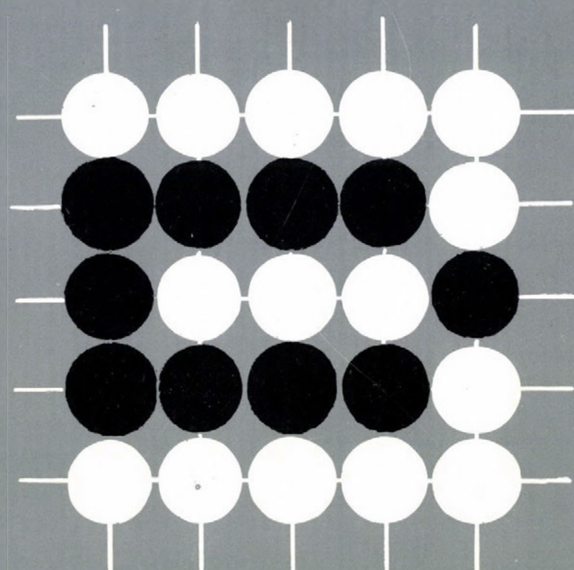
közlemények

34/1986

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

34/1986

Felelős kiadó:

KEVICZKY LÁSZLÓ

ISSN 0133-7459

Készült az Országos Széchényi Könyvtár Sokszorosító üzemében
Budapest, 1986-ban
Felelős vezető: Rosta Lajosné
Példányszám: 390, terjedelem: 23 A/5 ív.
Munkaszám: 86 241

C O N T E N T S

СОДЕРЖАНИЕ

MTA SZTAKI Közlemények 34/1986 pp.5-35

ТАБЛИЦЫ КОНЪЮНКТИВНЫХ ЗАПРОСОВ И ИХ ПРИМЕНЕНИЕ

Галя Ангелова

Лаборатория математической лингвистики

Институт математики с вычислительным центром

Болгарская академия наук

1. ВВЕДЕНИЕ

Понятие "таблица" введено в [1] и [7] как средство изучения класса реляционных запросов, которые в реляционной алгебре представлены при помощи выражений, содержащих только операции селекцию σ, проекцию Π и (естественного) соединение J. Это так называемые SPJ - выражениям, составляющие важный класс выражений в реляционной алгебре. Каждому SPJ - выражению поставлена в соответствие таблица, а путем преобразования этой таблицы возможно получить множество SPJ - выражений, эквивалентных данному; возможно также выделить среди них то SPJ - выражение, которое содержит наименьшее число J - операций и таким образом оптимизировать начальное SPJ - выражение по отношению числа J - операций. При помощи понятия таблицы можно представить необходимое и достаточное условие эквивалентности двух заданных SPJ - выражений [7]. Так как между конъюнктивными запросами и SPJ - выражениями существует соответствие, то при помощи таблиц конъюнктивных запросов можно исследовать проблему эквивалентности данных конъюнктивных запросов. Понятие таблицы играет важную роль и в алгоритме интерпретации запросов пользователя в System/U ([6], [7]), где оно применяется для нахождения оптимального внутреннего представления запроса, которое является эквивалентным начальному представлению запроса.

2. ОСНОВНЫЕ ПОНЯТИЯ

В соответствии с [2], будем использовать следующие понятия и обозначения:

Реляционной схемой R будем называть конечное множество имен атрибутов $\{A_1, A_2, \ldots, A_n\}$. Будем записывать $R = \{A_1, A_2, \ldots, A_n\}$. Реляционные схемы будем обозначать через $R_1, R_2, \ldots, R_k, \ldots$.

Каждому атрибуту A_i сопоставляется множество значений - так называемый домен. Домен атрибута A_i будем обозначать через dom (A_i). Если dom $(A_i) = \{c_1, c_2, \ldots, c_m, \ldots\}$, каждое значение c_i будем называть константой домена dom (A_i).

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

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

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

Пусть $\alpha = <a_1 a_2 \ldots a_n>$ - кортеж отношения $r(A_1 A_2 \ldots A_n)$. Тогда $\alpha[A_i] = a_i$, т.е. через $\alpha[A_i]$ будем обозначать значение кортежа α для атрибута A_i. Если $X = \{A_{i_1}, A_{i_2}, \ldots, A_{i_k}\}$ то $\alpha[X] = <\alpha[A_{i_1}]\ \alpha[A_{i_2}] \ldots \alpha[A_{i_k}]> = <a_{i_1} a_{i_2} \ldots a_{i_k}>$.

Будем использовать определения операций селекции, проекции и (естественного) соединения.

Пусть $R = \{A_1, A_2, \ldots, A_n\}$, $r(R)$ и $\alpha \in dom(A_i)$.

Тогда

- Селекция $A_i \Theta a$ (обозначаемая через $\sigma_{A_i \Theta a}(r)$) представляет собой:

$$\sigma_{A_i \Theta a}(r) = \{\alpha \mid \alpha \in r \text{ и } \alpha(A_i)\Theta a\} .$$

(Θ - одно из бинарных отношений $=, <, >, \leqq, \geqq, \neq$). Таким образом из отношения r берутся только те кортежи, имеющие в атрибуте A_i значение b и $b \Theta a$. $\sigma_{A_i \Theta a}(r)$ является отношением над множеством атрибутов $\{A_1, A_2, \ldots, A_n\}$ и следовательно представляет собой подмножество отношения r.

- Проекция $\Pi_Y(r)$ представляет собой:

$$\Pi_Y(r) = \{\alpha [Y] \mid \alpha \in r \text{ и } \{A_1, A_2, \ldots, A_n\} \supseteq Y\},$$

т.е. из всех возможных кортежей отношения r берутся только значения атрибутов множества Y и одинаковые кортежи отождествляются. $\Pi_Y(r)$ является отношением над множеством атрибутов Y.

- (Естественное) соединение $r_1 \bowtie r_2$.

Пусть R_1 и R_2 являются реляционными схемами и $r_1(R_1)$ и $r_2(R_2)$. Тогда

$$r_1 \bowtie r_2 = \{\alpha \mid \alpha \text{ является кортежем над атрибутами } R_1 \cup R_2 \text{ и существуют кортежи } \alpha_1 \in r_1 \text{ и } \alpha_2 \in r_2\text{, такие, что } \alpha_1 = \alpha[R_1] \text{ и } \alpha_2 = \alpha[R_2]\}.$$

<u>Определение 1.</u>

SPJ - выражения будем называть выражениями реляционной алгебры, если:

а) операнды выражений являются реляционными схемами;

б) операции выражений представляют собой селекцию, проекцию и (естественное) соединение, т.е. эти выражения являются формулами над σ, Π, J и именами реляционных схем.

<u>Определение 2.</u>

Конъюнктивным запросом в реляционном языке запросов будем называть выражение вида:

(1) $\{a_1 a_2 \ldots a_n \mid (\exists b_1) \ldots (\exists b_m) (P_1 \wedge P_2 \wedge \ldots \wedge P_i \wedge \ldots \wedge P_k)\}$

где P_i $(1 \leqq i \leqq k)$ терм вида а) или б);

а) $R(c_1 c_2 \ldots c_s)$, это означает, что кортеж $c_1 c_2 \ldots c_s$ принадлежит отношению над реляционной схемой R. Здесь $c_j (1 \leqq j \leqq S)$ являются константами соответствующего домена или $c_j \in \{a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m\}$;

б) $c \ominus d$, где c и d - либо константы соответствующих доменов, либо элементы множества $\{a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m\}$. Здесь $\ominus$ одно из бинарных отношений $=, <, >, \leqq, \geqq, \neq$.

В выражении (1) символы $a_1, a_2, \ldots, a_n$ - свободные переменные, а $b_1, b_2, \ldots, b_m$ связанные переменные.

Пример 1.

Рассмотрим следующую базу данных, состоящую из пяти примерных реляционных схем:

ЧАСТЬ (ЧИМЯ, ЧНОМЕР, ЦЕНА)

ПОСТАВЩИК (ПИМЯ, ПНОМЕР, ПАДРЕС, ПГОРОД)

КЛИЕНТ (КИМЯ, КНОМЕР, КАДРЕС, КГОРОД)

ПОСТАВКА (ЧНОМЕР, ПНОМЕР, КНОМЕР, КОЛИЧЕСТВО)

ОБЯЗАННОСТЬ (ЧНОМЕР, ПНОМЕР).

Реляционные схемы нормализованы в третьей нормальной форме [7]. Ключи подчеркнуты [7]. Отношение ОБЯЗАННОСТЬ дает информацию об обязанностях, присущих каждому поставщику.

В качестве примера конъюнктивного запроса к этой базе данных можно рассмотреть следующие запросы:

q_1 : Найти имена всех поставщиков, живущих в городе c_1.

Этот запрос можно представить и следующим образом:

(2) $\{a_1 \mid (\exists b_1)(\exists b_2)$ такие, что ПОСТАВЩИК $(a_1, b_1, b_2, c_1)\}$.

q_2 : Найти имена всех поставщиков, поставляющих часть c_1 клиентам из города c_2.

Этот запрос можно записать при помощи выражения:

(3) $\{a_1 \mid (\exists b_1)(\exists b_2)(\exists b_3)(\exists b_4)(\exists b_5)(\exists b_6)(\exists b_7)(\exists b_8)(\exists b_9)$

ПОСТАВЩИК (a_1, b_1, b_2, b_3)

$\wedge$ КЛИЕНТ (b_4, b_5, b_6, c_2)

$\wedge$ ЧАСТЬ (c_1, b_7, b_8)

$\wedge$ ПОСТАВКА $(b_7, b_1, b_5, b_9)\}$

Как известно, каждый конъюнктивный запрос может быть представлен как SPJ - выражение и наоборот, каждому SPJ - выражению соответствует конъюнктивный запрос [7]. По этой причине мы будем строить таблицы [7] для SPJ - выражений и часто будем интерпретировать эти таблицы как конъюнктивные запросы.

Определение 3.

Введем понятие таблицы, строя таблицу для конъюнктивного выражения (1).

Каждая таблица представляет собой двумерную матрицу. Столбцы матрицы соответствуют заданному множеству атрибутов

$\{A_1, A_2, \ldots, A_l\}$, участвующих в выражение (1). Таблица может содержать произвольное число строк. Ее элементами (символами) могут быть:

а) свободные переменные - соответствуют $a_1, a_2, \ldots, a_n$ в (1). Свободные переменные в таблицах будем обозначать через букву а с нижным индексом;

б) связанные переменные - соответствуют $b_1, b_2, \ldots, b_m$ в (1). Будем обозначать их через букву b с нижным индексом;

в) константы - полагается, что константы, находящиеся в j-том столбце, принадлежат домену, соответствующему атрибуту A_j;

г) пробелы (пустые символы).

Над таблицей (или в качестве ее нулевой строки) задаются все атрибуты, участвующие в выражении (1). В следующей строке (она является первой строкой таблицы) находятся все свободные переменные и могут находиться константы и пробелы. Эта строка называется резюме таблицы. Способ расположения свободных переменных в резюме таблицы показывает к каким атрибутам следует их отнести. Например, $a_1 a_2$ не означает, что a_1 является свободной переменной над атрибутом A_1, а a_2 - свободной переменной над атрибутом A_2. Запись $a_1 a_2$ имеет смысл только в конкретной таблице, причем расположение переменных a_1 и a_2 в резюме показывает к каким атрибутам относятся эти две переменные. Остальные позиции резюме - пустые или содержат константы.

Остальные строки будем называть просто "строками" таблицы и будем их использовать для описания термов вида $R(c_1, c_2, \ldots, c_s)$ в пункте а) определения 2. Каждому терму $R(c_1, \ldots, c_s)$ отводим одну строку таблицы следующим способом: если отношение R задано над атрибутами $A_{i_1}, A_{i_2}, \ldots, A_{i_s}$, то тогда в столбцы, соответствующие этим атрибутам, ставим c_1 для A_{i_1}, c_2 для $A_{i_2}, \ldots$, и c_s для A_{i_s}. В столбцы атрибутов, которые не участвуют в отношении R, ставим пробелы. Из пункта а) определения 2 видно, что таким образом в строке могут участвовать свободные переменные, связанные переменные, константы и пробелы.

Каждой строке ставим маркер (R) с правой стороны таблицы, если строка отведена терму $R(c_1, c_2, \ldots, c_s)$; таким образом отмечается "откуда" берется эта строка.

Видно, что при этом построении резюме и строк таблицы переменные участвуют только в столбцах атрибутов, к которым они относятся - т.е. одна переменная не может фигурировать одновременно в двух разных столбцах. Кроме того требуется, чтобы свободная переменная не появлялась в строках таблицы, если она не фигурирует в ее резюме.

Пункт б) определения 2 показывает, что в выражении (1) может участвовать и терм вида $c\Theta d$. Каждый терм вида $c\Theta d$

записывается под строками таблицы. Таким образом формируется список ограничений, который тоже рассматривается как часть таблицы.

Пример 2.

Для выражения (2) над базой данных из примера 1

$\{a_1 \mid (\exists b_1)(\exists b_2)$ такие, что ПОСТАВЩИК $(a_1, b_1, b_2, c_1)\}$

получаем таблицу

(4)

ПИМЯ	ПНОМЕР	ПАДРЕС	ПГОРОД	
a_1				
a_1	b_1	b_2	c_1	(ПОСТАВЩИК)

Резюме этой таблицы содержит свободную переменную a_1. Кроме резюме таблица содержит и строку $<a_1\ b_1\ b_2\ c_1>$,соответствующую терму ПОСТАВЩИК (a_1, b_1, b_2, c_1) из выражения (2). Поэтому строка отмечена маркером (ПОСТАВЩИК), поставленном с правой стороны таблицы.

Результатом таблицы (а также результатом конъюнктивного запроса) является отношение. Это отношение-результат над атрибутами, содержащими свободные переменные в резюме данной таблицы.

Пример 3.

Для таблицы (4) отношением-результатом является

$r_1 = \{<a_1> \mid a_1 \in$ ПИМЯ и существуют значения b_1 атрибута ПНОМЕР и b_2 атрибута ПАДРЕС такими, что кортеж $a_1 b_1 b_2 c_1$ принадлежит отношению ПОСТАВЩИК$\}$.

Здесь мы будем интерпретировать отношение r_1 как "результат" таблицы (4).

<u>Пример 4.</u>

Построим таблицу для конъюнктивного запроса q_2, используя выражения (3):

ЧИМЯ	ЧНОМЕР	ЦЕНА	ПИМЯ	ПНОМЕР	ПАДРЕС	ПГОРОД	КИМЯ	КНОМЕР	КАДРЕС	КГОРОД	КОЛИЧЕСТВО	
			a_1									
			a_1	b_1	b_2	b_3						(ПОСТАВЩИК)
							b_4	b_5	b_6	c_2		(КЛИЕНТ)
c_1	b_7	b_8										(ЧАСТЬ)
	b_7			b_1				b_5			b_9	(ПОСТАВКА)

Для этой таблицы отношением-результатом является

$r_2 = \{\langle a_1 \rangle \mid a_1 \in$ ПИМЯ и существуют $b_1 \in$ ПНОМЕР, $b_2 \in$ ПАДРЕС, $b_3 \in$ ПГОРОД, $b_4 \in$ КИМЯ, $b_5 \in$ КНОМЕР, $b_6 \in$ КАДРЕС, $b_7 \in$ ЧНОМЕР, $b_8 \in$ ЦЕНА, $b_9 \in$ КОЛИЧЕСТВО такие, что

$\langle a_1 b_1 b_2 b_3 \rangle \in$ ПОСТАВЩИК

$\wedge \langle b_4 b_5 b_6 c_2 \rangle \in$ КЛИЕНТ

$\wedge \langle c_1 b_7 b_8 \rangle \in$ ЧАСТЬ

$\wedge \langle b_7 b_1 b_5 b_9 \rangle \in$ ПОСТАВКА$\}$.

Пример 5.

Рассмотрим другую примерную таблицу T_1

T_1:

A_1	A_2	A_3	A_4	A_5	
	a_1		a_2		
b_1	a_1	b_2			(R_1)
		b_2	a_2	b_3	(R_2)
	b_4		b_5	c_1	(R_3)

$$b_3 < c_1$$

Строки этой таблицы показывают, что

$R_1 = \{A_1, A_2, A_3\}$, $R_2 = \{A_3, A_4, A_5\}$ и $R_3 = \{A_2, A_4, A_5\}$.

Отношение-результат можно записать следующим образом:

$R(T_1) = \{\langle a_1 a_2 \rangle \mid a_1 \in A_2,\ a_2 \in A_4$ и существуют $b_1 \in A_1$,
$b_2 \in A_3,\ b_3 \in A_5,\ b_4 \in A_2,\ b_5 \in A_4$
такие, что $\langle b_1 a_1 b_2 \rangle \in R_1$ и $\langle b_2 a_2 b_3 \rangle \in R_2$
и $\langle b_4 b_5 c_1 \rangle \in R_3$ и $b_3 < c_1\}$.

