 1-мерного интегрального неравенства Хенстока и Мацбита /Henstock, Macbeat, Proc. London Math. Soc., Ser III., 3 (1953), 182-194/. Результат для ν_n, а также доказанное интегральное неравенство уточняют и обобщают все предыдущие результаты похожего типа.

EGY REDUKCIÓS TÉTEL ÖSSZEG-HALMAZOK MÉRTÉKEIRE AZ R^n-BEN

Uhrin B.

Összefoglaló

Legyen μ_k az R^k-ban levő Lebesgue-mérték, $0 \leq k \leq n$. Két $A,B \subset R^k$ L-mérhető halmazra definiáljuk $\lambda A \boxplus (1-\lambda)B := \{z \in R^k : \mu_k(\lambda A \cap (z-(1-\lambda)B)) > 0\}$, $0 \leq \lambda \leq 1$ (a halmazok "lényeges konvex kombinációja"). Legyenek $S \subset R^n$ és $T \subset R^n$ k- ill. $(n-k)$-dimenziós alterek, amelyek direkt összegben kifeszitik a teret. A szerző egy előbbi cikkében /Coll. Math. Soc. J. Bolyai, Vol 48, North-Holland, 1987, 551-571/ a $\mu_n(\lambda A \boxplus (1-\lambda)B)$ mértékre egy alsó becslést adott, amelyben a $\varphi(u) := \mu_k(A \cap (S+u))$, $\psi(u) := \mu_k(B \cap (S+u))$, $u \in T$, függvények lényeges supremumai ill. ezen függvények alsó szinthalmazaira vonatkozó lényeges konvex kombinációinak μ_{n-k} mértékei szerepelnek. Jelen cikkben a szerző ezt az eredményt olyan ν_n mértékre terjeszti ki, amelyeket az R^n-en definiált unimodális függvényosztály bizonyos jól definiálható alosztályaiban levő függvények generálnak (az u.n. α-konkáv függvények).
Az eredmény bizonyitásához a szerző először egy klasszikus 1-dimenziós integrál-egyenlőtlenség (Henstock, Macbeath, Proc. London Math. Soc., Ser III. 3 (1953), 182-194) n-dimenziós kiterjesztését bizonyitja be. Mind a bizonyitott n-dimenziós integrál-egyenlőtlenség, mind a ν_n-re vonatkozó eredmény az eddigi hasonló eredményeket élesiti és általánositja.

MTA SZTAKI Közlemények 36/1987 pp. 197-222

AN EXTENDED RELATIONAL DATABASE BY APPLICATION OF FUZZY SET THEORY AND LINGUISTIC VARIABLE

LIE TIEN VUONG[*] and *HO THUAN*[**]

[*] Institute of Computer Science and
Cybernetic - Hanoi - Vietnam

[**] Computer and Automation Institute
Hungarian Academy of Sciences

1. INTRODUCTION

The relational database have been studied since Codd [4]. Such database can only deal with well-defined and unambiguous data. But in the real world there exist data which can not be defined in certain and well-defined form by any means. The databases for above mentioned data have been investigated by different authors. [8,15] have developed the models for data with incomplete information and null-values. In [10,13,14] the authors have used the concept of linguistic variables to design intelligent database systems. The use of linguistic variable for a database is complicated but it is every important for describing objects that we do not have enough information such as "he is young", "A is far from B" ... These objects may be presented in a table as below:

STUDENT	NAME	AGE	HEIGHT
	A	20	about 1,70 m
	B	young	1,80
	C	about 25	high

The terms "young", "height", "about 25",... are called fuzzy terms. The fuzzy terms are a great class of data. To extened a database with fuzzy terms, the authors use in this paper the possibility distribution function [16] and multivalued

This work was supported by the National Scientific Research Fund No. 1066/1985

logic.

In section 2, the basic definitions of fuzzy set theory and linguistic variables are briefly mentioned. In section 3, we introduce the conceptual framework for a fuzzy database. The evaluation of a fuzzy query in a fuzzy database by relational algebra is presented in section 4. The section 5 extends the concept of data dependencies in relational database. In this section a concept of ternary degenerate decomposition of an extended relation is also introduced.

2. THE BASIC DEFINITION OF FUZZY SETS

In this section we shall briefly present the fuzzy notations and concepts which are minimally required for this paper. More details of discussions may be seen in [9,16].

Definition 2.1.

Let $U = \{u\}$ be a universe of discourse. A fuzzy set $\underset{\sim}{u}$ of U is a set of ordered pairs $\{(u, \mu_{\underset{\sim}{u}}(u))\}$, $u \in U$, where $\mu_{\underset{\sim}{u}}(u)$ is the grade of membership of u in $\underset{\sim}{u}$, and $\mu_{\underset{\sim}{u}} : U \to [0,1]$ is the membership function.

Definition 2.2.

Let $\underset{\sim}{u}$ and $\underset{\sim}{v}$ be two fuzzy sets of U.

a. Equality:

$\underset{\sim}{u}$ and $\underset{\sim}{v}$ are equal, written as $\underset{\sim}{u} \overset{f}{=} \underset{\sim}{v}$, iff $\mu_{\underset{\sim}{u}}(u) = \mu_{\underset{\sim}{v}}(u)$, $\forall u \in U$.

b. Containment:

$\underset{\sim}{u}$ is a subset of $\underset{\sim}{v}$, written as $\underset{\sim}{u} \overset{f}{\subseteq} \underset{\sim}{v}$, iff $\mu_{\underset{\sim}{u}}(u) \leq \mu_{\underset{\sim}{v}}(u)$ $\forall u \in U$.

c. Complementation:

The complement of a fuzzy set $\underset{\sim}{u}$ of U, denoted by $\overset{f}{\neg}\underset{\sim}{u}$, is defined by $\mu_{\overset{f}{\neg}\underset{\sim}{u}}(u) = 1 - \mu_{\underset{\sim}{u}}(u)$, $\forall u \in U$.

d. Union:

The union of $\underset{\sim}{u}$ and $\underset{\sim}{v}$, denoted by $\underset{\sim}{u} \overset{f}{\cup} \underset{\sim}{v}$, is defined by $\mu_{\underset{\sim}{u} \overset{f}{\cup} \underset{\sim}{v}}(u) = \mu_{\underset{\sim}{u}}(u) \vee \mu_{\underset{\sim}{v}}(u)$, $\forall u \in U$.

e. Intersection:

The intersection of $\underset{\sim}{u}$ and $\underset{\sim}{v}$, denoted by $\underset{\sim}{u} \overset{f}{\cap} \underset{\sim}{v}$, is defined by $\mu_{\underset{\sim}{u} \overset{f}{\cap} \underset{\sim}{v}}(u) = \mu_{\underset{\sim}{u}}(u) \wedge \mu_{\underset{\sim}{v}}(u)$, $u \in U$.