Легко представить запись конъюнктивного запроса, для которого составлена таблица T_1:

$$\{a_1 a_2 \mid (\exists b_1)(\exists b_2)(\exists b_3)(\exists b_4)(\exists b_5) \text{ такие, что } R_1(b_1 a_1 b_2) \wedge R_2(b_2 a_2 b_3) \wedge R_3(b_4 b_5 c_1) \wedge b_3 < c_1\}.$$

Задавая более сложные запросы, мы часто представляем в виде столбца таблицы все атрибуты, участвующие в реляционных схемах конкретной базы данных. Так как таблица используется и для описания конъюнктивных запросов в универсальных реляционных системах, для таких таблиц приходится перечислять столбцы всех атрибутов универсального отношения. В связи с этим нужно отметить некоторые особенности процесса отождествления разных атрибутов как один столбец данной таблицы.

Рассмотрим следующий запрос к базе данных из примера 1:

q_3 : Найти имена всех поставщиков и имена всех клиентов, живущих в одном и том же городе.

Конъюнктивное представление запроса:

(5) $$\{a_1 a_2 \mid (\exists b_1)(\exists b_2)(\exists b_3)(\exists b_4)(\exists b_5)(\exists b_6) \text{ такие, что } \text{ПОСТАВЩИК } (a_1 b_1 b_2 b_3) \wedge \text{КЛИЕНТ } (a_2 b_4 b_5 b_6) \wedge (b_3 = b_6)\}.$$

Соответствующая таблица имеет вид:

	ПИМЯ	ПНОМЕР	ПАДРЕС	ПГОРОД	КИМЯ	КНОМЕР	КАДРЕС	КГОРОД	
T_2:	a_1				a_2				
	a_1	b_1	b_2	b_3					(ПОСТАВЩИК)
					a_2	b_4	b_5	b_6	(КЛИЕНТ)
								$b_3 = b_6$	

В этом примере атрибуты ПАДРЕС и ПГОРОД изменяются в тех же доменах, в которых соответственно изменяются КАДРЕС и КГОРОД. Представляется очень заманчивым объединить их в виде двух столбцов таблицы с именами например АДРЕС и ГОРОД.

Тогда для q_3 мы получили таблицу:

T_2' :

ПИМЯ	ПНОМЕР	АДРЕС	ГОРОД	КИМЯ	КНОМЕР	
a_1				a_2		
a_1	b_1	b_2	b_3			(ПОСТАВЩИК)
		b_4	b_3	a_2	b_5	(КЛИЕНТ)

В случае допущения такого отождествления атрибутов в столбцах таблицы могут возникнуть проблемы в процессе построения таблицы для запроса q_4:

q_4 : Найти адреса всех поставщиков и всех клиентов, живущих в одном и том же городе.

Его конъюнктивная запись имеет вид:

(6) $\{a_1 a_2 \mid (\exists\, b_1)(\exists\, b_2)(\exists\, b_3)(\exists\, b_4)(\exists\, b_5)(\exists\, b_6)$ такие, что

ПОСТАВЩИК $(b_1 b_2 a_1 b_3) \wedge$ КЛИЕНТ $(b_4 b_5 a_2 b_6) \wedge$

$\wedge\ (b_3 = b_6)\}$.

В этом случае невозможно записать (6) в виде таблицы со столбцами, как таблицу T_2' , так как мы нуждаемся в двух свободных переменных, которые нужно внести в столбец АДРЕС (что согласно определения понятия таблицы не является возможным). Нам необходима таблица, столбцы которой должны выглядеть как столбцы таблицы T_2.

Следовательно можно заключить, что при отождествлении атрибутов и столбцов таблиц нужно соблюдать так называемое предположение о единственной роли (unique role assumption [3]). В этом случае можем быть уверены, что данное выше определение таблицы позволит нам сопоставлять каждому конъюнктивному запросу соответствующую ему таблицу.

3. ПОСТРОЕНИЕ ТАБЛИЦ ПО ДАННЫМ SPJ-ВЫРАЖЕНИЯМ

Определение 3 показывает построение таблицы по данному конъюнктивному запросу. Рассмотрим алгоритм построения таблицы для данного SPJ-выражения.

Таблица данного SPJ-выражения содержит столбцы атрибутов для всех реляционных схем, участвующих в данном SPJ-выражении.

Значение каждого SPJ-выражения является отношением и его можно рассматривать в качестве ответа некоторого конъюнктивного запроса. Следовательно для данного SPJ-выражения мы можем построить таблицу, представляющую собой запрос, ответ которого данное SPJ-выражение. Так как SPJ-выражение является формулой и его можно строить индуктивным образом, то таблицу SPJ- выражения также можно строить индуктивным образом.

При построении таблицы для данного SPJ-выражения выполняется индукция по отношению к числу операций Π, σ, J, которые содержатся в данном SPJ-выражении.

Пусть E является SPJ-выражением, которое содержит 0 операций Π, σ, J. Тогда E = R, где R реляционная схема $R = \{A_{i_1}, A_{i_2}, \ldots, A_{i_s}\}$ для некоторого множества атрибутов $A_{i_1}, A_{i_2}, \ldots, A_{i_s}$. Тогда таблица состоит из резюме и еще одной строки. В столбцах, соответствующих именам атрибутов схемы R, в резюме находятся свободные переменные. Другие столбцы в резюме пустые. В строке в столбцах, соответствующих именам атрибутов схемы R, находятся те же самые свободные переменные, которые находятся и в резюме; другие столбцы этой строки заполнены разными связанными переменными.

Допустим, что возможно построить таблицу для данного SPJ-выражения, которое содержит n операций Π, σ, J. Будем строить таблицу для SPJ-выражений, которое содержит n+1 операций Π, σ, J.

Пусть E является SPJ-выражением, которое содержит n+1 операций Π, σ, J. Тогда имеет место одна из следующих трех возможностей:

a) $E = \Pi_X(E_1)$, где $X \subseteq \{A_1, A_2, \ldots, A_k\}$ и

E_1 является SPJ-выражением, которое содержит не больше чем n операций Π, σ, J;

б) $E = \sigma_{A_i=c}(E_1)$, где $A_i \in \{A_1, A_2, \ldots, A_k\}$ и E_1 является SPJ-выражением, которое содержит не больше чем n операций Π, σ, J;

в) $E = E_1 \bowtie E_2$, где E_1 и E_2 являются SPJ-выражениями, которые содержат не больше чем n операций Π, σ. J.

Покажем как строится таблица для каждого из этих случаев:

а) Предположим, что $E = \Pi_X(E_1)$, причем T_1 - таблица для E_1. Таблицу T для E строим из таблицы T_1 для E_1 следующим образом: в резюме T_1 ставим "пустые символы" в столбцы, непринадлежащие X. Во всех остальных строках для этих столбцов свободные переменные заменяются разными связанными переменными.

б) Предположим, что $E = \sigma_{A_i=c}(E_1)$. Тогда, если T_1 - таблица для E_1, таблицу T для E можно получить из T_1 следующим способом:

- если столбец A_i в резюме T_1 - пустой, то выражение E не имеет смысла и таблица T - неопределена;
- если в столбце A_i в резюме T_1 имеется константа c_1, то таблица T совпадает с T_1, если $c_1 = c$. В противном случае резюме таблицы T не содержит свободные переменные и таблица T не содержит ни-

какие строки. В таком случае будем обозначать таблицу T как пустое множество;

- если в столбце A_i в резюме T_1 имеется свободная переменная a, то таблица T получается от таблицы T_1 путем замещения a через c, независимо от того, в каком месте встречается **a** в T_1.

в) Предположим, что $E = E_1 \bowtie E_2$ и T_1 и T_2 - таблицы для E_1 и E_2 соответственно. Без потери общности можно предположить, что множества связанных переменных T_1 и T_2 не пересекаются и что если в одних и тех же столбцах в строках резюме T_1 и T_2 фигурируют свободные переменные, то они являются одинаковыми. Таблицу T для E конструируем следующим способом: если в резюме T_1 и T_2 на одном и том же месте фигурируют разные константы, то $T = \emptyset$. В противном случае строками T являются строки T_1 и T_2, причем резюме T образовано из резюме T_1 и T_2 как следует. Если в данном столбце A_i фигурируют:

- константа c в резюме одной из таблиц T_1 и T_2, то в резюме T ставится c и везде свободная переменная другой таблицы заменяется константой c;
- свободная переменная в одной из таблиц или в обеих, то в резюме T ставится та же переменная;
- пустые символы в обеих таблицах T_1 и T_2, то в таблицу T ставим так же пустой символ.

Пример 5.

Проиллюстрируем процесс конструирования таблицы для SPJ-выражения на следующем примере:

q_5 : Найти имена поставщиков, поставляющие части ценой 50.

Ответ запроса можно представить при помощи SPJ-выражения:

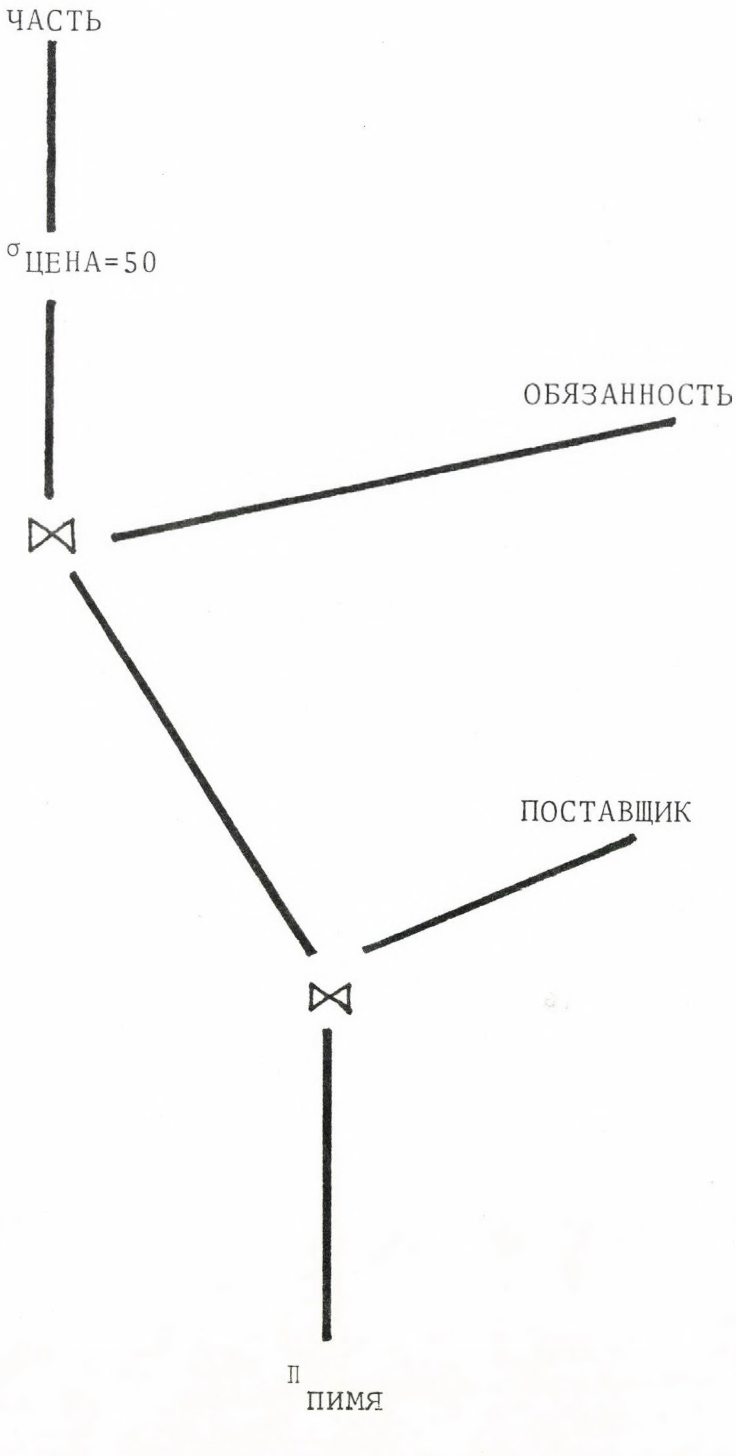
(6) $\Pi_{ПИМЯ}((ОБЯЗАННОСТЬ \bowtie \sigma_{ЦЕНА=50}(ЧАСТЬ)) \bowtie ПОСТАВЩИК)$

Дерево разбора [7] этого выражения дано на фиг.1.

На фиг.2 представляется последовательность конструирования таблицы для SPJ-выражения (6). Во всех таблицах записаны атрибуты, участвующие в реляционных схемах ОБЯЗАННОСТЬ, ЧАСТЬ и ПОСТАВЩИК.

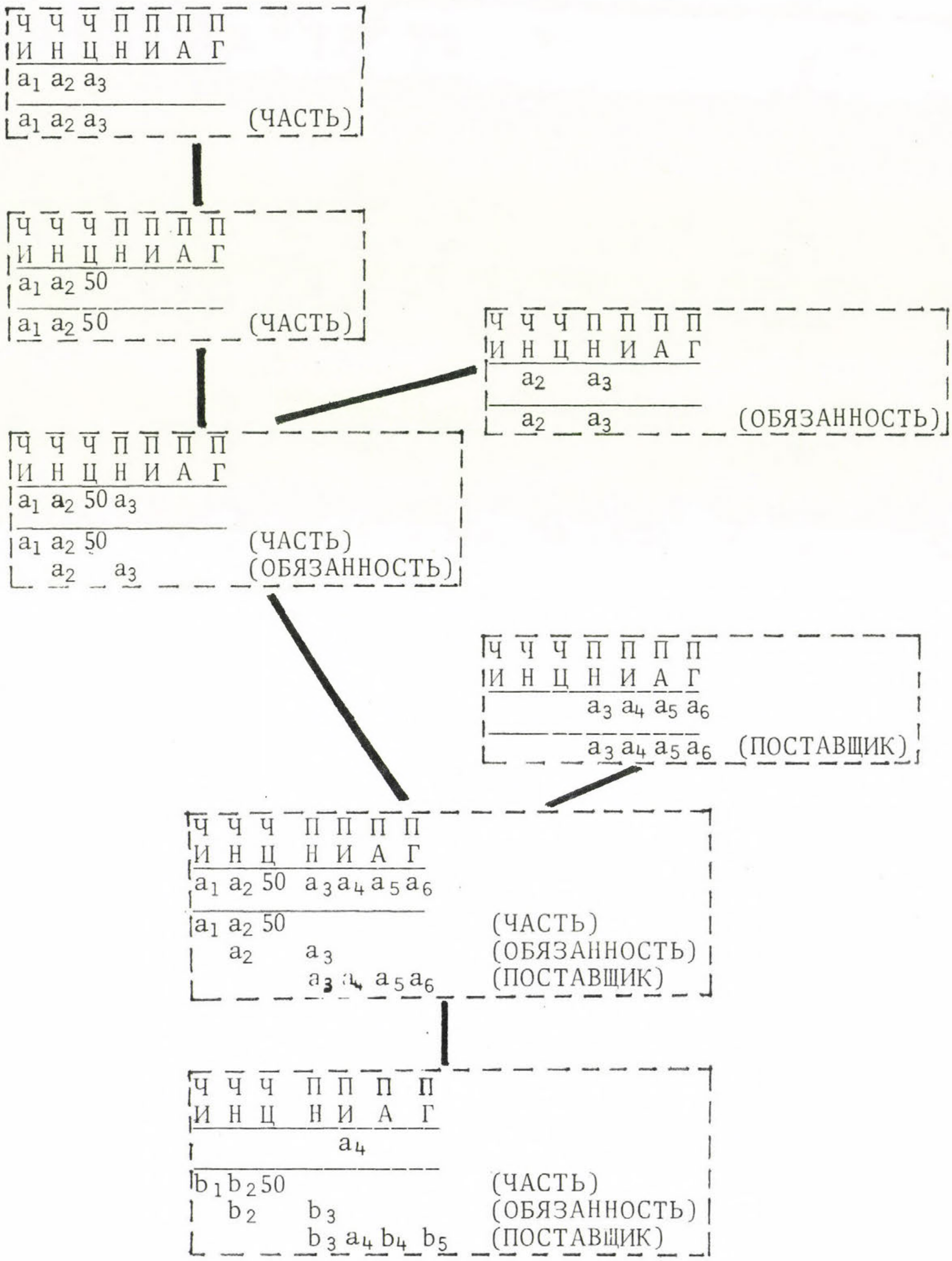
В таблицах на фиг.2 атрибуты обозначены только через две буквы (например, ЧИМЯ обозначено через ЧИ - $\genfrac{}{}{0pt}{}{Ч}{И}$), а все связанные переменные, встречающиеся только один раз, пропущены (т.е. замещены пустыми символами).

Расположение отдельных таблиц на фиг.2 соответствует расположению элементов в дереве разбора на фиг.1. Различные таблицы отделены одна от другой пунктирной линией.



Фиг.1.

Дерево разбора для SPJ - выражения (6).



Фиг.2.