The symbols $\vee$ and $\wedge$ denote the maximum and the minimum, respectively.

Definition 2.3.

Let $\underset{\sim}{u}$ be a given fuzzy set of U. A λ-level fuzzy set, denoted by $\underset{\sim}{u}_\lambda$, is defined by

$$\underset{\sim}{u}_\lambda = \{(u, \mu_{\underset{\sim}{u}}(u)) \mid u \in u(\lambda)\}$$

where λ-level set $u(\lambda)$ is defined by

$$u(\lambda) = \{u \mid \mu_{\underset{\sim}{u}}(u) \geq \lambda, \quad u \in U\}, \qquad \lambda \in [0,1].$$

Definition 2.4

A linguistic variable is characterized by a quintuple (A,T(A),U,G,M) in which A is the name of the variable; T(A) (or simply T) denotes the term set of A, that is, the set of names of linguistic values of A; G is a syntactic rule for generating the names in T; M is a semantic rule for associating, with each t in T, its meaning M(t), which is a fuzzy set of U.

The meaning of a fuzzy term can be presented in the form $M(t) = \{(u, \mu_t(u)) \mid u \in U\}$.

It is easy to express a λ-level meaning of a fuzzy term $t \in T$. Let t be a linguistic value of A of universe discourse U. The λ-level meaning (or simply λ-meaning) $M_\lambda(t)$, $t \in T$ is a fuzzy set in the form $M_\lambda(t) = \{(u, \mu_t(u))\ u \in M_t(\lambda)\}$, where $M_t(\lambda)$ denoting λ-level set of fuzzy term t, is defined by

$$M_t(\lambda) = \{u \mid \mu_t(u) \geq \lambda,\ u \in U\}.$$

3. AN EXTENDED DATABASE BY APPLICATION OF FUZZY SETS AND LINGUISTIC VARIABLES

A relation over a set of attributes $W = \{A_1, \ldots, A_n\}$ is denoted by $R(W)$ (or simply R). Each attribute $A \in W$ is associated with a basic domain $U(A)$ which specifies all possible real values for A. Each basic domain can be extended by a corresponding set of linguistic values $T(A)$. Then the domain of the attribute A can be presented in the form $Dom(A) = U(A) \cup T(A)$ (or simply $D = U \cup T$).

A relation R over a set of attributes W is said to be a full relation of it contains no linguistic values, that is, for any $A \in W$, $Dom(A) = U(A)$ (i.e. $T(A) = \emptyset$). If R is not a full relation, that is $T(A) \neq \emptyset$ for some $A \in W$, then R is called extended relation.

We use A,B,C,... (or with indexes) to denote single attribute and X,Y,Z,.. (or with indexes) to denote sets of attributes of W. For a set of attributes $X \subseteq W$, a X-value is a mapping r that assigns to each attribute A_i of X an value from its domain $D(A_i) = U(A_i) \cup T(A_i)$ (or simply $D_i = U_i \cup T_i$). The value assigned to the attribute by such a mapping is denoted by $r[A_i]$. An extended relation over X is a set of X-values.

Without loss of generality it is assumes that the set of attributes W is finite. An extended relational database can be defined as follows:

Definition 3.1.

An extended relational database DB is defined as a set of extended relations R_i, $i = \overline{1,n}$, i.e.

$$DB = \{R_1,\ldots,R_n\},$$

in which every relation R_i is defined as a subset of the Cartesian product of a collection of domains, i.e.

$$R_i \subseteq \{U(A_{i_1}) \cup T(A_{i_1})\}\times\ldots\times\{U(A_{i_k}) \cup T(A_{i_k})\},$$

where $U(A_{i_j})$, $j = \overline{1,k}$ are basic domains and $T(A_{i_j})$, $j = \overline{1,k}$ are the set of fuzzy terms (linguistic values) of linguistic variables A_{i_j}.

To evaluate the meaning of any fuzzy term $t \in T(A_i)$, $A_i \in X$, in this paper can be used the techniques developed by [16]. The meaning of all values $u \in U(A_i)$, $A_i \in X$, is denoted by $M(u)$ and presented in the special form $M(u) = \{(u,1)\}$.

We introduce some (mathematical) concepts as follows.

Definition 3.2.

Let r_1, r_2 be two tuples of an extended relation $R(X)$ over the set of attributes $X \subseteq W$.

a. $r_1[A] \overset{\lambda}{\approx} r_2[A]$ iff $M_\lambda(r_1[A]) = M_\lambda(r_2[A])$ for $A \in X$,

$r_1[A],\ r_2[A] \in D(A),\ \lambda \in [0,1]$.

b. $r_1, r_2 \in R(X)$, $r_1 \overset{\lambda}{\approx} r_2$ iff $r_1[A] \overset{\lambda}{\approx} r_2[A]$ for all $A \in X$.

The relation $\overset{\lambda}{\approx}$ is called λ-level equivalence (or briefly λ -equivalence).

Remark.

If the relation R is full, then the concept of λ-level equivalence is identified with the equality of two real values, i.e. $r_1[A] = r_2[A]$, $r_1[A], r_2[A] \in U(A)$.

It is easy to show that the relation $\overset{\lambda}{\approx}$ is an equivalence relation.

In the following $\overset{\lambda}{\approx}$ is written by $\approx$ for sake of simplicity. Let W^* be the set of all possible tuples which are defined on W, i.e. it contains all X-values for all $X \subseteq W$. Every λ-level extended (written x-relation for short) is a λ-equivalence class defined by $\approx$. The class of relations equivalent to R is denoted $\bar{R}$ and R is called a representation of $\bar{R}$. Two relations R_1, R_2 over X are λ-equivalent, denoted by $R_1 \approx R_2$, iff

for $\forall r_1 \in R_1$, $\exists r_2 \in R_2$ such that $r_1 \approx r_2$ and

for $\forall r_2 \in R_2$, $\exists r_1 \in R_1$ such that $r_1 \approx r_2$.

Given a set of tuples $\{r_1, \ldots, r_n\}$, one can eliminate all tuples that are λ-equivalent to other tuples, and enlarge the others to their λ-equivalent X-values. The x-relation represented by the set of X-values so obtained will be denoted $\{r_1, \ldots, r_n\}_f$.

A tuple t is said to belong to or to be an element of $\bar{R}$, written $t \in \bar{R}$ when for some R' in $\bar{R}$, $t \in R'$

The following proposition is straightforward.

Proposition 3.1.

t is a tuple of $\bar{R}$ iff there exists a tuple r of R such that $r \approx t$.

In other words, a tuple t belongs to an x-relation iff its representation contains a tuple r which is λ-equivalent to t. An x-relation over $X \subseteq W$ is represented by the set of

X-values and will be denoted by the set $\{r_1,\dots,r_n\}_f$, in which all λ-equivalent tuples have been identified. The set operations on the set of x-relations can be defined as follows.

Let $\bar{R}_1$, $\bar{R}_2$ be two x-relations over X. We have

Union: $\bar{R}_1 \cup \bar{R}_2 = \{r \mid r \in R_1 \text{ or } r \in R_2\}_f$.

Intersection: $\bar{R}_1 \cap \bar{R}_2 = \{r \mid \exists r_1 \in R_1,\ r \approx r_1 \text{ and } \exists r_2 \in R_2,\ r \approx r_2\}_f$

Diference: $\bar{R}_1 \setminus \bar{R}_2 = \{r \mid r \in R_1 \text{ and } \nexists r_2 \in R_2 \text{ such that } r \approx r_2\}_f$.

Given a set of all λ-meanings of a linguistic variable A (domain of A is D = U $\cup$ T), denoted by $\mathcal{D}_\lambda$. Let $M_\lambda(u)$ and $M_\lambda(v)$ be λ-meanings of $u, v \in D$, respectively. Some operations in $\mathcal{D}_\lambda$ can be defined as follows:

Definition 3.3.

Union:

$$M_\lambda(u) \overset{f}{\cup} M_\lambda(v) = \{(u_i, \mu_u(u_i) \vee \mu_v(u_i)) \mid u_i \in M_u(\lambda) \cup M_v(\lambda)\}.$$

Intersection:

$$M_\lambda(u) \overset{f}{\cap} M_\lambda(v) = \{(u_i, \mu_u(u_i) \wedge \mu_v(u_i)) \mid u_i \in M_u(\lambda) \cap M_v(\lambda)\}$$

Complementation:

$$\overset{f}{\neg} M_\lambda(u) = \{(u_i,\ 1-\mu_u(u_i)) \mid u_i \in U,\ 1-\mu_u(u_i) \geq \lambda\}.$$

It is assumed that there exist in $\mathcal{D}_\lambda$ two elements MO and M1 of two values u_o and $u_1 \in D$, where MO and M1 are defined by:

$$MO = M(u_o) = \emptyset.$$

$$M1 = M(u_1) = \{(u_i, \mu_{u_1}(u_i)) \mid \mu_{u_1}(u_i) = 1 \text{ for } \forall u_i \in U\}.$$

MO is called λ-empty meaning and M1 is called λ-full meaning. Clearly, for all $M(u) \in \mathcal{D}_\lambda$ (the set of all λ-meanings of a linguistic variable A)

$$M(u) = M0 \text{ iff } \mu_u(u_i) < \lambda \text{ for } \forall u_i \in U \text{ and}$$

$$M(u) = M1 \text{ iff } \mu_u(u_i) = 1 \text{ for } \forall u_i \in U.$$

The λ-empty meaning and λ-full meaning have following properties:

For all $M(u) \in \mathcal{D}_\lambda$

$$M(u) \overset{f}{\cap} M0 = M0; \; M(u) \overset{f}{\cup} M0 = M(u);$$

$$M(u) \overset{f}{\cap} M1 = M(u); \; M(u) \overset{f}{\cup} M1 = M1.$$

Proposition 3.2.

The set of all λ-meanings $\mathcal{D}_\lambda$ of a linguistic variable A of an x-relation $\bar{R}(X)$ with the operations $\overset{f}{\cup}$, $\overset{f}{\cap}$ is a distributive lattice but not a Boolean algebra with the operations $\overset{f}{\cup}$, $\overset{f}{\cap}$ and $\overset{f}{\neg}$.

Proof.

It is easy to show, that all laws such as idempotency, commutativity, associativity, absorption and distributivity for the operations $\overset{f}{\cup}$ and $\overset{f}{\cap}$ are satisfied.

In order to show that $\mathcal{D}_\lambda$ is not Boolean algebra, we consider a following example.

Let us assume that there exist two elements of λ-empty meaning M0 and λ-full meaning M1. We must show that the laws of complementarity are not satisfied, i.e. $M(u) \overset{f}{\cap} \overset{f}{\neg} M(u) \neq M0$, and $M(u) \overset{f}{\cup} \overset{f}{\neg} M(u) \neq M1$ for some $M(u) \in \mathcal{D}_\lambda$.

Let $\lambda = 0.5$,

$$M(u) = \{(u_1, 0.3), (u_2, 0.6), (u_3, 0.7), (u_4, 0.8), (u_5, 1.), (u_6, 0.5), (u_7, 0.4)\}$$

$M_{0.5}(u) = \{(u_2,0.6),(u_3,0.7),(u_4,0.8),(u_5,1.),(u_6,0.5)\}$

$\overset{f}{\neg} M_{0.5}(u) = (u_1,0.7),(u_6,0.5),(u_7,0.6)$

$M_{0.5}(u) \overset{f}{\cap} \overset{f}{\neg} M_{0.5}(u) = \{(u_6,0.5)\} \neq M0.$

$M_{0.5}(u) \overset{f}{\cup} \overset{f}{\neg} M_{0.5}(u) = \{(u_1,0.7),(u_2,0.6),(u_3,0.7),(u_4,0.8),$

$(u_5,1.),(u_6,0.5),(u_7,0.6)\} \neq M1.$

4. QUERY EVALUATION

If a query Q with fuzzy terms is formulated on an extended database then there are three important bounds of interest:

(a). The set of all objects which surely satisfy the query Q, i.e. they satisfy Q with truth-value 1.

(b). The set of all objects which very probably satisfy the query Q, i.e. they satisfy Q with truth-value equal or greater than λ ($0 < \lambda < 1$).

(c). The set of all objects which may possibly satisfy the query Q, i.e. they satisfy Q with truth-value less than λ.

In this paper we are interested only in the problems (a) and (b) for a language based upon the relational algebra. The λ-value depends on database users. The problem (c) is very complicated and its part is investigated together with null--value problem in [8,14,15].

To evaluate a query in an extended database, the operations ∇ and Δ must be used here for defining the upper element and lower element of any two elements of lattice $\mathcal{D}_\lambda$ ($\mathcal{D}_\lambda$ is a partially ordered set), respectively (see [7]).