Конструирование таблицы для SPJ - выражения (6).

Определение 6.

Пусть T_1 и T_2 - таблицы. Отображение h символов T_1 на символы T_2 называется "содержащим отображением", если:

а) h отображает символы резюме T_1 в символы резюме T_2;

б) h отображает символы любой строки T_1 в символы строки T_2 с таким же маркером, как у строки из T_1 . При этом h сохраняет значение всех констант;

в) h отобажает список ограничений T_1 в множество ограничений, являющееся подмножеством ограничений T_2.

Если h отображает все символы строки в символы другой строки, говорим, что h отображает всю данную строку в другую.

Таким образом будем рассматривать h и как отображение одного символа в другой, и как отображение одной строки в другую.

Теорема 1. [7] $T_1 \supseteq T_2$ тогда и только тогда, когда существует "содержащее отображение" h из T_1 на T_2.

Следовательно $T_1 \equiv T_2$ тогда и только тогда, когда существует "содержащее отображение" h_1 из T_1 на T_2 и h_2 из T_2 на T_1.

Определение 7.

Пусть Т - таблица. Минимальной таблицей для таблицы Т будем называть таблицу, содержащую минимальное число строк и эквивалентной таблице Т. Процесс нахождения минимальной

4. ЭКВИВАЛЕНТНОСТЬ И МИНИМИЗАЦИЯ ТАБЛИЦ

Как указано выше, понятие таблицы вводится с целью исследовать эквивалентность конъюнктивных запросов, полагая, что таким образом запрос легче поддается формализации. Как следует ожидать, два запроса являются эквивалентными тогда и только тогда, когда их таблицы эквивалентны.

Использование понятия таблицы основывается на следующих определениях.

Пусть $d = \{r_1, r_2, \ldots, r_n\}$ представляет собой состояние базы данных, т.е. множество отношений над реляционными схемами $\{R_1, R_2, \ldots, R_n\}$. Тогда отношение-результат, сопоставленное данной таблице T при помощи вышеописанной интерпретации, будем означать через T(d), (естественно, значение этого отношения является различным для различных состояний d).

Определение 4.

Будем говорить, что $T_1 \subseteq T_2$, если для каждого d в силе $T_1(d) \subseteq T_2(d)$.

Определение 5.

T_1 и T_2 являются эквивалентными ($T_1 \equiv T_2$) тогда и только тогда, когда $T_1 \subseteq T_2$ и $T_2 \supseteq T_1$.

В основе алгоритма для определения эквивалентности двух данных таблиц [1] и [7], лежит понятие "отображения" между символами и строками таблиц.

таблицы для данной таблицы Т будем называть оптимизацией таблицы Т.

В рассматриваемых до сих пор таблицах маркеры показывают из какого отношения берется данная строка. Определенная выше эквивалентность, где в пункте б) определения 6 требуется сохранить маркер строки при отображении одной таблицы в другую, называется сильной эквивалентностью. Если мы поставим себе задачу оптимизировать число J-операций в процессе реализации данного конъюнктивного запроса, то пользуясь техникой таблиц, можем найти таблицу, эквивалентную данной, содержащую минимальное число строк (эта задача является NP - полной [7]). Эта минимальная таблица,сильно эквивалентная данной таблице, должна содержать строки с такими же маркерами, как и выходная таблица. Такая минимальная таблица предполагает, что при обработке начального конъюнктивного запроса будут применяться J-операций ко всем отношениям, упомянутым в первоначальной формулировке запроса. Однако это не всегда является необходимым. Требование рассматривать строки таблиц вместе с их маркерами не в силе, если предположить существование универсального отношения I над атрибутами

$$U = R_1 \cup R_2 \cup \ldots \cup R_n \ ,$$

такого, что $r_i = \Pi_{R_i}(I)$ где $1 \leq i \leq n$. Это предположение известно под именем "предположение существования универсума" (universal instance assumption [3]). В таком случае каждая строка таблиц для конъюнктивных запросов снабжена маркером U, т.е. каждая строка берется из универсума. Таким образом не нужно учитывать от куда взялась каждая строка и маркеры могут быть пропущены. Тогда при оптимизации таблицы возможно исчезновение всех строк с маркерами R_s для некоторого отношения r_s, которые присутствовали в первоначальном представлении таблицы. Предположение о существовании универсума ведет к определению понятия слабой эквивалентности.

Определение 8.

Пусть E_1 и E_2 - две выражения над данным состоянием d и пусть I - универсальное отношение для d. E_1 и E_2 являются слабо эквивалентными (записываем $E_1 \equiv_w E_2$), если $E_1(I) \equiv E_2(I)$ длякаждого возможного состояния I.

Аналогичное определение вводим и для таблиц.

Определение 9.

Пусть T_1 и T_2 - таблицы соответственно для конъюнктивных запросов E_1 и E_2. T_1 и T_2 являются слабо эквивалентными (записываем $T_1 \equiv_w T_2$) тогда и только тогда, когда $E_1 \equiv_w E_2$.

Пример 6.

Понятия "сильная эквивалентность" и "слабая эквивалентность" будут проиллюстрированы на следуюших выражениях:

(7) $\quad \pi_{AB}(AB \bowtie BC)$ и

(8) $\quad AB$

Можно показать, что выражения (7) и (8) - слабо эквивалентны.

Пусть $U = \{A, B, C\}$ - универсум над атрибутами A, B, C и пусть I - конкретное отношение этого универсума. Тогда

$r(AB) = \{ab \mid (\exists\, c)$ такое, что $abc \in I\}$;

$r(BC) = \{bc \mid (\exists\, a)$ такое, что $abc \in I\}$;

$r(AB) \bowtie r(BC) = \{abc \mid (\exists\, c')(\exists\, a')$ такие, что $abc' \in I \wedge a'bc \in I\}$.

Тогда

(9) $\Pi_{AB}(r(AB) \bowtie r(BC)) =$

$\{ab \mid (\exists\, c)(\exists\, c')(\exists\, a')$ такие, что $abc' \in I \wedge a'bc \in I\}$.

Выражение (8) можно записать следующим образом:

(10) $\{ab \mid (\exists\, c)$ такое, что $abc \in I\}$.

Полагая в (9) $a = a'$ и $c = c'$ получаем, что выражения (9) и (10) - эквивалентны и следовательно (7) и (8) - слабо эквивалентны.

Покажем, что выражения (7) и (8) не являются сильно эквивалентными.

Пусть

A	B
a_1	b_1
a_2	b_2

B	C
b_1	c_1
b_1	c_2
b_3	c_3

Тогда для выражения $AB \bowtie BC$ имеем

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2

Следовательно для выражения $\Pi_{AB}(AB \bowtie BC)$ имеем

A	B
a_1	b_1

.

Видно, что на этом примере $\Pi_{AB}(AB \bowtie BC) \neq AB$. Следовательно, выражения (7) и (8) - слабо эквивалентны, но не сильно эквивалентны.

В [7] показано, что необходимое и достаточное условие слабой эквивалентности двух таблиц T_1 и T_2 это существование "содержащего отображения" h_1 из T_1 на T_2 и h_2 из T_2 на T_1. Как мы уже отметили, в этом случае отображения h_1 и h_2 не будут учитывать маркеры разных строк таблиц (так как U является маркером всех строк).

<u>Пример 7.</u>

Рассмотрим таблицы T_3 и T_4:

T_3:

ПИМЯ	ПНОМЕР	ПАДРЕС	ПГОРОД	ЧНОМЕР	КНОМЕР	КОЛИЧЕСТВО	
a_1							
a_1	b_1						(ПОСТАВЩИК)
	b_1			b_2			(ОБЯЗАННОСТЬ)
	b_1			b_2		500	(ПОСТАВКА)

T_4:

ПИМЯ	ПНОМЕР	ПАДРЕС	ПГОРОД	ЧНОМЕР	КНОМЕР	КОЛИЧЕСТВО	
a_1							
a_1	b_1						(ПОСТАВЩИК)
	b_1					500	(ПОСТАВКА)

Сразу видно, что таблицы T_3 и T_4 не сильно эквивалентны, потому что T_4 не содержит строку с маркером (ОБЯЗАННОСТЬ). Но если предположим существование универсального отношения, тогда не нужно учитывать маркеры в таблицах T_3 и T_4 и эти таблицы слабо эквивалентны, так как существуют отображение h_1 символов строк T_3 в символы строк T_4 и отображение h_2 символов строк T_4 в символы строк T_3. h_1 отображает первую строку T_3 в первую строку T_4, вторую строку T_3в первую строку T_4 и последную строку T_3 во вторую строку T_4; h_2 отображает первую строку T_4 в первую строку T_3 и вторую строку T_4 в последную строку T_3.

5. ИСПОЛЬЗОВАНИЕ ТАБЛИЦ КОНЪЮНКТИВНЫХ ЗАПРОСОВ

Так как техника таблицы дает нам возможность устанавливать эквивалентность двух конъюнктивных запросов, она может быть применена для нахождения оптимального представления запроса относительно данного критерия. Таким критерием является: вычислить значение данного SPJ-выражения, используя минимальное число J-операций (реализация J-операции достаточно тяжела).

Из алгоритма построения таблицы для данного SPJ-выражения видно, что операция J порождается парой строк в таблице. Следовательно для данной таблицы n строками можно конструировать реализацию соответствующему запросу при помощи n-1 J-операций. При таком критерии оптимальности проблема оптимизирования данного запроса сводится к нахождению таблицы, слабо эквивалентной данной таблице.

Пример 8.

Предположим, что для базы данных из примера 1 выполнено предположение о существовании универсума. Ищем ответ для следующего запроса:

q_6 : Найти имена всех поставщиков, поставляющие (или уже поставшие) части, количество которых 500.

SPJ-выражение, реализующее ответ:

(11) $\pi_{\text{ПИМЯ}}(\text{ПОСТАВЩИК} \bowtie \text{ОБЯЗАННОСТЬ} \bowtie \sigma_{\text{КОЛИЧЕСТВО}=500}(\text{ПОСТАВКА}))$.

Соответствующая этому запросу таблица T_3.(При описании таблицы T_3 пропущены столбцы атрибутов, не участвующие в описании).

Сразу видно, что T_4 является оптимальной слабо эквивалентной таблицей для таблицы T_3.

Таблица T_4 представляет выражение

(12) $\pi_{\text{ПИМЯ}}(\text{ПОСТАВЩИК} \bowtie \sigma_{\text{КОЛИЧЕСТВО}=500}(\text{ПОСТАВКА}))$.

Таким образом ясно, что выражения (11) и (12) являются слабо эквивалентными.

Нужно отметить, что здесь допущение существования универсума является существенным. (Таблицы T_3 и T_4 не являются сильно эквивалентными).

Пример 8 иллюстрирует и применение понятия таблицы в System/U [6] и [7]. Так как в System/U основное предположение - предположение о существовании универсума, каждый конъюнктивный запрос пользователя представлен с помощью своей оптимальной таблицы (являющейся слабо эквивалентной данной таблице)и таким образом осуществляется более эффективная реализация запроса.

Другие применения техники таблиц даны например в [2], [4] и [5].

6. ЗАКЛЮЧЕНИЕ

В настоящей работе описаны основные характеристики понятия таблицы и основные возможности его применения. Сейчас таблицы рассматриваются как средства для изучения конъюнктивных запросов. Так как таблица дает синтезированное описание содержания некоторого отношения, ее можно применять и для изучения связей между разными типами зависимостей в рамках этого отношения.

ЛИТЕРАТУРА

1. Aho, A., Sagiv, Y., and Ullman, J. Efficient Optimization of a class of Relational Expressions. ACM TODS, Vol. 4, No.4, December 1979, 435-454.

2. Maier D. The Theory of Relational Databases. Computer Science Press, 1983.

3. Maier, D., Ullman, J. and Vardi, M. On the Foundations of the Universal Relation Model. ACM TODS, Vol.9, No. 2, June 1984, 283-308.

4. Mendelzon, A. Database States and Their Tableaux. ACM TODS Vol. 9, No. 2, June 1984, 264-282.

5. Klug, A. and Price, R. Determining View Dependencies Using Tableaux. ACM TODS, Vol. 7, No. 3, September 1983, 361-380.

6. Korth, N., Kuper, G., Feigenbaum, J., Val Gelder, A. and Ullman, J. System/U: a Database System based on the Universal Relation Assumption. ACM TODS, Vol. 9, No. 3, September 1984, 331-347.

7. Ullman, J. Principles of Database Systems. Computer Science Press, 1982.

Conjunctive queries tableaux and their application

G. Angelova

Summary

The paper discusses the concept of tableau for conjunctive queries for a relational data base. The basic definition of tableau is presented. The algorithm building tableaux for given select-project-join relational expressions is discussed. The problem of tableaux equivalence and optimization is considered. Using tableaux, a necessary and sufficient condition for equivalence of select-project-join relational expressions is given. Some applications of tableaux are presented.

Konjuktiv lekérdezési táblák és alkalmazásaik

G. Angelova

Összefoglaló

A cikk a relációs adatbázisokkal kapcsolatos konjuktiv lekérdezési tábla fogalmát tárgyalja. Megadja a tábla definicióját. Egy algoritmust ad meg, amely egy adott kiválasztás-projekció-összekapcsolás relációs kifejezés számára épiti fel a táblát. A táblák ekvivalenciájának és optimalizálásának kérdéseivel is foglalkozik. A k-p-ö relációs kifejezések ekvivalenciájával is foglalkozik, felhasználva a táblákat. A táblák alkalmazásáról is szó van.

MTA SZTAKI Közlemények 34/1986 pp.37-51

SOME ADDITIONAL PROPERTIES OF KEYS FOR RELATION SCHEME

J. DEMETROVICS and HO THUAN

MTA SZTAKI

ABSTRACT

In this paper we prove some additional properties of keys and superkeys for relation schemes. Some of them and their variants have been proved (perhaps by different methods) and used to design an algorithm to find all keys for any relation scheme [3].

INTRODUCTION

In [1] some characteristic properties of keys for a given relation scheme $S = \langle\Omega, F\rangle$ have been investigated, in particular the necessary condition under which a subset X of Ω is a key.

In this paper we prove some additional properties of keys and super keys for relation scheme. Some of them and their variants have been proved (perhaps by different methods) and used to design an algorithm to find all keys for a relation scheme [3].

The notation used here is the same as in [1] and [2].

1. §

In this section we recall some notions and results which will be needed in the sequel.

Let $S = \langle\Omega,F\rangle$ be a relation scheme, where

$$\Omega = \{A_1,A_2,\ldots,A_n\}$$

$$F = \{L_i \to R_i \mid i = 1,2,\ldots,k;\ L_i,\ R_i \subseteq \Omega\}$$

Let us denote:

$$L = \bigcup_{i=1}^{k} L_i,\quad R = \bigcup_{i=1}^{k} R_i$$

$$C_i = \Omega\setminus L_i^+,\ i = 1,2,\ldots,k$$

$I = \{i \mid \text{there is no } j \text{ such that } L_i \supset L_j\} \subseteq \{1,2,\ldots,k\}$
Recall that for $X \subseteq \Omega$,

$$X^+ = \{A \mid (X \to A) \in F^+\}$$

is the closure of X w.r.t. F.

Without loss of generality, in this paper we assume that

$$L_i \cap R_i = \emptyset,\quad i = 1,2,\ldots,k$$

$$L_i \neq L_j \quad \text{with} \quad i \neq j$$

$$L \cup R = \Omega$$

We have the following lemmas.

Lemma 1 [2]. Let $S = \langle\Omega,F\rangle$ be a relation scheme, $X,\ Y \subseteq \Omega$, then

$$(X^+ Y)^+ = (X Y^+)^+ = (X Y)^+$$

Lemma 2 [2]. Let K be a key for $S = \langle\Omega,F\rangle$. Then

$$Z^+ \cap (K \setminus Z) = \emptyset \quad \text{for all} \quad Z \subseteq K.$$

Lemma 3 [2]. For any $i \in I$,

$$L_i \text{ is a key for } S \text{ iff } C_i = \emptyset$$

2. §

We are now in a position to prove some properties of keys and superkeys for relation scheme which can be used for the design of algorithms to find the keys of a relation scheme.

Theorem (key representation).

Let $S = \langle\Omega,F\rangle$ be a relation scheme.

Then any key K for S has the following form

$$K = L_i X_i$$

where $X_i \subseteq C_i$, $i \in I$.

Proof. Let be given $K \in K_S$ - the set of all keys for S. If $K = \Omega$, then obviously $K = L_i X_i$ $\forall i \in I$. If $K \subset \Omega$, then by the algorithm to find the closure K^+ of K w.r.t. F, there exists L_j such that $L_j \subseteq K$. [see 4]

Consequently, $\exists i \in I$ such that $L_i \subseteq K$.

Thus $K = L_i X_i$, $i \in I$.

Now we have to prove that $X_i \subseteq C_i$.