Let $M_\lambda(u)$, $M_\lambda(v)$ be two λ-meanings in $\mathcal{D}_\lambda$ in the form

$$M_\lambda(u) = \{(u_i,\ \mu_u(u_i))\,|\,u_i \in M_u(\lambda)\},\quad u \in D \quad \text{and}$$

$$M_\lambda(v) = \{(v_j,\ \mu_v(v_j))\,|\,v_j \in M_v(\lambda)\}\quad v \in D$$

or in other form (see definition 2.2)

$$M_\lambda(u) = \overset{f}{\bigcup_{u_i \in M_u(\lambda)}} \{(u_i, \mu_u(u_i))\} \quad \text{and}$$

$$M_\lambda(v) = \overset{f}{\bigcup_{v_j \in M_v(\lambda)}} \{(v_j, \mu_v(v_j))\}.$$

The operations are defined as follows:

$$M_\lambda(u) \nabla M_\lambda(v) = \overset{f}{\bigcup_{u_i \in M_u(\lambda)}} \{(u_i \vee v_j,\ \mu_u(u_i) \wedge \mu_v(v_j))\,|\,v_j \in M_v(\lambda)\} \quad \text{and}$$

$$M_\lambda(u) \Delta M_\lambda(v) = \overset{f}{\bigcup_{u_i \in M_u(\lambda)}} \{(u_i \wedge v_j,\ \mu_u(u_i) \wedge \mu_v(v_j))\,|\,v_j \in M_v(\lambda)\},$$

for all $u, v \in D$.

The predicate calculus based on languages contains two simple relational expressions such as $r_1[A]\Theta\ r_2[B]$ and $r[A]\Theta c$, where r_1, r_2 and r are tuple variables, A and B are attributes, c is a fuzzy constant from domain D(A), Θ is one of the comparision operations $=$, $\neq$, $>$, $\geq$, $<$, $\leq$. The evaluation value of an expression is in the interval [0,1].

W.l.o.g. the above expressions can be presented in the form of a simple fuzzy predicate, denoted by p:

$$p =: u\ \Theta\ v \quad \text{or} \quad p := u\ \Theta\ c, \quad \text{where} \quad u, v, c \in D.$$

Let f be an evaluation function with respect to p. The truth-

-value of function f with respect to p is a number τ of the set $\{0\} \cup [\lambda, 1]$. The Boolean operators AND, OR, NOT may be defined as τ AND $\tau' = \tau \wedge \tau'$, τ OR $\tau' = \tau \vee \tau'$ and

$$\text{NOT}\tau = \begin{cases} 1-\tau & \text{if } 1-\tau \geq \lambda \\ 0, & \text{otherwise} \end{cases}$$

where τ, τ' are numbers of the set $\{0\} \cup [\lambda, 1]$.

Now we consider the comparison operations $\Theta \in \{=, \neq, >, \geq, <, \leq\}$. These are defined as follows.

$$f(u = v) = \begin{cases} 1, & \text{if } M_\lambda(u) = M_\lambda(v),\ u, v \in D \\ \tau, & \text{if } \quad 1 > \tau \geq \lambda \\ 0, & \text{otherwise} \end{cases}$$

where τ is defined by

$$\tau = \frac{1}{1 + \left(\sum_{u_i \in M_u(\lambda) \cup M_v(\lambda)} (\mu_u(u_i) - \mu_v(u_i))^2 \right)^{1/2}}$$

$$f(u > v) = \begin{cases} 1, & \text{if } M_\lambda(u) \Delta M_\lambda(v) = M_\lambda(v) \quad \text{and} \quad M_\lambda(u) \neq M_\lambda(v) \\ 0, & \text{otherwise.} \end{cases}$$

$f(u < v)$ is defined analogously.

$$f(u \neq v) = \begin{cases} 0, & \text{if } M_\lambda(u) = M_\lambda(v), \\ \tau' = 1-\tau & \text{if } 1-\tau \geq \lambda \\ 1, & \text{otherwise.} \end{cases}$$

The remaining operations can be defined by

$$f(u \leq v) = f(u = v) \vee f(u < v),$$

$$f(u \geq v) = f(u = v) \vee f(u > v).$$

Based upon evaluation of a simple fuzzy predicate we can evaluate a fuzzy predicate expression as follows:

Let p and q be two simple fuzzy predicates. Then we have

$$f(p \text{ AND } q) = f(p) \wedge f(q).$$

$$f(p \text{ AND } q) = f(p) \vee f(q).$$

Given two tuples of W^* $r_1[X]$ and $r_2[X]$ W^* with $X \subseteq W$. The truth-value of an expression $r_1[X]\Theta r_2[X]$ is defined by the following equality

$$f(r_1[X]\Theta r_2[X]) = \bigwedge_{A_i \in X} (f(r_1[A_i]\Theta r_2[A_i])).$$

where $\Theta \in \{=, \neq, >, \geq, <, \leq\}$.

From the above concepts of an expression evaluation we can define the following relational operations.

Definition 4.1.

Let $\bar{R}(Y)$ be an x-relation. Let X be a subset of $Y \subseteq W$. The projection of $\bar{R}(Y)$ on X, denoted by $\bar{R}[X]$, is a set of tuples r for which

- there exists r' in $\bar{R}(Y)$ such that $r[X] \overset{\lambda}{\approx} r'[X]$,
- there exists no r'' in $\bar{R}(Y)$ such that $r''[X] \neq r'[X]$, $r''[X] \overset{\lambda}{\approx} r[X]$

i.e.

$$\bar{R}[X] = \{r[X] = (r[A_1],\ldots,r[A_k]) \mid r \in R \text{ and } A_i \in X,\ i=\overline{1,k}\}_f.$$

Definition 4.2.

Let $\bar{R}$ be an x-relation over X. A,B are two attributes of X and c is a fuzzy constant of D(A). The selection $A \Theta B$ and $A \Theta c$ can be defined by

$$\bar{R}[A\Theta B] = \{r \mid f(r[A]\Theta r[B]) \geq \tau, \quad r \in R, \quad \tau \geq \lambda\}_f ,$$

$$\bar{R}[A\Theta c] = \{r \mid f(r[A]\Theta c) \geq \tau, \quad r \in R, \quad \tau \geq \lambda\}_f ,$$

respectively, where Θ is one of $\{=, \neq, >, \geq, <, \leq\}$.

Definition 4.3.