By lemma 1, we have

$$L_i^+ X_i \subseteq (L_i^+ X)^+ = (L_i X_i)^+ = K^+ = \Omega = L_i^+ C_i \qquad (1)$$

By lemma 2:

$$L_i^+ \cap (K \setminus L_i) = L_i^+ \cap X_i = \emptyset.$$

On the other hand, it is clear that

$$L_i^+ \cap C_i = \emptyset .$$

Hence, from (1) we have:

$$X_i \subseteq C_i .$$

The proof is complete.

Remark 1. This theorem can be considered as an immediate corollary of theorem 2.2 [2]. But for our purposes the representation given here is more appropriate.

Lemma 4. Let $S = \langle \Omega, F \rangle$ be a relation scheme. If $C_i \neq \emptyset$ then with $j \neq i$

$$C_i \cap L_j\, C_j \neq \emptyset .$$

Proof. Assume the contrary that

$$C_i \cap L_j\, C_j = \emptyset ,$$

In that case, it is easy to see that /see Fig, 1/

$$C_i \subseteq L_j^+ \setminus L_j$$

and

$$L_j\, C_j \subseteq L_i^+ .$$

Thus we have

$$L^+ \xrightarrow{*} L_j\, C_j \xrightarrow{*} \Omega \quad {}^{*)}$$

---

*) In the following instead of $(X \to Y) \in F^+$, $X \cup Y$ we write $X \xrightarrow{*} Y$ and $X\,Y$ respectively.

This contradicts the hypothesis of the lemma stating that L_i is not a superkey (because $C_i \neq \emptyset$).
The lemma is proved.

Property 1. Let $S = \langle\Omega,F\rangle$ be a relation scheme and suppose that $|C_i| \leq 1 \quad \forall\, i \in I$.
Then for any $i \in I$, $L_i\, C_i$ is a key for S iff there is no $j \in I$ such that $L_i\, C_i \supset L_j\, C_j$.

Proof. If $L_i\, C_i$ is a key for S with $i \in I$, then it is impossible to have a j such that $L_i\, C_i \supset L_j\, C_j$ because $L_j\, C_j$ is itself a superkey for S.

We have only to consider the case $|C_i| = 1$. (Otherwise by lemma 3, L_i is a key).
Suppose that $L_i\, C_i$ contains strictly a key

$$K = L_i'\, C_i \subset L_i\, C_i, \quad L_i' \subset L_i.$$

By the theorem just proved above

$$L_i'\, C_i = L_j\, X_j, \quad j \in I, \quad j \neq i.$$

The case $X_j = \emptyset$, by lemma 3, it is clear that $C_j = \emptyset$ and $L_i\, C_i \supset L_i'\, C_i = L_j\, C_j$ (with $C_j = \emptyset$). The case $X_j \neq \emptyset$, obviously $X_j = C_j$.
But this shows (in both cases) that $L_i\, C_i \supset L_j\, C_j$, a contradiction.

Remark 2. Property 1 is still true if the set I is replaced by the set $\{1,2,\ldots,k\}$

Corollary 1. Let $S = \langle\Omega,F\rangle$ be a relation scheme with $|C_i| \leq 1 \quad \forall\, i \in I$.
If for any $i \in I$, $C_i \cap L_j = \emptyset \quad \forall\, j \neq i$, then $L_i\, C_i$ is a key for S.

Proof. From the conditions of corollary 1, it is evident that there is no $j \in I$ such that $L_i C_i \supset L_j C_j$.
Indeed, were this false and there exists a $j \in I$ such that $L_i C_i \supset L_j C_j$. Since $C_i \notin L_j$ and $L_j \subset L_i C_i$, we have $L_j \subset L_i$, a contradiction.
By property 1, $L_i C_i$ is a key for S.

Property 2. Let $S = \langle \Omega, F \rangle$ be a relation scheme.
For any $i \in \{1,2,\ldots,k\}$,
if $|C_i| = 1$ and $L_i \cap R_j = \emptyset \quad \forall j \neq i$ then $L_i C_i$ is a key for S.

Proof. From $L_i \cap R_j = \emptyset \ \forall j = i$, it follows that

$$L_i \cap R = \emptyset.$$

Thus $L_i \subseteq L \setminus R$.

Moreover [1]

$$L \setminus R \subseteq K \quad \forall K \in K_S.$$

On the other hand $L_i C_i$ is a superkey for S and $L_i^+ \subset \Omega$.
This shows that $L_i C_i$ is a key for S.

Remark 3. Let $S = \langle \Omega, F \rangle$ be a relation scheme.
If $|C_i| = 1$ then C_i is a subset of Ω which consists of a single prime attribute.
Indeed, since $L_i C_i$ is a superkey for S and $L_i^+ \subset \Omega$ (because $C_i \neq \emptyset$), it follows that $L_i C_i$ must contains a key K which has the form

$$K = L_i' C_i \quad \text{where} \quad L_i' \subseteq L_i.$$

This shows that C_i is a prime attribute.

Property 3. Let $S = \langle \Omega, F \rangle$ be a relation scheme. Then $\forall\, i \neq j,\ \ i, j \in \{1,2,\dots,k\}$

$$L_i(C_i \cap L_j\, C_j) \text{ is a superkey for } S.$$

Proof. The case $C_i = \emptyset$, we have

$$L_i(C_i \cap L_j\, C_j) = L_i.$$

But in that case it is obvious that L_i is a superkey.
We now consider the case $C_i \neq \emptyset$.
By Lemma 4 we have $C_i \cap L_j\, C_j \neq \emptyset$.
It is clear that

$$L_i \overset{*}{\rightarrow} L_i^+$$

$$C_i \cap L_j\, C_j \overset{*}{\rightarrow} C_i \cap L_j\, C_j\ .$$

Consequently,

$$L_i(C_i \cap L_j\, C_j) \overset{*}{\rightarrow} L_i^+(C_i \cap L_j)(C_i \cap C_j).$$

On the other hand we have:

$$L_j = (L_j \setminus C_i)(C_i \cap L_j) \subseteq L_i^+(C_i \cap L_j),$$

$$C_j = (C_j \setminus C_i)(C_j \cap C_i) \subseteq L_i^+(C_i \cap C_j).$$

Hence

$$L_j\, C_j \subseteq L_i^+(C_i \cap L_j)(C_i \cap C_j). \quad \text{(see Fig, 2)}$$

Finally we have

$$L_i(C_i \cap L_j\, C_j) \overset{*}{\rightarrow} L_j\, C_j$$

showing that $L_i(C_i \cap L_j\ C_j)$ is a superkey for S.

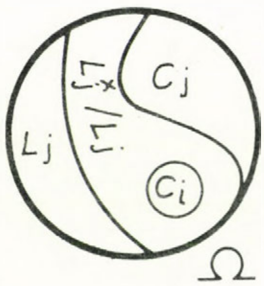


Fig. 1

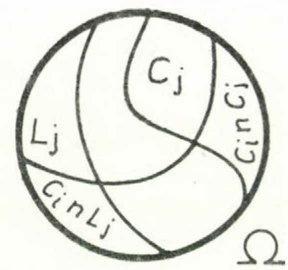


Fig. 2

<u>*Remark 4.*</u> The case $|C_i| = 1$ we have:

$$L_i(C_i \cap L_j\ C_j) = L_i\ C_i, \quad j \neq i.$$

<u>*Remark 5.*</u> Let $L_i\ X$ and $L_j\ Y$ be superkey for S, $i \neq j$. In general $L_i(X \cap L_j\ Y)$ is not a superkey.
Let us consider the relation scheme;

$$\Omega = 1\ 2\ 3\ 4\ 5\ 6\ 7\ ,$$

$$F = \{12 \to 36, \quad 64 \to 52, \quad 23 \to 17\}$$

It is easy to verify that 641 and 234 are superkeys.

$$(641)^+ = 6415237 = \Omega$$

$$(234)^+ = 2341765 = \Omega\ .$$

On the other hand we have:

$$64(1 \cap 234) = 64,$$

and

$$(64)^+ = 6452 \neq \Omega.$$

Property 4. Let $S = \langle\Omega, F\rangle$ be a relation scheme. If $C_i \neq \emptyset$, $C_j \neq \emptyset$ then

$T = L_i[(C_i \cap L_j C_j) \cap L_j(C_j \cap L_h C_h)]$ is a superkey for S,

where $i \neq j$, $j \neq h$.

Proof. By property 3, it is clear that $L_j(C_j \cap L_h C_h)$ is a superkey for S.
Let us denote

$$Y = L_i^+[(C_i \cap L_j C_j) \cap L_j(C_j \cap L_h C_h)] =$$

$$= L_i^+(C_i \cap L_j C_j) \cap L_i^+ L_j(C_j \cap L_h C_h);$$

Obviously $T \overset{*}{\rightarrow} Y$ and

$$L_i^+ L_j(C_j \cap L_h C_h) \supseteq L_j(C_j \cap L_h C_h).$$

On the other hand,

$L_j C_j \subseteq L_i^+(C_i \cap L_j C_j)$ (from the proof of property 3)

It follows that

$$L_i^+(C_i \cap L_j C_j) \supseteq L_j(C_j \cap L_h C_h).$$

Hence

$$T \overset{*}{\rightarrow} Y \overset{*}{\rightarrow} L_j(C_j \cap L_h C_h)$$

showing that T is a superkey for S.

Corollary 2. With the same conditions as in Property 4, we have:

$$(C_i \cap L_j C_j) \cap L_j(C_j \cap L_h C_h) \neq \emptyset .$$

Property 5. Let K be any key for $S = <\Omega, F>$ having the form $K = L_i X$, $(X \subset C_i)$.
Then there exists $j_o \neq i$ such that

$$K \subseteq L_i (C_i \cap L_{jo} C_{jo}).$$

Proof. Assume the contrary that

$$L_i X \not\subseteq L_i (C_i \cap L_j C_j) \quad \forall j \neq i,$$

or, equivalently

$$X \not\subseteq C_i \cap L_j C_j \qquad \forall j \neq i.$$

Then, for all $j \neq i$ there exists an attribute

$$A_{i_j} \in (L_j^+ \setminus L_j) \cap X \qquad \text{(see Fig. 3)}$$

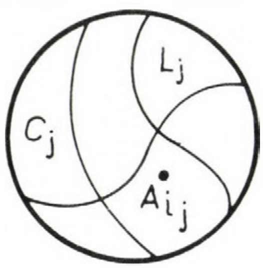


Fig. 3

Obviously we have:

$$L_i X \overset{*}{\rightarrow} L_i R_i X.$$

Then there must exist p such that $L_p \subseteq L_i R_i X$ (otherwise $L_i X \overset{*}{\not\rightarrow} \Omega$, a contradiction).

Let $A_{i_p} \in (L_p^+ \setminus L_p) \cap X$ and let $X' = X\setminus\{A_{i_p}\}$.

Since $A_{i_p} \notin L_p$, so $L_p \subseteq L_i\ R_i\ X'$.

Therefore, it is easy to see that

$$L_i\ X' \overset{*}{\to} L_i\ R_i\ X' \overset{*}{\to} L_i\ R_i\ L_p\ R_p\ X' \overset{*}{\to} L_i\ R_i\ L_p^+\ X'.$$

Moreover, $A_{i_p} \in L_p^+$.

Consequently

$$L_i\ X' \overset{*}{\to} L_i\ X \overset{*}{\to} \Omega,$$

showing that $L_i\ X$ is not a key, a contradiction.
The proof is complete.

Corollary 3. The family

$$\{L_i(C_i \cap L_j\ C_j) \mid j \neq i, \quad 1 \leq i, \quad j \leq k\}$$

can be used to find all keys for the relation scheme S.

Property 6. Let $S = \langle\Omega,F\rangle$ be a relation scheme. If $L_i(C_i \cap L_j\ C_j) = L_i\ C_i\ \forall\ j \neq i$, then either $L_i\ C_i$ is the unique key for S including L_i, or S has no key of the form $K = L_i\ X$ with $X \subseteq C_i$.

This means that if $L_i\ C_i$ contains strictly any key of the form $L_i\ X$ then there always exists $j_o \neq i$ such that

$$L_i(C_i \cap L_{j_o}\ C_{j_o}) \subset L_i\ C_i.$$

In other words, $L_i\ C_i$ is a key for S iff

$$L_i(C_i \cap L_j\ C_j) = L_i\ C_i \quad \forall\ j \neq i$$

and there is no key of the form $K = L_i' X$ contained in $L_i C_i$ with $L_i' \subset L_i$, $X \subseteq C_i$.

Proof. Since $C_i \cap L_j C_j = C_i \quad \forall j \neq i$, it follows that

$$C_i \cap (L_j^+ \backslash L_j) = \emptyset \quad \forall j.$$

Therefore if $A \in C_i$ then

$$\{A\} \cap (L_j^+ \backslash L_j) = \emptyset, \quad \forall j.$$

Let A be any element of C_i, $A \in C_i$ and $X = C_i \backslash \{A\}$.
It is easy to see that

$$L_i X \overset{*}{\rightarrow} L_i R_i X.$$

Since $L_i R_i \cap C_i = \emptyset$ (because $L_i R_i \subseteq L_i^+$), $A \in C_i$, $A \notin X$, it follows that $A \notin L_i R_i X$.
Now suppose that there exists $L_h \subseteq L_i R_i X$, $h \neq i$.
Obviously $A \notin L_h$ and

$$L_i X \overset{*}{\rightarrow} L_i R_i X \overset{*}{\rightarrow} L_i R_i L_h R_h X.$$

It is clear that $A \notin R_h$, otherwise

$$A \in (L_h^+ \backslash L_h), \quad \text{a contradiction.}$$

By repeating the same reasoning, we can prove that

$$L_i X \overset{*}{\nrightarrow} \Omega.$$

This shows that either $L_i C_i$ is a key for S, or S has no key of the form $L_i X$, $X \subseteq C_i$.
The proof is complete.

Corollary 4. Let $S = \langle \Omega, F \rangle$ be a relation scheme.

If

$$L_i(C_i \cap L_j C_j) = L_i C_i \qquad \forall j \neq i$$

and

$$L_i \cap R_j = \emptyset \qquad \forall j \neq i$$

then $L_i C_i$ is a key for S.

Proof. The proof follows immediately by the application of Property 6 and by the fact that in this case L_i is contained in any key for S [1].

$$(L_i \backslash R = L_i \subseteq L \backslash R \subseteq K, \quad \forall K \in \mathcal{K}_S).$$

Remark 6. The case $L_i C_i$ is a key for S, we have:

$$T = L_i[(C_i \cap L_j C_j) \cap L_j(C_j \cap L_h C_h)] = L_i C_i.$$

Remark 7. Property 1 can be considered as a corollary of property 6 in the case $|C_i| = 1$ for all i.

ACKNOWLEDGEMENTS

The authors thank *Vu Duc Thi* for reading the paper and providing useful comments.

REFERENCES

[1] Ho Thuan and Le Van Bao: Some results about keys of relational schemas.
Acta Cybernetica, Tom. 7, Fasc. 1, Szeged, 1985 pp. 99-113.

[2] J. Demetrovics, Ho Thuan et al.: Balanced relation scheme and the problem of key representation.
MTA-SZTAKI Közlemenyek 32/1985, pp. 51-80.

[3] Maria C. Fernandez: Determining the normalization level of a relation on the basis of Armstrong's axioms.
Computer and Artificial Intelligence, 3 (1984) N. 6, Bratislava, pp. 495-504.

[4] Ullman, J.D.: Principles of Database systems.
Computer Science Press, Second edition, 1982.

A relációs sémák kulcsainak néhány ujabb tulajdonságáról

Demetrovics János - Ho Thuan

Összefoglaló

A szerző néhány ujabb kiegészitő tulajdonságot bizonyit a relációs sémák kulcsaira. Néhány tulajdonságot már [3]-ban bebizonyitott, de az ittlevő bizonyitások módszere már.
A tulajdonságok jól használhatók az összes kulcs megkeresésére vonatkozó algoritmusban [3].

Некоторые дополнительные свойства ключей для реляционной схемы

Я. Деметрович - Хо Тхуан

Р е з ю м е

В статье доказываются некторые дополнительные свойства ключей и сверх-ключей для реляционной схемы. Некоторые из них и их варианты были доказаны /может быть другими методами/ и использваны в обработке одного алгоритма для нахождения всех ключей для любой реляционной схемы [3].

MTA SZTAKI Közlemények 34/1986 pp. 53-78

A DISTRIBUTED DATABASE MANAGEMENT SYSTEM IN TERRITORIAL PLANNING

P. Dipotet[2], A. Benczur[3]

INTRODUCTION

Programs concerning the territorial planning and the long range development of economic regions are extremely needed, but only seldom carried out in developing countries. In Cuba, conditions have been created[1] in order to successfully face those important Programs. Regional planning is concerned with the collection, filtering, storage, processing, protection and dissemination of quite a lot of different kind of data (geographical, social, economical, etc.) spatially distributed. In other words, a very efficient and complex information system is needed, helping the experts in the regional planning environment, to make good decisions in the adequate time.
The former information system belongs then to a territorial-multi-site organisation,[1] with decentralised operations, but requiring intersite information for integrated control at the higher level.
The need of a decentralised management with local control and at the same time cooperating at a higher level for greater good, makes distributed databases very desirable for this application. With cheaper ad more

1 DIPOTET P.-81. Elaboración de un Sistema Hombre-Maquina para la Proyección del Desarrollo de Nuevos Territorios. Habana. IMACC-ACC. T-31904

2. IMACC-ACC Habana, Cuba

3. ELTE University. Budapest

reliable hardware and communication links, this technology is a real alternative to centralised processing. In our case, without big installed mainframes and with good quality micro and mini-computers available, the distributed approach has been adopted in our project.
In this paper we present the preliminary ideas of the distributed database system we are designing for a long range regional planning Program.