Let $\bar{R}$ and $\bar{S}$ be two x-relations over XY[1)] and YZ, respectively, where X,Y and Z are subsets of W. The natural join of x-relation $\bar{R}$ and x-relation $\bar{S}$ on the common set of attributes is defined by

$$\begin{aligned} R[XY]*S[YZ] = \{r \mid & (\exists t \in R)(\exists s \in S)[(\forall A \in Y)(r[A] = t[A] \text{ or } \\ & r[A] = s[A])(f(t[A] = s[A]) = 1)] \text{ and } \\ & (\forall A \in X)[r[A] = t[A]] \text{ and } \\ & (\forall A \in Z)[r[A] = s[A]]\}_f . \end{aligned}$$

5. THE DATA DEPENDENCIES IN AN EXTENDED RELATIONAL DATABASE

5.1. Lossless decomposition of an extended relation

The concept of loss-less decomposition of a relation is very important in the process for designing a database, because instead of storing the relation R in the database, we can store only its projections.

---

1) *The union $X \cup Y$ is denoted by XY.*

In this paper we only investigate the lossless decomposition of an x-relation into a family of some of its projections. Let an x-relation $\bar{R}$ be a representation of a λ-equivalence class under ≈ in the universe x-relation R(W). $r_1[A] \stackrel{\lambda}{=} r_2[A]$ means that $r_1[A] \approx r_2[A]$, i.e. $M_\lambda(r_1[A]) = M_\lambda(r_2[A])$ for $r_1, r_2 \in R$ and $A \in W$. When no confusion occurs, in this section we write the symbol R instead of $\bar{R}$, being a λ-level extended relation.

Definition 5.1.

Let X,Y be two subsets of W with XY = W. An x-relation R(W) is said lossless decomposable (or simply: decomposable), denoted by $R(W) = R[X]*R[Y]$, if for all tuples pairs $r_1, r_2 \in R$ satisfying $r_1[X \cap Y] \stackrel{\lambda}{=} r_2[X \cap Y]$, there is a tuple $r \in R$ such that $r[X] \stackrel{\lambda}{=} r_1[X]$ and $r[Y] \stackrel{\lambda}{=} r_2[Y]$.

From the above definition 5.1 the concept of decomposition of an x-relation into n different projections can be generalized as follows.

Definition 5.2.

Let X_i, $i = \overline{1,n}$ be subsets of W with $\bigcup_{i=1}^{n} X_i = W$. An x-relation R(W) is said n-ary decomposable if for any n tuples $r_i \in R$, $i = \overline{1,n}$ such that $r_i[X_i \cap X_j] \stackrel{\lambda}{=} r_j[X_i \cap X_j]$, $i \neq j$, $i,j = \overline{1,n}$, there is a tuple $r \in R$ such that $r[X_i] \stackrel{\lambda}{=} r_i[X_i]$, $i = \overline{1,n}$.

Some important data dependencies can be defined as follows:

Definition 5.3.

Let $X \subseteq W$, $Y \subseteq W$. A λ-functional dependency (abbr. λFD) $X \overset{\lambda}{\rightarrow} Y$ holds in an x-relation $R(W)$ if for every two tuples $r_1, r_2 \in R$, $r_1[X] \overset{\lambda}{=} r_2[X]$ implies $r_1[Y] \overset{\lambda}{=} r_2[Y]$.

Definition 5.4.

Let $X \subseteq W$, $Y \subseteq W$ and $Z = W \setminus XY$.
A λ-multivalued dependency (abbr. λMVD) $X \overset{\lambda}{\twoheadrightarrow} Y|Z$ holds in an x-relation $R(W)$ if for every pair of tuples

$r_1 = (r_1[X], r_1[Y], r_1[Z])$ and $r_2 = (r_2[X], r_2[Y], r_2[Z])$ belong to R such that $r_1[X] \overset{\lambda}{=} r_2[X]$,

$r_3 = (r_1[X], r_1[Y], r_2[Z])$ and $r_4 = (r_1[X], r_2[Y], r_1[Z])$ belong to R, too.

A different way to view an λMVD and a decomposition is given below, which again is a generalization of a similar result of R. Fagin.

Proposition 5.1.

λMVD $X \overset{\lambda}{\twoheadrightarrow} Y$ holds in an x-relation $R(W)$, where $X \subseteq W$, $Y \subseteq W$, if and only if, whenever R is lossless decomposable in two projections $R[XY]$ and $R[X(W \setminus XY)]$.

Proof.

It is assumed that R is decomposable.
From definition 5.1 there are two tuples $r_1, r_2 \in R$ such that $r_1 = (r_1[X], r_1[Y], r_1[Z])$ and $r_2 = (r_2[X], r_2[Y], r_2[Z])$, where $Z = W \setminus XY$, $r_1[X] \overset{\lambda}{=} r_2[X]$. Then there is tuple $r_3 \in R$ such that

$r_3[XY] \overset{\lambda}{=} r_1[XY] = (r_1[X], r_1[Y])$ and

$r_3[XZ] \overset{\lambda}{=} r_{.}[XZ] = (r_1[X], r_2[Z])$.

This means, that $r_3 = (r_1[X], r_1[Y], r_2[Z]) \in R$.
Similarly there is $r_4 = (r_1[X], r_2[Y], r_1[Z]) \in R$. Then $X \overset{\lambda}{\twoheadrightarrow} Y$ holds in R. The converse is also easily shown.

The reader can verify that the inference rules, as has been done for FD and MVD [2] satisfy for λFD and λMVD, too. The inference rules are presented in following:
X,Y,Z and V are subsets of W.

<u>λFD inference rules</u>

λFD1 : if $Y \subseteq X$ then $X \overset{\lambda}{\rightarrow} Y$.

λFD2 : if $Z \subseteq V$ and $X \overset{\lambda}{\rightarrow} Y$ then $XV \overset{\lambda}{\rightarrow} YZ$.

λFD3 : if $X \overset{\lambda}{\rightarrow} Y$ and $Y \overset{\lambda}{\rightarrow} Z$ then $X \overset{\lambda}{\rightarrow} Z$.

λFD4 : if $X \overset{\lambda}{\rightarrow} Y$ and $YV \overset{\lambda}{\rightarrow} Z$ then $XV \overset{\lambda}{\rightarrow} Z$.

λFD5 : if $X \overset{\lambda}{\rightarrow} Y$ and $X \overset{\lambda}{\rightarrow} Z$ then $X \overset{\lambda}{\rightarrow} YZ$.

λFD6 : if $X \overset{\lambda}{\rightarrow} YZ$ and $X \overset{\lambda}{\rightarrow} Y$ then $X \overset{\lambda}{\rightarrow} Z$.

<u>λMVD inverence rules</u>

λMVD0 : $X \overset{\lambda}{\twoheadrightarrow} Y$ iff $X \overset{\lambda}{\twoheadrightarrow} W \setminus Y$.

λMVD1 : if $Y \subseteq X$ then $X \overset{\lambda}{\twoheadrightarrow} Y$.