PROBLEM STATEMENT

When we analyze the regional planning problems[1].We observe there are external factors, that we must take as compulsory, and internal factors, particular for the Region, bounding, to some extent the development of the Region.
The external factor, policy for long range development of the country, give us:

- global requirements in products, raw materials and services from the Region;
- the external resources to be allocated in the Region during the planning period;
- indicators for social and institutional infrastructure to be developed in the Region.

The main internal factors are: natural and human resources; population requirements; tradition, experiences and material resources accumulated in the Region.

The economic development depends on the efficient management of external and internal reources in the sense of satisfying actual and forecasted requirements.

In order to have a clear picture of the former factors and, in general, of the Region current situation and trends, a retrospective analysis of its economic development was performed. The former analisys allowed us to get more precision about the objectives, tasks and functions of the Program.

Main Objectives of the Program

To elaborate alternatives for the long range socio-econimical development of the region allowing to satisfy, as much as possible, the needs, motivations and aspirations of the local population; and to maximize the regional contribution to the global social product of the country

To maximize the satisfaction of people needs, motivations and aspirations.

To develop the food and agricultural production needed to satisfy regional and national requirements.

To develop the assortment of industrial products, according to regional and national requirements and local possibilities.

To develop the social and productive infrastructure satisfying the regional development requirements.

To look for the balanced rate and proportions of the regional development

PRINCIPAL TASKS OF THE PROGRAM

To study the regional way of life and to caracterize the factors and indicators of the social development of the region.	Social Models	System for simulating the long range development alternatives of a developing region, and also to supervise and control the performences of selected plans.
To develop a model to prognosticate the population growth in the region.		
To develop models simulating alternatives for the long range agricultural and food production in the region		
To develop models simulating alternatives for the long range planning of the industrial sector of the region.	Economical Models and Systems	
To develop models and systems simulating alternatives for the long range planning of the building sector of the region.		
To develop models and systems simulating alternatives for the long range development of the regional infrastructure.		
To develop systems updating the economical, material and resources balances and communicating to given people new information and information changes.	Balance and Information Systems	

Some Indicators of the Socio-Economic Development of the Region

In order to know and to follow the development of the region and to have references allowing us decision making, some indicators are next given. These indicators give us quantitative information which can be used in economical planning, supervision and control processes.

One known indicator is the relation between fixed assets and the labor. It measures the intensity of economical investments in the region.

We denote it $F_1 = f_1 \frac{f.a.}{w_r}$, where f.a. = fixed assets in S (residual value)

w_r = number of workers.

f_1 = coeffient of propertionality

Indicators of economic efficiency are:

Productivity $= f_2 \frac{IP}{w_r} = F_2$, IP = Internal Production (S)

w_r = Number of workers

Cost per dollar $= F_3 = f_3 \frac{C}{P}$

C = total investment (S)

P = total output production (S)

Expected time to recuperate investments $= F_4 = f_4 \frac{I}{P}$

I = Investment (S)

P = production per year (S/Year)

Indicator of way of life (Related to the national media):

$$F_5 = f_5 \; \frac{E_r \cdot S_r \cdot O_r \cdot F_{2r} \cdot N_r \cdot m_r \cdot 1/F_{3r}}{E_n \cdot S_n \cdot O_n \cdot F_{2n} \cdot N_n \cdot m_n \cdot 1/F_{3n}}$$

where E = Expected life

S = mean salary

O = Ocupation coefficient = $\frac{w_r}{h}$, h=number of in habitents

N = cultural (school) level

m = migration coefficient = $1+g\,\frac{n}{h}$

n = number of inmigrants

g = coefficient (100 in our case)

$()_r, ()_n$ = regional, national mean value respectively

Former indicators are simplified ones. In developing countries these indicators are frequently affected by technological, skilfulness and tradition, organizational, etc. One of the main points in the organizational factor is the quality of the actual information system in our geographically distributed and complex environment.

In our developing region, a dynamic approach in the development of the Program[2], was applied (Figure 1). This approach allowed us to gradually improve the available information about de region, and to use also the new information in decision making.

With the improvement of the information base and, therefore, the new knowledge about the object and its environment, it is now possible to develop an information system for planning the long range development of the region. In other words, it is needed to perform the simulation of long range investments planning projects for the region, and this work must be based on some kind of man-machine system, helping the "man" (responsible for decision making) to evaluate the consequences of different plan alternatives and to control the deviations of actual plans (Fig. 2.)

The proposed information system must be geographically distributed with one center (regional node) and 14 sub-regional nodes). Each site should be capable of operating independently. In our case, each node has strong relation with the center, and weak relations with the other nodes. Most informational activity originates from pre-defined transactions that are executed repeatedly using the same statements.

2 Dipotet P. Models in the socio-econimic development of a new agricultural region.
Közlemények 27/1982. Budapest. MTA SZTAKI

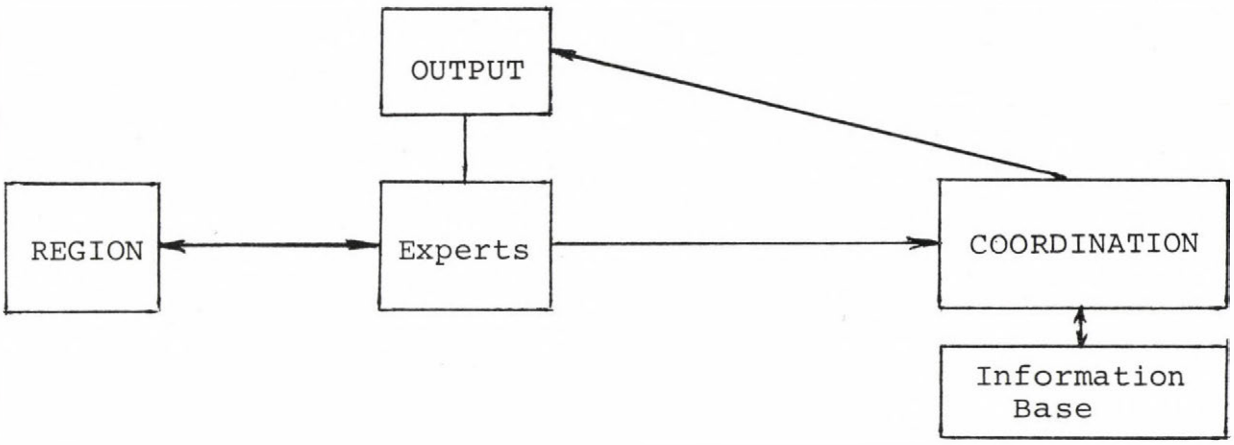


Figure 1. Preliminary system to study the Region

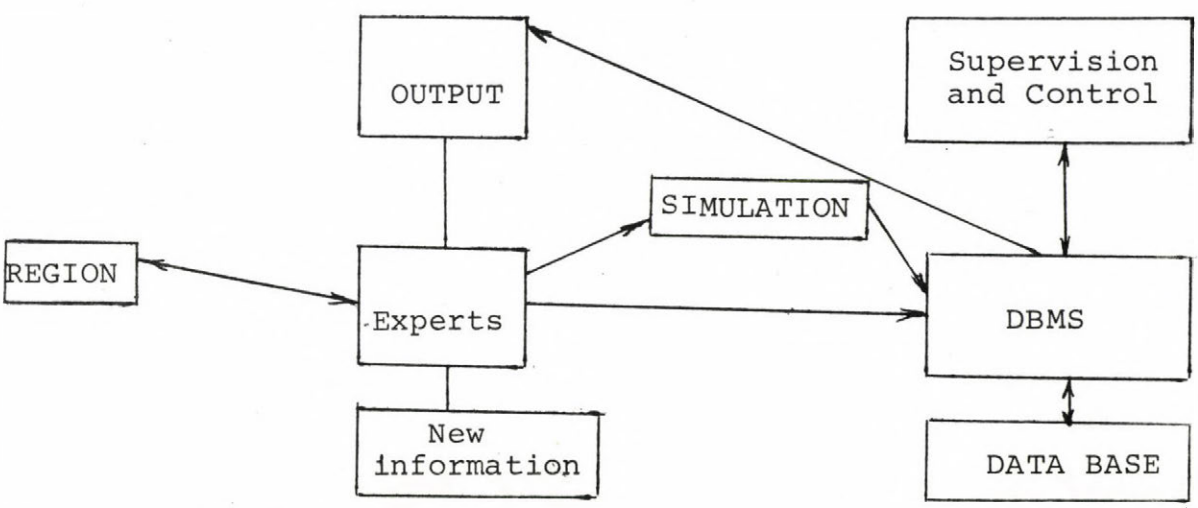


Figure 2. System for long range planning of a developing Region

In our case we require a distributed data base management system for the following reasons:[3]

- Economic. Large databases exceed the storage capacity and processing power of a single machine. In the former three years the cost of micro-machines have been lowered to be competitive with the large mainframes on which DBMS systems have traditionally relied. For several reasons, the cost of communication links between computers has also been reduced.

- Organizational. Departments expected to use DBMSs are both geographically and organizationally distributed but require integration due to new requirements and organizational changes. DBMSs must therefore be distributed, in order to model effectively the territorial organization that they serve. They must provide to each component a degree of local responsibility for, and control over, the data that it "owns".

- Technical. Performance and availability are enhanced when data can be situated close to the majority of its users, rather than in a distant center. The impact of the failure of one machine - or the link to it - is reduced, particularly when redundant copies of the same data are maintained on other machines.

- Reliability. Local data processing and site autonomy in general, allow us to have an up-to-date information, reducing what might be called "information float", which is not desirable for territorial gevernment decision making.

[3] B.G. Lindsay. Query Processing in R^*. North Holland. 1984.

PROBLEM ANALYSIS

The proposed distributed database is purpose-built and a good control can be exercised on the nature of each node and their contents. We may recognise two types of users, the global user processing the data of the distributed database under the control of some kind of Distributed Database Management System (DDBMS) and local or nodal user (figure 3) processing the data of a particular node under the control of the Nodal Database Management System /NDBMS).

Due to the presence of global and nodal DBMS we have two levels of control. The global control is centralised. All global processing are controlled by a central computer (a node act as the center) onto which all global transactions must be chanelled. The NDBMS controls the activies of each node, but these nodes are fully autonomous having its exclusive local users who need not be aware of the existence of the DDN (Figure 4). In this case some of the data captured by a node is contributed to the DDB and, of course, some of the data of the DDB is contributed to the node (Figure 5).

Next we present some definitions used in this paper.
A fragment (a set of records) is the elementary object of a DDB.
A logical fragment is a part of a logical Nodal Database.
Physical Nodal Databases consist of physical fragments
A logical Node is the owner of a Logical Nodal Database
A Physical Node is responsible for the Physical Nodal Database
A site is responsible for a physical Node.
A Node is a set of physical Nodal Databases.

Figure 3 represents the generalised architecture of the proposed homogeneous and centralised DDB.

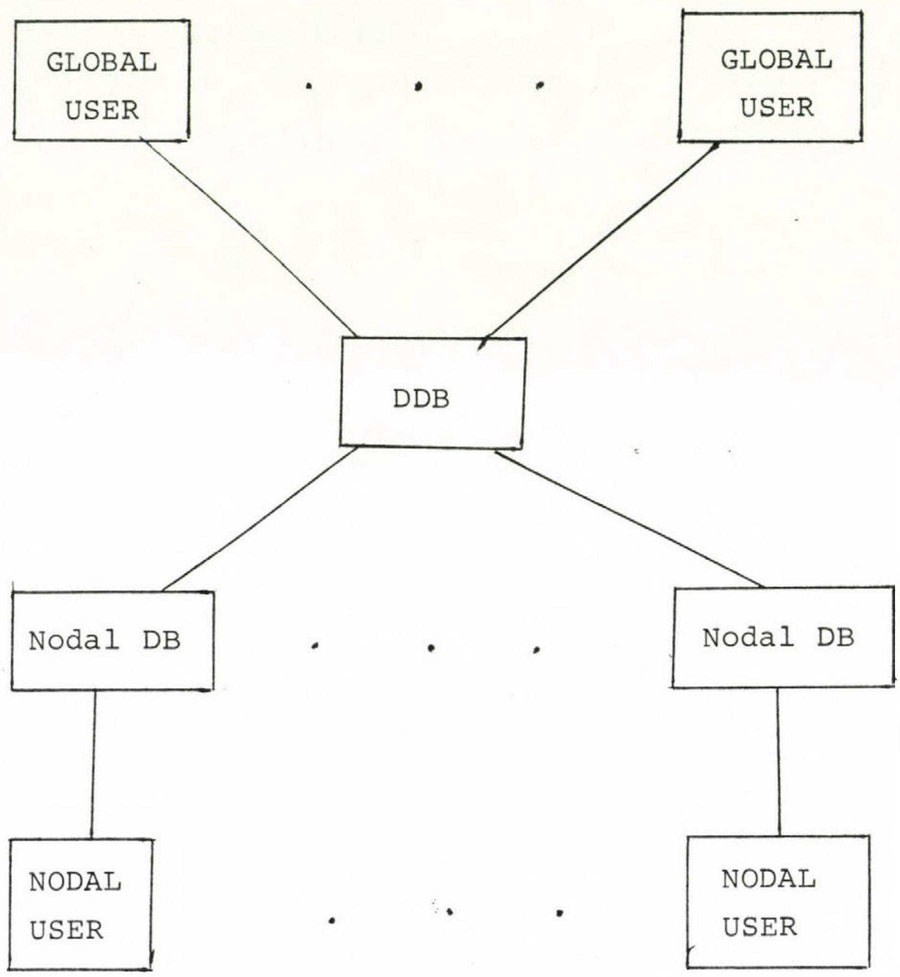


Figure 3
The distributed data base

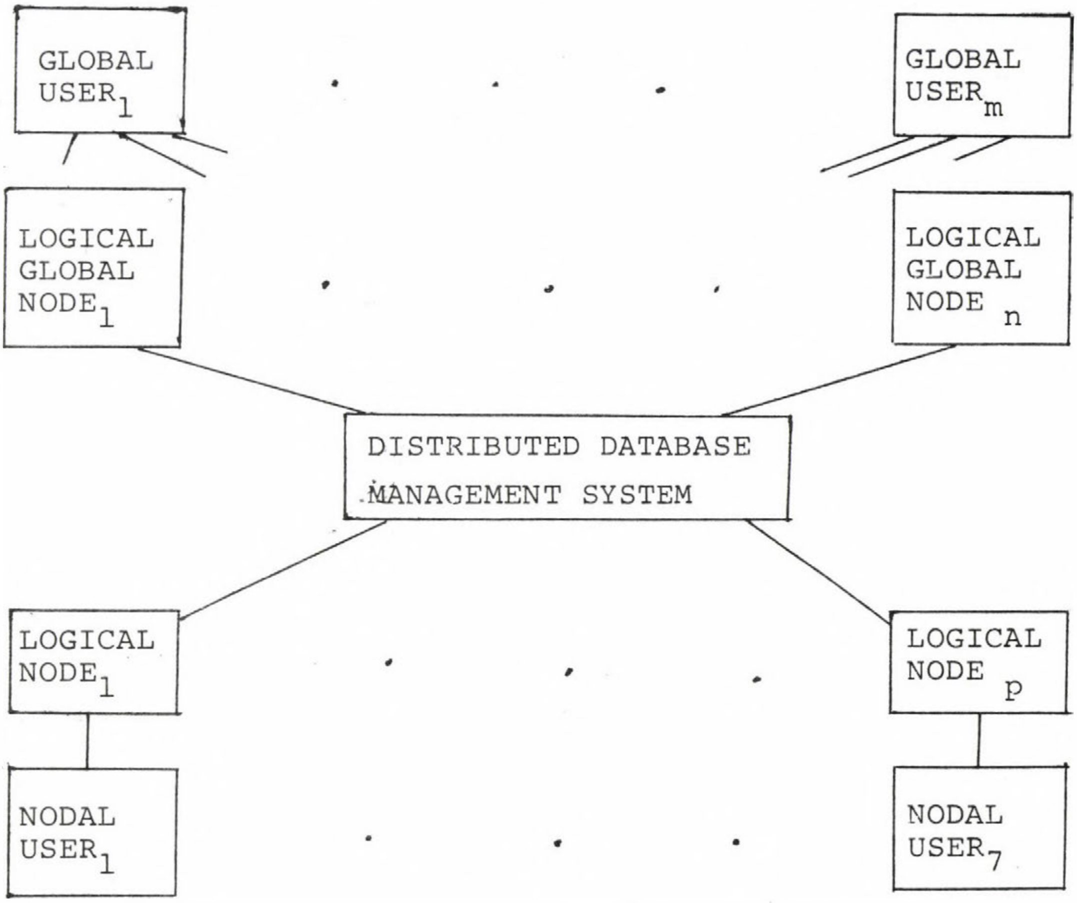


Figure 4
LOGICAL DDB

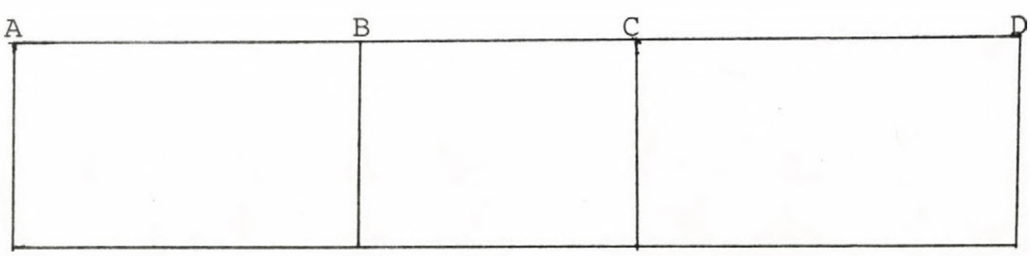


Figure 5 Logical Nodal Database

Nodal user is responsible for the collection of information A-C. The Logical Node contributes information B-C to the DDB. Information C-D is received from the DDB.

The proposed DDB is fully homogeneous allowing only identical data models at each node and having as well identical machines.
Ideally a global user should be able to formulate his transactions treating the DDB as a single database, without having to specify where the data of his interest reside (data transparency). In our case in a Logical Global Database it is possible to have the same Logical data than in a set of Logical Nodal Databases.

In order to improve processing efficiency in the DDB, a statistical analysis of users data requirements was performed.[4] This analysis allowed us to fragment data and to distribute and replicate them "optimally" over selected nodes to improve the overall performance of the DDB.

The processing of global transactions requires at least one user language that is acceptable in all nodes. When a user transaction is submitted, the DDBMS have to decompose it down to subtransactions optimally - one subtransaction for each node, taking the replicated data into account. The aim is to reduce the total cost of the execution, including the cost of communications.

In our particular case, privacy is a very sensitive issue, and safeguards for its protection is very important. We are implementing some protection facilities in all logical nodes.

[4] Dipotet P., et al. 1985. "Introduction of Mathematical Modelling and Computer Techniques in Territorial Planning." TEchnical Report. IMACC-ACC Habana.

Constraints are specified in the nodal data model /and also imposed by user programs/ in order to assure semantic integrity. The protection of the database against errors caused by the simultaneous update of related data by two or more user programs (interprocess integrity, also known as concurrency or internal consistency) is tackled by the NDBMS. The protection of the DDB as a whole,against the potential inconsistency that can be caused by the update of replicated data or related data is faced in our case (centralised control) using a locking strategy.

In our conditions, the need for database recovery is of paramount importance. It arises due to[5]: user program failure; physical storage failure (in parts or whole); processor failure; DBMS failure; Communication link failure. Up to now communication link failure is a serious problem for our proposed DDB. We are providing adequate (and expensives) backup and recovery facilities to enhance the reliability of the proposed DDB. Next future we will use special (dedicated) lines improving the quality and reliability of the communication links.

In this purpose-built DDB, the global requirements have priorities respecting local requirements, dictating completely which data should be in which node.

[5]Rien Van de R.P.; Litwin W: "Distributed Data Sharing Systems"
North-Holland, 1982.

In our case a single command is called a transaction (query or update). Transactions[6] are the units of atomic interaction between the DDB and the external world.

The DDB consists of relations (logical). Each relation is partitioned into subrelations called logical fragments which are the unit of data distribution, meaning that each may be stored at any of several sites in the system. Logical fragments are defined and the assignment of fragments to sites is made when the database is designed. A stored copy of a logical fragment is called a physical fragment.

User transactions are unaware of data distribution or redundancy. They reference relations, not fragments. It is DDB responsibility to translate from relations to logical fragments, and then to select the physical fragments to access in processing any given transaction.

[6]Bernstein P.A. et all.: Introduction to a System for Distributed Databases ACM Transactions on Database Systems. Vol. 5, No 1, March 1980

In our proposed DDB there are two types of transaction, global and nodal transactions respectively. The system of global and nodal logical databases gives a distribution of the logical fragments.

A transaction must specify the local or global database wanting to access, but only global transactions can have access to logical Global Databases.

Then, our DDB is a collection of three elements: Logical Databases, Physical Databases and Communication Network.

The Logical Databases plan and control the distributed execution of transactions, performing the following tasks:

- fragmentation, the LDB translates queries on relations into queries on logical fragments and decides which instances of stored fragment to access;

- Concurrency control, the LDB synchronizes the transaction with all other active transactions in the system.

The Physical Databases store all data managed by the DDB. PDP respond to commands from LDB /Read, Move, Manipulate, write/.

The Communication Network interconnects LDB and PDB (Global and Nodal) providing the following services:

- guaranteed delivery, allowing messages to be delivered in all circunstances;
- transaction control, guarenteing update sequenses and validation;
- site monitoring, to keep track of which sites have failed and to inform sites impacted by failures;
- network clock, a virtual clock kept approximately synchronized at all sites.

SELECTED EXAMPLE

Our application exhibits two important characteristies: First, the activity requires an integrated database. That is the activity entails access to a single pool of information by multiple persons and dependencies (global users). And second, the users of the information and ist sources (Nodes)are distributed geographically. Then, centralized control and Global Databases are needed to ensure operation in accordance wich the Regional overall policy and goals; and decentralized processing in Nodal Databases is required, for reasons of performance, reliability and flexibility of function. By meeting both former goals (centralized control and decentralized processing) in one system, the DDBMS offers very good benefits to our Program as it is possible to see in the following example:

Within a Region, it is needed to develop databases according to different functions (labor, transport, finance, health care, education, inventory, etc.) and the higher (regional or global) level must have complete access to the lower (sub-regional or nodal) levels. Nevertheless, subregions need to develop their respective plans, and require data from the higher level and from other subregions (Figure 5). In Figure 6 we present the most important actual logical data records (forming the basis of the respective logical Nodal Database) related to people (Persons) living or working in subregions. Data item identity may be used as a common (numeric) key for records shown in Figure 6.

It is apparent that the relational data model approach can be successfully applied, operating with the former four databases to get new information about the Personal (in our case Logical Nodal) Databases shown in Figure 6. The higher (regional) level contributes (Figure 5) data about national and regional educational, health care, migration and housing possibilities, normatives, etc.

Personal record (in Personal and Housing Departments)

IDENTITY	ADDRESS DATA	FAMILY DATA

Labor record (in Labor Department)

IDENTITY	ADDRESS	LABOR

Health Care record (in Health Care Department)

IDENTITY	ADDRESS	HEALTH CARE

Public Education record (in Public Education Department)

IDENTITY	ADDRESS	PUBLIC EDUCATION

Figure 6.

Logical Data Records related to persons in the Region

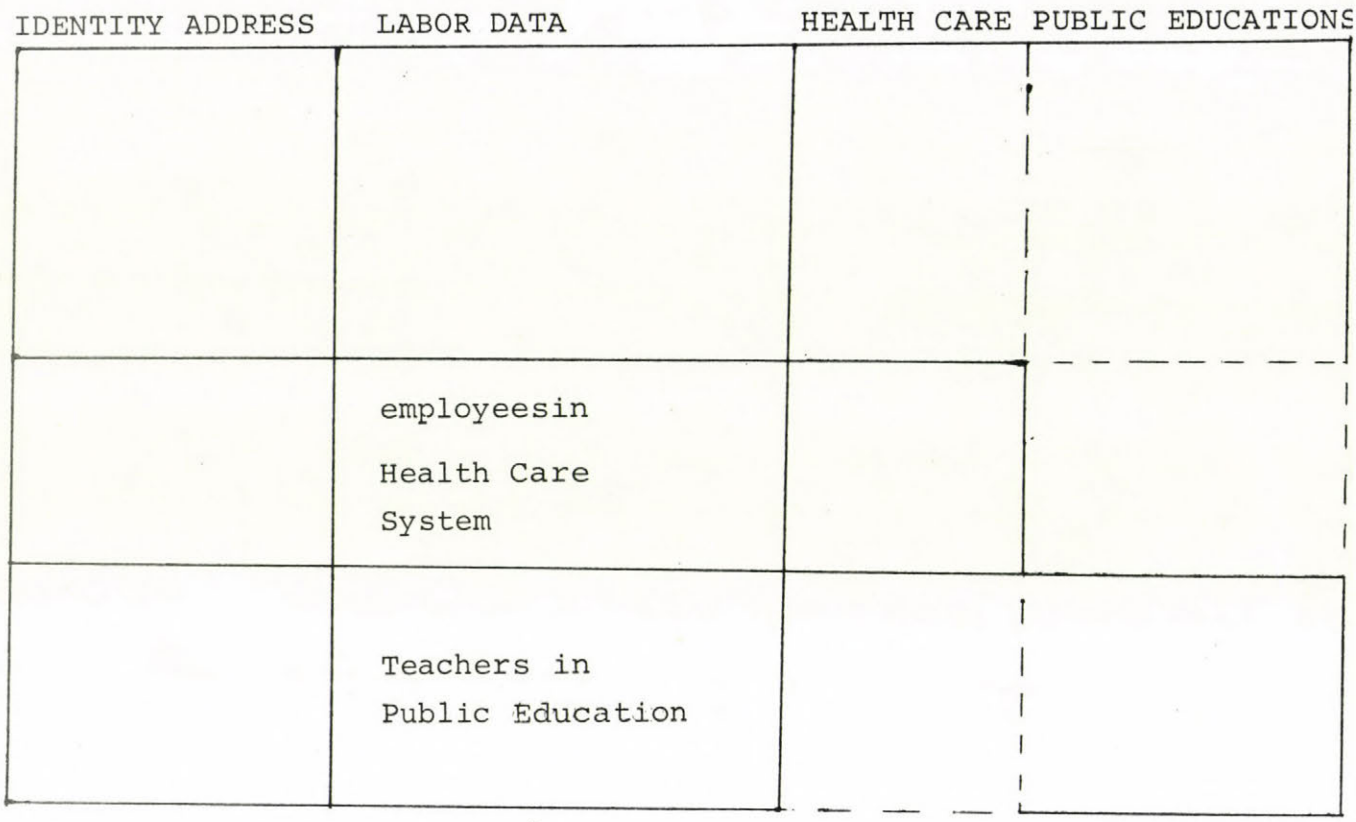
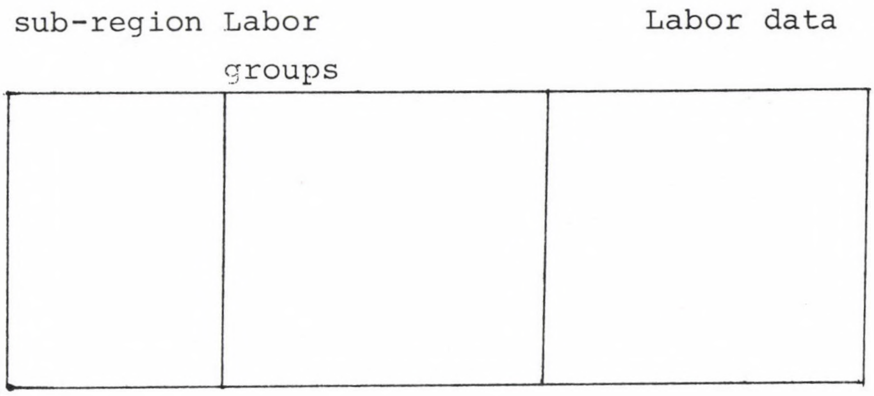


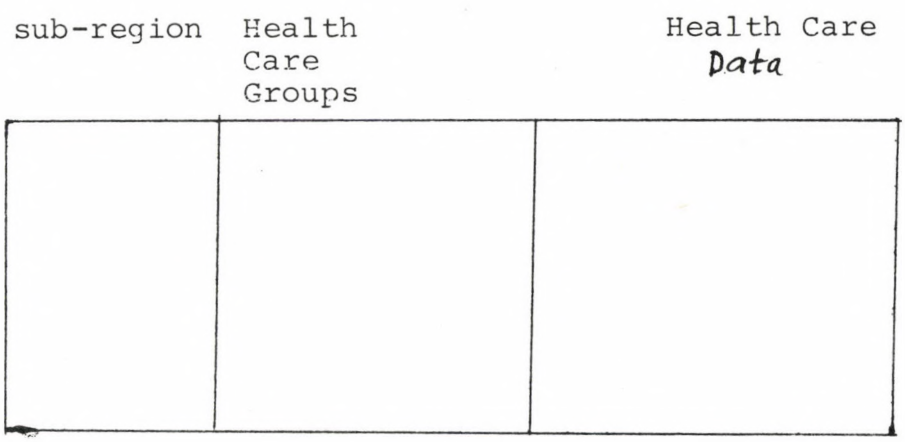
Figure 7

Personal (Logical Nodal) Database

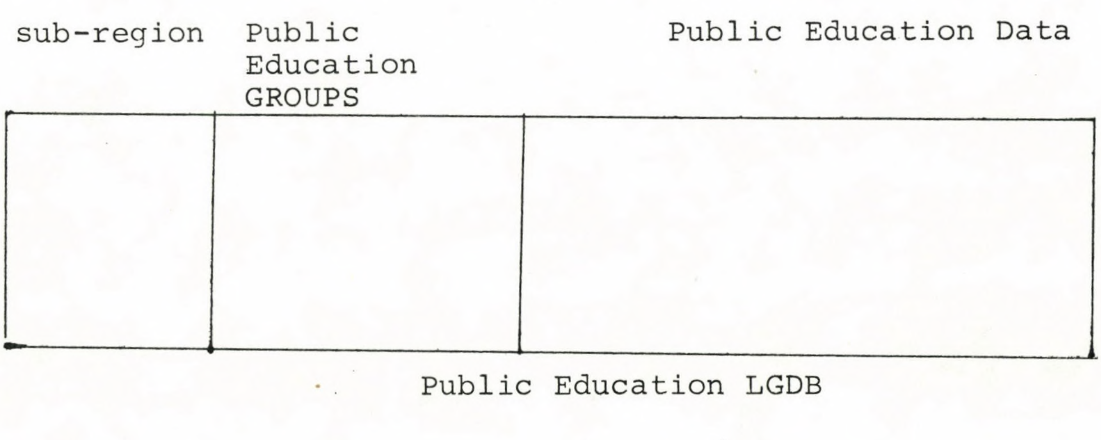
For example Health Care Logical Nodal Database consists of identity, address and health care data for each person living or working in the sub-region, and Labor data concerning employees in Health Care System of the sub-region.



Logical Global Database (LABOR LGDB)



HEALTH CARE LGDB



Public Education LGDB

Figure 8. Logical Global Databases

In the Personal LNDB, of course, using union and projection (relational operations) it is possible to derive logical fragments aswering query requirements on database relations

In Figure 8. some Logical Global Databases related with personal data, are shown. An important task for global users is to have statistical data, concerning their respective interests about LNDB. In This case, data in LGDBs are aggregated data contributed from LNDBs. Access to both Global and Nodal Databases, is controlled in sites by database administrators and also by secret passwords in the DDBMS.

It is convenient to maintain directories (for Global users) containing identity locations (identity key vs LNDB where they occur) and usage statistics. It is possible to use directories as ordinary (global) user data. Directories speed up the information retrieval process and, knowing the LGDB content, allow global users to have a complete picture of available information in the DDB.

CONCLUSION

We have discussed several aspects of the design of a DDBMS within a Program for long range planning for a development region. Constraints imposed to the system, and previous knowledge and evaluation of main global and nodal users requirements; allow us to purpose a centralised control, decentralized processing, homogeneous DDB. In our case, the relational data model is an appropriate one satisfying users and system requirements.

Next future we will present a report describing also other aspects of the proposed DDB we didn't face in this paper

Egy osztott adatbázis-kezelő rendszer területi tervezés céljára

P. Dipotet, A. Benczur

Összefoglaló

A szerzők a Kubában megvalósitandó területi tervezési rendszer számára szánt osztott adatbázis-kezelő rendszer tervezetét ismertetik.

Разделенная управляющая система базы данных в территориальном планировании

П. Дипотет, А. Бенцур

Р е з ю м е

Авторы дискутируют несколько аспектов осуществления разделенной управляющей системы базы данных для целей долгосрочного регионального планирования на Кубе.