λMVD2 : if $Z \subseteq V$ and $X \overset{\lambda}{\twoheadrightarrow} Y$ then $XV \overset{\lambda}{\twoheadrightarrow} YZ$.

λMVD3 : if $X \overset{\lambda}{\twoheadrightarrow} Y$ and $Y \overset{\lambda}{\twoheadrightarrow} Z$ then $X \overset{\lambda}{\rightarrow} Z \setminus Y$.

λMVD4 : if $X \overset{\lambda}{\twoheadrightarrow} Y$ and $XV \overset{\lambda}{\twoheadrightarrow} Z$ then $XV \overset{\lambda}{\twoheadrightarrow} Z \setminus YV$.

λMVD5 : if $X \overset{\lambda}{\twoheadrightarrow} Y$ and $X \overset{\lambda}{\twoheadrightarrow} Z$ then $X \overset{\lambda}{\twoheadrightarrow} YZ$.

λMVD6 : if $X \overset{\lambda}{\twoheadrightarrow} Y$ and $X \overset{\lambda}{\twoheadrightarrow} Z$ then $X \overset{\lambda}{\twoheadrightarrow} Y \cap Z$, $X \overset{\lambda}{\twoheadrightarrow} Y \setminus Z$, $X \overset{\lambda}{\twoheadrightarrow} Z \setminus Y$.

λFD-λMVD inference rules

λFD-λMVD1 : if $X \overset{\lambda}{\rightarrow} Y$ then $X \overset{\lambda}{\twoheadrightarrow} Y$.

λFD-λMVD2 : if $X \overset{\lambda}{\twoheadrightarrow} Z$ and $Y \overset{\lambda}{\rightarrow} Z'$, $(Z' \subseteq Z,\ Y \cap Z = \emptyset)$ then $X \overset{\lambda}{\rightarrow} Z'$.

λFD-λMVD3 : if $X \overset{\lambda}{\twoheadrightarrow} Y$ and $XY \overset{\lambda}{\twoheadrightarrow} Z$ then $X \overset{\lambda}{\rightarrow} Z \setminus Y$.

Let $\mathcal{B}$ be a family of all n-ary decompositions of the x-relation R over W (denoted by $(X_1,\ldots,X_n)$). This family has the following properties.

Theorem 5.1. [6].

Let R an x-relation over W. $X_i \subseteq W$, $i = \overline{1,n}$, $\bigcup_{i=1}^{n} X_i = W$. The family $\mathcal{B}$ of all n-ary decompositions of the x-relation R satisfies the following conditions:

1. $(\emptyset,\ldots,\emptyset,W) \in \mathcal{B}$.
2. If $(X_1,\ldots,X_n) \in \mathcal{B}$ then $(X_{\pi(1)},\ldots,X_{\pi(n)}) \in \mathcal{B}$ where $\pi: \{1,\ldots,n\} \to \{1,\ldots,n\}$ is a permutation.
3. If $(X_1,\ldots,X_n) \in \mathcal{B}$ and $X_i \subseteq Y \subseteq W$, $i = \overline{1,n}$, then $(X_1,\ldots,X_{i-1},Y,X_{i+1},\ldots,X_n) \in \mathcal{B}$.
4. If $(X_1,\ldots,X_n) \in \mathcal{B}$ and $X_i \subseteq X_j$, $i \neq j$ then $(X_1,\ldots,X_{i-1},\emptyset,X_{i+1},\ldots,X_j,\ldots,X_n) \in \mathcal{B}$.
5. If $(X_1,\ldots,X_n)$ and $(Y_1,\ldots,Y_n) \in \mathcal{B}$ with $Y_1 \cap Y_i = X_i$, $i = \overline{2,n}$ and $Y_i \cap Y_j \subseteq X_i \cap X_j$, $i \neq j$, $i,j = \overline{2,n}$ then $(X_1 \cap Y_1, Y_2,\ldots,Y_n) \in \mathcal{B}$.

Proof.

Like the proof in [6].

5.2. Ternary degenerate decomposition of an x-relation

An interesting class of decompositions of an x-relation which plays an important role in the design process of a data-base, is the acyclic decomposition [3].

In [5] the author has investigated the general properties of a full family of all ternary decompositions of a relation. In this subsection we will only consider the class of ternary acyclic decompositions of an x-relation. (The following results are correct for a ternary acyclic decomposition of an usual relation). Let (X,Y,Z) be a ternary decomposition of an x-relation R(W). If X,Y,Z are no-empty subsets of W with $XYZ = W$, $X \neq Y \neq Z$ and $X \cap Y \neq \emptyset$, $Y \cap Z \neq \emptyset$ and $X \cap Z = \emptyset$ then this decomposition is said acyclic.

Proposition 5.2. [3,6]

Let R be an x-relation over W. X,Y,Z are no empty subsets of W with $XYZ = W$ and $X \cap Y \neq \emptyset$, $Y \cap Z \neq \emptyset$, $X \cap Z = \emptyset$. (X,Y,Z) is a ternary acyclic decomposition of x-relation R iff the following data dependencies are satisfied in R:

$$X \cap Y \overset{\lambda}{\twoheadrightarrow} X,\quad (X \cap Y)(Y \cap Z) \overset{\lambda}{\twoheadrightarrow} Y \quad\text{and}\quad (Y \cap Z) \overset{\lambda}{\twoheadrightarrow} Z.$$

Proof.

It is easy to verify.

It follows directly from definition 5.2 and theorem 5.1 the following:

Proposition 5.3.

If (X,Y,Z) is a ternary acyclic decomposition of an x-relation R over W, then (XY,Z),(X,YZ),(XY,YZ) are binary decompositions of R and (X,Y),(Y,Z) binary decompositions of projections R[XY] and R[YZ], respectively.

To capture more the semantics of data,we use here the concept of degenerate multivalued dependencies [12] for determining which join of relations can be updated by insertion or deletion of a tuple without other tuples entering or leaving the join.

Definition 5.5. [12]

A λ-multivalued degenerate dependency $X \overset{\lambda}{\twoheadrightarrow} Y|Z$ holds in an x-relation $R(W)$ with $Z = W \setminus XY$ if for every pair of tuples $r_1, r_2 \in R$, $r_1[X] \overset{\lambda}{=} r_2[X]$, either $r_1[Y] \overset{\lambda}{=} r_2[Y]$ or $r_1[Z] \overset{\lambda}{=} r_2[Z]$.

The definition 5.5 can be reformulated in the other form of the corresponding binary decomposition as follows.

Definition 5.6.