MTA SZTAKI Közlemények 34/1986 pp. 79-89

MODULA-2 USED IN THE IMPLEMENTATION OF DATA ACCESS CONTROL MECHANISM

B. SZAFRAŃSKI

Warsaw, Poland

SUMMARY

Experience gained by using Module-2 for implementing a data security mechanism based on the data access control is presented. It entails some questions of the data access control idea but its main aim is showing the use of Module-2 in a real application. We show how in the simple way you can amplify capabilities of Modula-2 as a result of introducing data access control features.

1. INTRODUCTION

The problem of data security for any programming language system is considered in the three following aspects:

- security of standard translation facilities (like-compiler, linker, library of subprograms),
- security of translation products,
- security of data units processed by the programs written in this language.

Two first questions belong to the wider security problem of standard library and files. Such kind of data security should be resolved by the operating system and will not be considered in this paper. However, we focus on the third problem. It touches, in general principle, some questions of the language system facilities to define access rules and to detect the violations of these rules. Development of the programming languages points out, that using their capabilities to build data security mechanisms is absolutely right. Among other things, the necessity of data protection has led to such high level languages features as module structure with IMPORT and EXPORT lists, possibility to define the own structure, SCOPE rules and so on. However in data processing systems the principal question is processing of data stored in the external files. One can find out that above listed facilities refer to internal program objects and not influence directly the data security in the external storage. To confirm above we can state, that the most popular languages as COBOL, FORTRAN, PASCAL, PL/1 have no capabilities of such data security. This paper describes the experiences gained by using Modula 2 for implementation a data security mechanism based on the data access control (DACM). The first implementation was done by Mr Wlodzimierz Kubalski.

2. FILES IN MODULA-2 FOR RT-11

Access operations to data file in this disc storage always depend on the file system, which is a part of the operating system. For this reason and to save machine's independence of Modula-2 system, operations mentioned above can not be determined by the Modula-2 implementation [WIRTH80]. However, to provide program compatibility, the module Files (containing file processing procedures) was created by the authors of Modula-2. Each data file is identified by a value of type File, which in terminology RT-11 is named a channel number. The data file is assigned a channel number by calling either Lookup or Create procedures. In both cases, it is necessary to provide a file name. A file name consists of 12 characters and can be given in a static (as a literal) or a dynamic (by value of variable) form.

3. THE MODEL OF DATA ACCESS CONTROL

The complete model of data access control is presented in [SZAF78]. We show below only those elements of the model, which are important for the aim of this paper. These elements create the simplified model of data access control, which determines capabilities and operational rules of implemented mechanism [SZAF81].

Definition

Data access control model is an ordered triple:

$$S = \langle Z, T, \Psi \rangle$$

where

- Z - finite object names set,
  $Z = U \cup B$
  where:
  - U - finite active objects set,
  - B - finite passive objects set
- T - finite operation names set,
- Ψ - set of model relations,
  $\Psi = \{\Psi_1, \Psi_2\}$
  where:
  - $\Psi_1 \subset U \times B \times T$, Ψ_1 is an access relation, which determines access privileges of active to passive objects,
  - $\Psi_2 \subset T \times T$, Ψ_2 is operation scope relation. To clarify this relation let us introduce formally the definition of the operation scope:

Definition

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 operation $t_1 \in \{t_j\}$.

Definition

The operation scope $\{t_j\}\in 2^T$ of operation t_1 is smaller (equal) than the operation scope $\{t_k\}\in 2^T$ of the operation t_2 if and only if $\{t_j\}\subset\{t_k\}$.

The operation scope relation Ψ_2 is defined on pairs of operation names.

Definition

For $t_1,t_2\in T$ we say, that $(t_1,t_2)\in\Psi_2$ if and only if the operation scope t_1 is smaller (equal) than the operation scope t_2.

Definition

The process of data processing, from data access control point of view, is defined in the following way:

$$P = \{(\alpha_1,\beta_1,\gamma_1),\dots,(\alpha_i,\beta_i,\gamma_i),\dots,(\alpha_I,\beta_I,\gamma_I)\}$$

where:

$$\alpha_i\in U,\ \beta_i\in B,\ \gamma_i\in T \quad \text{for} \quad i=1,2,\dots,I.$$

Definition

Process P is correct, from data access control point of view, if and only if:

$$\forall[(\alpha_i,\beta_i,\gamma_i)\in P]\exists[(\alpha_i,\beta_i,t)\in\Psi_1]\wedge(\gamma_i,t)\in\Psi_2$$

For implementation reasons it is better to check legality of a process by using the checking function f:

Definition

$$f : U\times B\times T\rightarrow\{0,1\} \text{ in such a way, that}$$

$$f(\alpha,\beta,\gamma) = \begin{cases} 1 \text{ if and only if } \exists[(\alpha,\beta,t)\in\Psi_1]\wedge(\gamma,t)\in\Psi_2 \\ 0 \text{ in other case} \end{cases}$$

Now we can say that:

Process $P = \{(\alpha_1,\beta_1,\gamma_1),\dots,(\alpha_i,\beta_i,\gamma_i),\dots,(\alpha_I,\beta_I,\gamma_I)\}$

for $\alpha_i \in U$, $\beta_i \in B$, $\gamma_i \in T$ is correct if and only if:

$$\prod_{i=1}^{I} f(\alpha_i, \beta_i, \gamma_i) = 1$$

4. DATA ACCESS CONTROL MECHANISM - DACM

4.1 Assumptions of DACM

A selective DACM is based on the model S and entirely respects its features describes above. The concrete forms of model elements, which determine the implementation conditions of DACM are the following:

- a set of active object names U is a set of character strings representing the user identifications,
- a set of passive object names B is a set of RT-11 file names,
- an ordered set of operation names which contains operation names provided by system Modula-2 in the module Files (see Fig.1):
  T={Nil,Lookup,ReadBlock,WriteBlock,Delete,Rename,Create},
- ordering of set T, important for operation scope relation Ψ_2 , is defined by above enumerating of set elements. This means that the operation scope is growing from left to right. As you see the file T does not include the Close and Release operations because they only have the technical significance. Moreover, we have introduced the operation Nil to forbid any processing of a data unit.

Such model allows DACM to grant access selectively to the files, depending upon user identification and privileges that are included in the user access privilege file - USER.PRV.

This file is an implementation of access relation Ψ_1. The organization of the file USER.PRV bases on capability list (C-list) and it can be treated as known security matrix [HOFF77]. The checking function f we should consider as two functions, because of static and dynamic features of file name declaration function f_t is a translation time checking function

```
DEFINITION MODULE Files; /*Ch.Jacobi, for RT-11 */
FROM SYSTEM IMPORT  ADDRESS, WORD;
EXPORT QUALIFIED FILE, File Name,Lookup,Release,Create,
Delete,Close,WriteBlock,ReadBlock,Rename;
TYPE FILE  = [0..15] /* chanel number */
File Name  =  ARRAY [0..11] OF CHAR  /*File name*/
Procedure  Lookup  (f:File, FileName, VAR reply INTEGER)
              /*lookup file f in dictionary*/
```

Fig.1 A fragment of definition module Files

and f_r is a run time checking function. The function f_t is called during translation whenever any file access operation occurs. In the case of the static declaration of file name the legality checking is performed. In the other case, the call of function f_r is introduced in the procedure form to the user module. Later at run time f_r is invoked whenever it is required (see Fig.2).

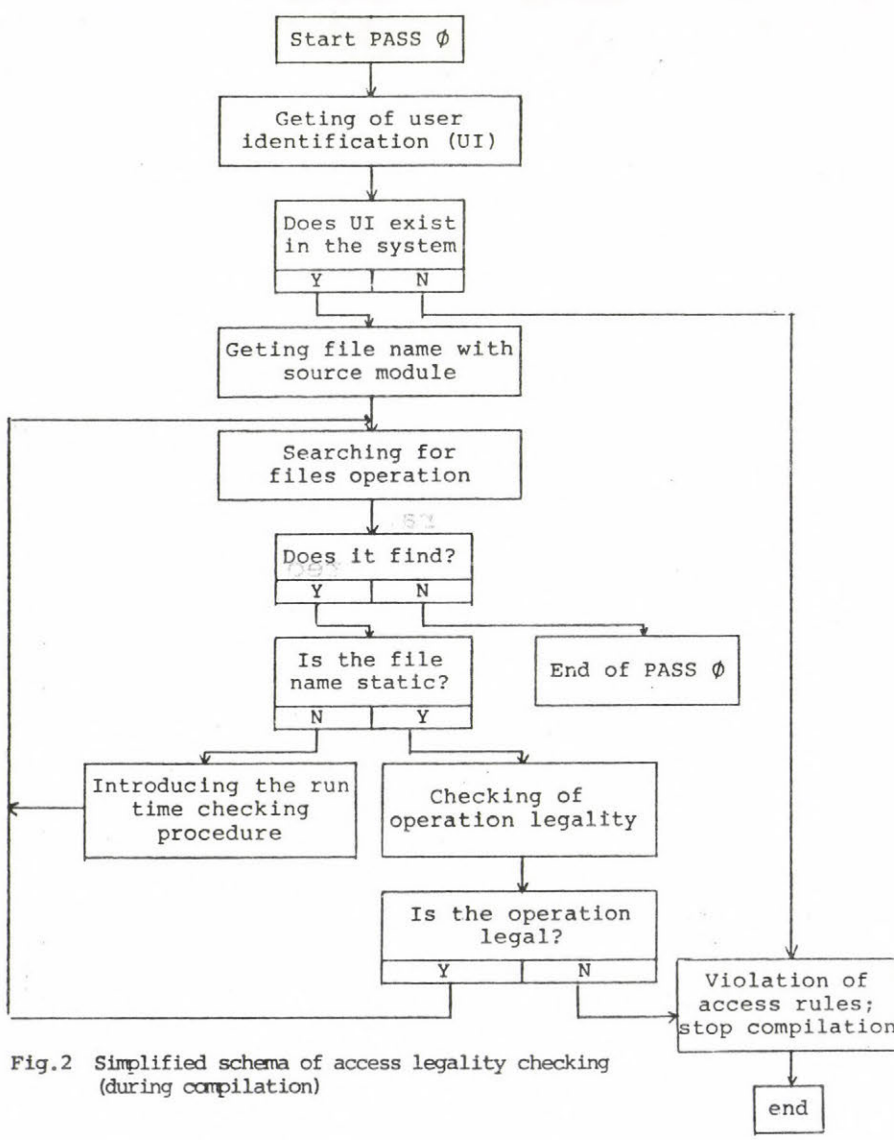


Fig.2 Simplified schema of access legality checking (during compilation)

4.2 Architecture of DACM

Implementation of the functions f_t,f_r and other elements of DACM requires additional language system features, a modification of standard language processors, and/or a language preprocessor. In our case the elements of DACM showed on the Fig.3 create two subsystems:

- subsystem of creating and maintenance file USER.PRV.
  The main element of this subsystem in the program module EDIPRV, which is used by data administrator to manage access privileges file USER.PRV.
- subsystem of operation legality checking.
  The main program module PREPROC of this subsystem realizes following functions:
  - geting and examining user's identification,
  - geting of file name including source user's module,
  - searching of file access operations and when occurs:
    - static file name - checking of operation legality (function f_t),
    - dynamic file name - introducing into user's module suitable procedure from module VERYFIER (function f_r) and into IMPORT list the name VERYFIER.

Modules USEMOD (access procedures to source user's module), USEPRV (access procedures to file USER.PRV), VERYFIER (checking procedures) include the auxiliary procedures which are used by PREPROC and EDIPRV.

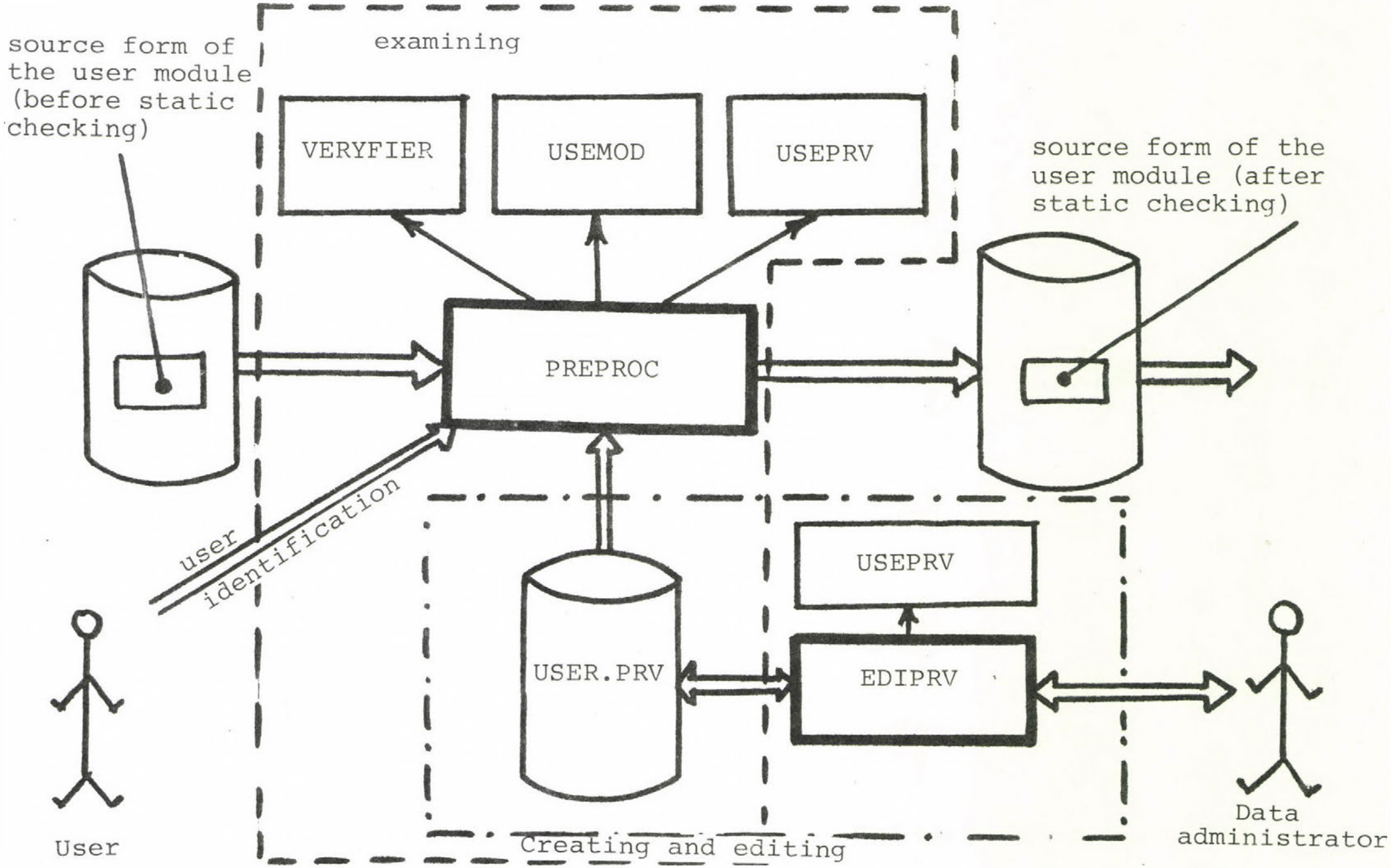


Fig.3 Elements of DACM

4.3 Compilation in MODULA-2 with DACM

The compiler is running on the Modula-2 system. It is written in MODULA-2 itself and generates code for PDP-11. The compiler is organized in a base part and several so called main parts. The base part remains in the memory during the whole compilation and schedules the execution of the main parts, which are called sequentially [WIRTH81]. The compiler has five passes (from 1 to 5) and the interpass files. According to the above, the DACM mechanism in current version was implemented as a PASS Ø of the compiler. The main part PREPROC is linked to COMP. Therefore some changes have appeared in the compile process. The compiler, as before, is invoked by typing the file name COMP but instead of string "source file" will appear the string "user identification". So one must write the corresponding UI. If it is right, the string "input file" and next "output file" will appear. The input file (default extention PRV) consists of source user's module, which will be checked by DACM. The output file is an interpass file between PASS 0 and PASS 1. It consists of the normal source file (module DEF or MOD but default extension is MOD) in terminology Modula-2 system by after checking by DACM. The following passes are running without any changes. If during PASS Ø the violation of access rules occurs, the compilation will stop and the error message will be written.

```
R MODULA <cr>
  COMP   <cr>
user ident >U 124   <cr> /*user identification*/
input file >PROG1   <cr> /*name DK:  PROG1. PRV is accepted*/
output file>PROG1   <cr> /*name DK:  PROG1. MOD is accepted*/
p0                       /*pass of access rule checking*/
p1
p2                       /*indicates succession of*/
p3                       /*activated compiler passes*/
p4
p5
end compilation
*
```

Fig.4 An example of compilation process with DACM

5. CONCLUSIONS

The problem of data security exists not only in data base but also in general data management systems (for example in file systems). Therefore, we should develop data security mechanisms in such systems. The best approach to design that (in my opinion) relies on capabilities built into programming language. The DACM implementation described in this paper provides a nearby satisfactory solution of this problem. However, this work should be completed and generalized in a number of directions. The following two directions are particularly important:

1. The problem how to add the DACM capabilities to MODULA-2 definition.
2. The problem how to expand possibilities of DACM for checking data access not only files but also to elements of files (for example record occurences, field of record).