Let R be an x-relation over W and (X,Y) be a binary decomposition of R.
(X,Y) is binary degenerate decomposition of R if there exist two x-relations R_1 and R_2 such that $R = R_1 \cup R_2$, $X \cap Y \overset{\lambda}{\to} X$ holds in R_1 and $X \cap Y \overset{\lambda}{\to} Y$ holds in R_2 with $R_1[X \cap Y] \cap R_2[X \cap Y] = \emptyset$.

In [1] the authors have studied the effect of type of binary degenerate multivalued dependency for the update problem. For our purpose, the most important of this result is that a relation R can be deletion- or insertion- viable if and only if a certain degenerate decomposition holds in R.

Attempts to generalize this result to decompositions of an x-relation (in meaning of λ-equivalence) into more than two components we introduce a new concept called ternary degenerate decomposition.

Definition 5.7.

A ternary acyclic decomposition (X,Y,Z) of an x-relation R over W is degenerate if there exist two x-relations R_1 and R_2 such that $R = R_1 \cup R_2$, $X \cap Y \overset{\lambda}{\to} YZ$ holds in R_1 and

$Y \cap Z \overset{\lambda}{\rightarrow} XY$ holds in R_2 with $R_1[X \cap Y] \cap R_2[X \cap Y] = \emptyset$ or $R_1[Y \cap Z] \cap R_2[Y \cap Z] = \emptyset$.

In the following it is assumed that (X,Y,Z) is a ternary acyclic decomposition of R(W). This decomposition has some properties as follows.

Proposition 5.4.

If (X,Y,Z) is a ternary degenerate decomposition of R, then (X,YZ),(XY,Z) and (XY,YZ) are binary degenerate decompositions of R; (X,Y) and (Y,Z) are binary degenerate decompositions of the corresponding projection R[XY] and R[YZ].

Proof.

It must be show that (X,YZ) is a binary degenerate decomposition. Since (X,Y,Z) is degenerate, then there exist R_1 and R_2 such that $R = R_1 \cup R_2$ and $X \cap Y \overset{\lambda}{\rightarrow} YZ$ in R_1, $Y \cap Z \overset{\lambda}{\rightarrow} XY$ in R_2. W.l.o.g. it is assumed that $R_1[X \cap Y] \cap R_2[X \cap Y] = \emptyset$.

First we have to show that (X,YZ) is a binary degenerate decomposition of R.
We have $X \cap Y \overset{\lambda}{\rightarrow} YZ$ that holds in R_1. It remains only to show that $X \cap Y \overset{\lambda}{\rightarrow} X$ holds in R_2.
Since (X,YZ) is a binary decomposition and $R_1[X \cap Y] \cap R_2[X \cap Y] = \emptyset$, the following data dependencies hold in R_2:

$$X \cap Y \overset{\lambda}{\twoheadrightarrow} X, \quad Y \cap Z \overset{\lambda}{\rightarrow} XY.$$

Hence $Y \cap Z \overset{\lambda}{\rightarrow} X$.
After the applications of inference rule λFD- MVD2 we have $X \cap Y \overset{\lambda}{\rightarrow} X$.

In order to show that (X,YZ) is degenerate under the above assumptions, we must construct R_1' and R_2' as follows. Since (XY,Z) is a binary decomposition then it satisfies the following conditions: for any pair of tuples $r_1, r_2 \in R$ with

$r_1[Y \cap Z] \overset{\lambda}{=} r_2[Y \cap Z]$, there exists a tuple $r \in R$ such that $r[XY] \overset{\lambda}{=} r_1[XY]$ and $r[Z] \overset{\lambda}{=} r_2[Z]$.

Three cases can happen.

- Case 1: if $r_1, r_2 \in R_1$ then it is easy to see that $r \in R_1$, too.
- Case 2: of $r_1, r_2 \in R_2$ then it is easy to see that $r \in R_2$, too.
- Case 3: w.l.o.g. it is assumed that $r_1 \in R_1$ and $r_2 \in R_2$.

If $r \in R_2$ then $r[XY] \overset{\lambda}{=} r_1[XY]$ and $r[Z] \overset{\lambda}{=} r_2[Z]$.
It follows that $r[X \cap Y] \overset{\lambda}{=} r_1[X \cap Y]$, which contradicts to the fact that $R_1[X \cap Y] \cap R_2[X \cap Y] = \emptyset$. It remains only the case that $r \in R_1$.
Since $r[XY] \overset{\lambda}{=} r_1[XY]$ and $X \cap Y \overset{\lambda}{\rightarrow} YZ$ holds in R_1, then $r \equiv r_1$ in R_1. Hence $r_1[Z] \overset{\lambda}{=} r_2[Z]$.
This shows that if there exists a tuple $r_1 \in R_1$ such that $r_1[Y \cap Z] \overset{\lambda}{=} r_2[Y \cap Z]$ where $r_2 \in R_2$, then it must satisfy $r_1[Z] \overset{\lambda}{=} r_2[Z]$. All such tuples of R_2 can be grouped to a set T. Then two x-relations R_1' and R_2' can be constructed as follows:

$$R_1' = R_1 \cup T, \quad R_2' = R_2 \setminus T.$$

We have also $R_1'[Y \cap Z] \cap R_2'[Y \cap Z] = \emptyset$ and the data dependencies $Y \cap Z \overset{\lambda}{\rightarrow} Z$ and $Y \cap Z \overset{\lambda}{\rightarrow} XY$ holds respectively in R_1' and R_2'. Hence (XY, Z) is a degenerate decomposition of R.

All remaining decompositions are easy to verify to be degenerate of R.

Proposition 5.5.

If (XY, Z) and (X, YZ) are degenerate decompositions of

R, then (X,Y) and (Y,Z) are degenerate decompositions of R[XY] and R[YZ], respectively.

Proof. Trivial.

Propositions 5.6.

If (XY,Z) is a degenerate decomposition of R and (X,Y) is a degenerate decomposition of R[XY], then (X,YZ) is a degenerate decomposition of R and (Y,Z) is a degenerate decomposition of R[YZ].

Proof.

Since (XY,Z) is a degenerate, then there exists two x-relations R_1 and R_2 with $R = R_1 \cup R_2$ such that $Y \cap Z \stackrel{\lambda}{\rightarrow} XY$ holds in R_1 and $Y \cap Z \stackrel{\lambda}{\rightarrow} Z$ holds in R_2 and $R_1[Y \cap Z] \cap R_2[Y \cap Z] = \emptyset$.

We want to construct now two x-relations R_1' and R_2' which satisfy all conditions of (X,YZ) to be degenerate. Since (X,YZ) is a binary decomposition, then for any pair of tuples $r_1, r_2 \in R$ with $r_1[X \cap Y] \stackrel{\lambda}{=} r_2[X \cap Y]$, there exists $r \in R$ such that $r[X] \stackrel{\lambda}{=} r_1[X]$ and $r[YZ] \stackrel{\lambda}{=} r_2[YZ]$.

There are three cases:

Case 1.

If $r_1, r_2 \in R_1$ then $r \in R_1$. Since, if $r \in R_2$ then $r[YZ] \stackrel{\lambda}{=} r_2[YZ]$. Therefore $r[Y \cap Z] \stackrel{\lambda}{=} r_2[Y \cap Z]$. It is contrary to $R_1[Y \cap Z] \cap R_2[Y \cap Z] = \emptyset$.

Case 2.

Similarly, it is shown that if $r_1, r_2 \in R_2$ then $r \in R_2$.

Case 3.

W.l.o.g. it is assumed that $r_1 \in R_1$ and $r_2 \in R_2$.

If $r \in R_1$ then $r[Y \cap Z] \overset{\lambda}{=} r_2[Y \cap Z]$. It is contrary to $R_1[Y \cap Z] \cap R_2[Y \cap Z] = \emptyset$.

There is also only the case that $r_1 \in R_1$ and $r, r_2 \in R_2$. Because (X,Y) is degenerate decomposition in $R[XY]$, the tuples r_1, r_2 satisfy $r_1[X \cap Y] \overset{\lambda}{=} r_2[X \cap Y]$, then $r_1[X] \overset{\lambda}{=} r_2[X]$ or $r_1[Y] \overset{\lambda}{=} r_2[Y]$. But from degenerate decomposition (XY,YZ) we get $r_1[Y] \neq r_2[Y]$. It follows that $r_1[X] \overset{\lambda}{=} r_2[X]$. This means, that $r \equiv r_2 \in R_2$. The last case show that if there exists a tuple $r \in R_2$ such that $r[X \cap Y] \overset{\lambda}{=} r'[X \cap Y]$, $r' \in R_1$, it must satisfy $r[X] \overset{\lambda}{=} r'[X]$. All such tuples of R_2 can be grouped to a set T. Then two x-relations can be constructed as follows:

$$R_1' = R_1 \cup T \quad \text{and} \quad R_2' = R_2 \setminus T.$$

We have also $R = R_1' \cup R_2'$, $R_1'[X \cap Y] \cap R_2'[X \cap Y] = \emptyset$. From the construction of R_1' and R_2' as above, it is easily seen that the data dependency $X \cap Y \overset{\lambda}{\to} X$ holds in R_1' and the data dependency $X \cap Y \overset{\lambda}{\to} YZ$ holds in R_2'. Therefore (X,YZ) is a degenerate decomposition of R.

REFERENCES

[1] W.W. Armstrong, C. Delobel: Decompositions and functional dependencies in relations, ACM TODS 5,4 (1980), 404-430.

[2] C. Beeri, R. Fagin, J.H. Howard: A complete axiomatization for functional and multivalued dependencies in database relations, Proc. 1977 ACM SIGMOD, Toronto, 47-61.

[3] C. Beeri, R. Fagin, D. Maier, M. Yannakakis: On the desirability of acyclic database schemes, J.ACM 30,3(1983), 479-513.

[4] E.F. Cood: A relational model for large schared data banks, C.ACM 13,6(1970) 377-387.

[5] Le Tien Vuong: Untersuchung zur ternaeren Dekomposition einer Relation und zur Anwendung der unscharfen Mengen im CRM, Diss. TU-Dresden 1983.

[6] Le Tien Vuong: Über n-fache Dekomposition einer Relation im Coddschen Relationenmodell, MTA SZTAKI Közlemenyek, 31/1984, 95-113.

[7] Le Tien Vuong and Ho Thuan: Retrieval from fuzzy database by fuzzy relational algebra, to appear in MTA SZTAKI Közlemények 37/1987.

[8] Y.E. Lien: Multivalued dependencies with null values in relational databases, 5th Int. Conf. VLDB, Rio de Janeiro, 1979, 155-168.

[9] C.V. Negoita, D.A. Ralesa: Applications of fuzzy sets to systems analysis, Birk. Verlag, 1975.

[1o] H. Prade, C. Testemale: Traitement de questions vagues dans une base de données imprécises 1,2, L'Infor. Process. 27(1984) 65-84, 28(1984) 49-66.

[11] R. Fagin: Multivalued dependencies and a new Normal Form for relational databases, ACM TODS 2,3(1977), 262-278.

[12] Y. Sagiv, C. Delobel, D.S. Jr. Parker, R. Fagin: An equivalence between relational database dependencies and a fragment of propositional logic, J.ACM 28,3(1981), 435-453.

[13] V. Tahani: A conceptual framework for fuzzy query processing-a step toward very intelligent database systems, Inf. Proc. and Manag., 13(1977), 289-303.

[14] M. Umano, M. Mizimoto, K. Tanaka: Fuzzy database systems, Proc. of Working Conf. on database Engineering (13th IBM Computer Science Symp. IBM Amagi Homestest, Japan 1979, 17-19.

[15] C. Zaniolo: Database relations with null values, J. of Computer and Syst. Sciences 28(1984) 142-166.

[16] L.A. Zadeh: The concept of a linguistic variable and its applications to approximate reasoning, Inf. Science 8(1975/ 199-248 and 301-357.

Обобщение реляционных баз данных применяя теорию "фаззи" множеств и лингвистических переменных

Лие Тиен Вуонг, Хо Тхуан

Резюме

В реальном мире существуют данные,содержание которых очень неточное /например, "он молодой"/. Для разработки таких данных и для построения теоретических основ таких баз данных уже применяется теория "фаззи" множеств. Другой подход - ввести в базу данных переменные нового типа, т.н. лингвистические переменные. В статье разработана теория баз данных, используя оба метода.

A "FUZZY" HALMAZ-ELMÉLET ILL. "LINGVISZTIKAI" VÁLTOZÓK ÁLTAL KIBŐVITETT RELÁCIÓS ADATBÁZISOK

Lien Tien Vuong, Ho Thuan

Összefoglaló

A valós világban olyan adatok is vannak, amelyek tartalma eléggé pontatlan (pl. "fiatal"). Az ilyen adatok feldolgozására "fuzzy" ("elmosódott") halmazelméleten alapuló ujfajta adatbázis-elméletet dolgoztak ki. Egy másik módszer az u.n. "lingvisztikai" ("nyelvészeti") változók bevezetése. A cikkben a két módszer alapján egy újfajta relációs adatbázis-elmélet részletes ismertetése van.

ALFAPRINT