REFERENCES

[HOFF77] Hofmann, L.: Modern methods for Computer Security and Privacy. New Jersey, USA, 1977.

[SZAF78] Szafranski, B.: The questions of program's data security. Diss., MAT, Poland, 1978.

[SZAF79] Szafranski, B.: A data security model in data base. ICS PAS Reports, Warsaw, Poland, 1979.

[WIRTH80] Wirth, N.: Modula-2. ETM, 1980.

[WIRTH81] Wirht, N.: Overview of the Modula-2 Compiler. M2 RT11, ETM, 1981.

[WIRTH82] Wirth, N.: Programming in Modula-2. Springer-Verlag, Berlin, 1982.

Adat-elérési vezérlési mechanizmusok megvalósitása Modula-2 segitségével

B. Safranski

Összefoglaló

A szerző azokat a tapasztalatokat ismerteti, amelyeket a Modula-2 felhasználásával megvalósitott adat-biztonsági mechanizmusok terén nyert.

Применение языка МОДУЛА-2 в осуществлении механизма контроля доступа к данным

Б. Шафраньски

Р е з ю м е

В статье представляется опыт собранный во время осуществления механизма защиты данных опирающегося на управлении доступа к данным на языке МОДУЛА-2. Этот механизм дает возможность контроля операции доступа к данным во время компиляции и исполнения программы. Механизм построен на основе формальной модели процесса защиты данных.

MTA SZTAKI Közlemények 34/1986 pp.91-104

ON STRONG OPERATIONS

Vu Duc Thi

MTA STTAKI

§1. INTRODUCTION

The families of strong dependencies were introduced and investigated in [1,2]. In this paper we define strong operations and investigate the properties of strong operations. Based on these properties we give some combinational results which are related to the families of strong dependencies.

First we give some necessary definitions, and in §2. formulate our results.

§1. DEFINITIONS

Definition 1.1. Let $R=\{h_1,\ldots,h_m\}$ be a relation over the finite set of attributes Ω, and $A,B\subseteq\Omega$. Then we say that B strongly depends on A in R (denote $A \overset{s}{\underset{R}{\rightarrow}} B$) if

$$(\forall h_i,h_j\in R)((\ a\in A)(h_i(a)=h_j(a))\rightarrow(\forall b\in B)(h_i(b)=h_j(b)));$$

B functionally depends on A in R (denote $A \overset{f}{\underset{R}{\rightarrow}} B$) iff

$$(\forall h_i,h_j\in R)((\forall a\in A)(h_i(a)=h_j(a))\rightarrow(\forall b\in B)(h_i(b)=h_j(b))).$$

Let

$$S_R=\{(A,B)\ :\ A \overset{s}{\underset{R}{\rightarrow}} B\}\ .$$

S_R is called the full family of strong dependencies.

Definition 1.2. Let R_1, R_2 be two relations over Ω. We say that R_1 and R_2 are s-equivalent if $S_{R_1}=S_{R_2}$.

R_1 is an irredundant relation if for all $R' \subset R_1$: $S_{R'} \neq S_{R_1}$.

Definition 1.3. Let Ω be a finite set, and denote $P(\Omega)$ its power set. Let $Y = P(\Omega) \times P(\Omega)$. We say that Y satisfies the S-axioms iff for any $A,B,C,D \subseteqq \Omega$, $a \in \Omega$.

(S1) $(\{a\}, \{a\}) \in Y$;

(S2) $(A,B) \in Y$, $(B,C) \in Y$, $B \neq \emptyset \rightarrow (A,C) \in Y$;

(S3) $(A,B) \in Y$, $C \subseteqq A$, $D \subseteqq B \rightarrow (C,D) \in Y$;

(S4) $(A,B) \in Y$, $(C,D) \in Y \rightarrow (A \cap C, B \cup D) \in Y$;

(S5) $(A,B) \in Y$, $(C,D) \in Y \rightarrow (A \cup C, B \cap D) \in Y$.

It is clear that S_R satisfies the S-axioms.

Definition 1.4. Let $Y = P(\Omega) \times P(\Omega)$.

We say that Y satisfies the C-axiom iff there is a family of subsets of Ω, $\{E_i : i=1,\dots,\ell ; \bigcup_{i=1}^{\ell} E_i = \Omega\}$ such that

(i) for any $A,B \subseteqq E_i \rightarrow (A,B) \in Y$;
$E_i \cap A \neq \emptyset$

(ii) $(C,D) \in Y$, $C \cap E_i \neq \emptyset \rightarrow D \subseteqq E_i$.

§2. RESULTS

Theorem 2.1. Let $Y = \underline{P}(\Omega) \times P(\Omega)$. Then Y satisfies the S-axioms iff Y satisfies the C-axiom.

Proof. First we suppose that Y satisfies the S-axioms. Then by (S1), (S3), and (S5) for each $a \in \Omega$ we can construct an E_i ($E_i \subseteqq \Omega$) so that $(\{a\}, E_i) \in Y$, and $\forall E' : E_i \subset E'$ implies $(\{a\}, E') \notin Y$. It is obvious that $a \in E_i$, and we obtain n such E_i-s, where $|\Omega|=n$. Thus, we have the set $E=\{E_i : i=1,\dots,n; \bigcup_{i=1}^{n} E_i = \Omega\}$. We assume that $A=\{a_1,\dots,a_k : a_j \in \Omega, j=1,\dots,k\} \neq \emptyset$, and B_1 is a set such that $(A,B_1) \in Y$, $\forall B_2 : B_1 \subset B_2$ implies $(A,B_2) \notin Y$. According to the construction of E, it is clear that for each a_j there is an $E_{i_j} \in E$ so that $(\{a_j\}, E_{i_j}) \in Y$. By (S4) we have $(\bigcup_{j=1}^{k} a_j, \bigcap_{j=1}^{k} E_{i_j}) = (A, \bigcap_{j=1}^{k} E_{i_j}) \in Y$. By the definition of B_1 we obtain $\bigcap_{j=1}^{k} E_{i_j} \subseteq B_1$. On the otherhand, by $(A,B_1) \in Y$, and by (S3) we have $(\{a_j\}, B_1) \in Y$ for all j $(j=1,\dots,k)$. Consequently, $B_1 \subseteq \bigcap_{j=1}^{k} E_{i_j}$, i.e. $B_1 = \bigcap_{j=1}^{k} E_{i_j}$. It is obvious that $\bigcap_{E_i \cap A \neq \emptyset} E_i \subseteq \bigcap_{j=1}^{k} E_{i_j}$. Thus, for all B $(B \subseteq \bigcap_{E_i \cap A \neq \emptyset} E_i)$: $B \subseteq B_1$.

Consequently, $(A,B) \in Y$. If $(C,D) \in Y$, $C \cap E_i \neq \emptyset$, then we assume that $a_1 \in C \cap E_i$. On the otherhand, suppose that a is an attribute such that $(\{a\}, E_i) \in Y$, and $\forall E' : E_i \subset E'$ implies $(\{a\}, E') \notin Y$. By $a_1 \in E_i$, and (S3) $(\{a\}, \{a_1\}) \in Y$ holds. By (S3), and $a_1 \in C$ we obtain $(\{a_1\}, D) \in Y$.

Consequently, by (S2), and $a_1 \neq \emptyset$ $(\{a\}, D) \in Y$ holds. According to the definition of E_i we have $D \subseteq E_i$. Thus, Y satisfies the C-axiom. It can be seen that by convention $\cap \emptyset = \Omega$, for all $B (B \subseteq B)$ we have $(\emptyset, B) \in Y$. The proof of the reverse direction is easy and so will be omitted. The theorem is proved.

Now, we define thefollowing operation.

Definition 2.2. Let Ω be a finite set. The mapping $F: P(\Omega) \to P(\Omega)$ is called a strong operation over Ω if for every $a,b \in \Omega$, and $A \subseteq \Omega$, the following properties hold:

(i) $a \in F(\{a\})$,

(ii) $b \in F(\{a\}) \to F(\{b\}) \subseteq F(\{a\})$,

(iii) $F(A) = \bigcap_{a \in A} F(\{a\})$.

Remark 2.3. It is easy to see the following elementary properties of strong operations.

For $A,B \in P(\Omega)$: $F(A \cup B) = F(A) \cap F(B)$. By convention $\cap \emptyset = \Omega$ we obtain $F(\emptyset) = \cap \emptyset = \Omega$.

For $A \subseteq B$ $F(B) \subseteq F(A)$.

Definition 2.4. Let $Y = P(\Omega) \times P(\Omega)$. We say that Y is an s-family over Ω, if Y satisfies the S-axioms.

Lemma 2.5. Let S be an s-family over Ω. We define the mapping F_S: $P(\Omega) \to P(\Omega)$ as follows: $F_S(A)=\{a\in\Omega: (A,\{a\})\in S\}$. Then F_S is a strong operation. Conversely, if F is a strong operation, then there is exactly one s-family S so that $F=F_S$, where $S=\{(A,B): A,B\in P(\Omega): B\subseteq F(A)\}$.

Proof. Suppose that S is an s-family. It is obvious that $\forall a\in\Omega: a\in F_S(\{a\})$. By Theorem 2.1. S satisfies the C-axiom, and so we have $(C,D)\in S$, and $C\cap E_i\neq\emptyset$ implies $D\subseteq E_i$. It can be seen that in Theorem 2.1 for any $a\in\Omega$, $F_S(\{a\})\in \{E_i: i=1,\ldots,n,\ /\Omega/=n\}$. Consequently, $(b,F_S(b))\in S$, $b\in F_S(\{a\})$, i.e. $(A,F_S(A))\in S$, $\forall a\in A$: $A\cap F_S(a)\neq\emptyset$ imply $F_S(A) = F_S(\{a\})$. Thus, $F_S(A) \subseteq \bigcap_{a\in A} F_S(\{a\})$. On the other hand, by (S5) in the S-axioms we obtain $\forall a\in A$: $(\{a\}, F_S(\{a\}))\in S$ implies $(A,\bigcap_{a\in A} F_S(\{a\}))\in S$, i.e. $\bigcap_{a\in A} F_S(\{a\})\subseteq F_S(A)$. Consequently, $F_S(A)=\bigcap_{a\in A} F(\{a\})$ holds.

Conversely, assume that F is a strong operation over Ω and $S=\{(A,B): B\subseteq F(A)\}$. We have to show that S is an s-family. By Theorem 2.1 we prove that S satisfies the C-axiom. We set $E=\{F(\{a\}): a\in\Omega,\ /\Omega/=n\}$. By the definition of S it is obvious that $B\subseteq \bigcap_{F(\{a\})\cap A\neq\emptyset} F(\{a\})$ implies $(A,B)\in S$ by $\bigcap_{F(\{a\})\cap A\neq\emptyset} F(\{a\}) \subseteq F(A)$. On the other hand if $(C,D)\in S$, and $C\cap F(\{a\})\neq\emptyset$, then we assume that $b\in C\cap F(\{a\})$, hence by (ii) $b\in F(\{a\})$ implies $F(\{b\}) \subseteq F(\{a\})$. It is obvious that $D\subseteq F(C)=\bigcap_{d\in C} F(\{d\})$. By $b\in C$, and $\bigcap_{d\in C} F(\{d\})\subseteq F(\{b\})$ we obtain $D\subseteq F(\{a\})$. It is clear that $\forall A\subseteq\Omega$: $(\emptyset,A)$, $(A,\emptyset)\in S$. It can be seen that $F=F_S$. Now, we suppose that there is a s-family S' so that $F_{S'}=F$. By the definition of S and F we obtain $S'\subseteq S$. If $(A,B)\in S$, then $B\subseteq F(A)=F_{S'}(A)$. By the definition of $F_{S'}$, we have $(A,B)\in S'$. Consequently, $S'=S$ holds. The proof is complete.

Remark 2.6. Clearly, if F_1 and F_2 are strong operations $(F_1\neq F_2)$, then $S_1\neq S_2$, where for $i=1,2$: $S_i=\{(A,B): A,B\subseteq\Omega: B\subseteq F_i(A)\}$.

Definition 2.7. Let S be an s-family over Ω. We say that a relation R represents S iff $S_R=S$.

In [2] the equality sets of relation are defined as follows.

Definition 2.8. Let $R=\{h_1,\ldots,h_m\}$ be a relation over Ω. Denote by E_{ij} the set $\{a\in\Omega: h_i(a)=h_j(a),\ 1\le i<j\le m\}$. We set $E=\{E_{ij}: 1\le i<j\le m\}$.

Now, we give a necessary and sufficient condition for a relation R to represent an s-family.

Theorem 2.9. Let S be an s-family, and R be a relation over Ω. Then R represents S iff for each $a\in\Omega$:

$$
\begin{aligned}
F_S(\{a\}) &= \bigcap_{a\in E_{ij}} E_{ij} \quad \text{if } \exists E_{ij} : a\in E_{ij} \\
F_S(\{a\}) &= \Omega \quad \text{otherwise.}
\end{aligned}
\tag{1}
$$

Proof. By Lemma 2.5 $S_R=S$ holds if and only if $F_{S_R}=F_S$. Consequently, first we show that

$$F_{S_R}(\{a\}) = \bigcap_{a\in E_{ij}} E_{ij} \quad \text{if} \quad E_{ij}: a\in E_{ij} ,$$

and in other case $F_{S_R}(\{a\})=\Omega$ holds. Clearly, $F_{S_R}(\{a\})=$ $=\{b\in\Omega : \{a\} \overset{s}{\underset{R}{\to}} \{b\}\}$. Accordint to the definition of strong dependency we know that for any $a\in\Omega$: $\{a\} \overset{s}{\underset{R}{\to}} B \leftrightarrow \{a\} \overset{f}{\underset{R}{\to}} B$, where $a\neq\emptyset$. Let us denote by T the set $\{E_{ij} : a\in E_{ij}\}$. It is obvious that if $T=\emptyset$, then $\{a\} \overset{f}{\underset{R}{\to}} \Omega$, i.e. $F_{S_R}(\{a\})=\Omega$. If $T\neq\emptyset$ holds, then we set $A = \bigcap_{a\in E_{ij}} E_{ij}$. If $T=E$ holds (E is the set of all equality sets of R), then it is obvious that $\{a\} \overset{f}{\underset{R}{\to}} A$. If $T\subset E$ holds, then for E_{ij}: $E_{ij}\notin T$, $h_i(a)\neq h_j(a)$. Consequently, we have also $\{a\} \overset{f}{\underset{R}{\to}} A$. Denote A' the set with the following properties:

(i) $\{a\} \overset{f}{\underset{R}{\to}} A'$,

(ii) $A'\subset A''$ implies $\{a\} \overset{f}{\underset{R}{\to}} A''$.

It can be seen that $A'=A$. According to the definition of F_{S_R} we obtain $F_{S_R}(\{a\}) = \bigcap_{a\in E_{ij}} E_{ij}$. Thus, if $S_R=S$ holds, then

$$F_S(\{a\}) = \bigcap_{a\in E_{ij}} E_{ij} \quad \text{if } \exists\, E_{ij} : a\in E_{ij} ,$$

$$F_S(\{a\}) = \Omega \quad \text{otherwise.}$$

Conversely, if F_S satisfies (1), then according to the above part for any $a \in \Omega$ we have $F_S(\{a\}) = F_{S_R}(\{a\})$. Because F_S and F_{S_R} are strong operations, and by Lemma 2.5. We obtain $\forall A \subseteq \Omega$: $F_S(A) = F_{S_R}(A)$. Consequently, $F_S = F_{S_R}$ holds. The proof is complete.

We say that a relation R represents a strong operation F iff $F = F_{S_R}$. Based on Theorem 2.9, the next Corollary is obvious.

Corollary 2.10. Let F be a strong operation, R be a relation over Ω. Then R represents F iff

$$F(\{a\}) = \bigcap_{a \in E_{ij}} E_{ij} \quad \text{if } \exists E_{ij} : a \in E_{ij} \ ,$$

$$F(\{a\}) = \Omega \quad \text{otherwise} \ ,$$

where $a \in \Omega$.

Clearly, from a relation R we can construct the set of all equality sets of R. Consequently, a following corollary is also obvious.

Corollary 2.11. Let R be a relation, and F be a strong operation over Ω. Then there is an effective algorithm, that decide whether R represent F or not. This algorithm requires time polynomial in the number of rows and columns of R.

Based on Theorem 2.9. We going to construct an effective algorithm, which determines an irredundant relation.

Algorithm 2.12. Let $R = \{h_1, \ldots, h_m\}$ be a relation over $\Omega = \{a_1, \ldots, a_n\}$.

Step 1: From relation R we construct

$$E = \{E_{ij} : E_{ij} \text{ is an equality set of R} \quad 1 \le i < j \le m\} \ .$$

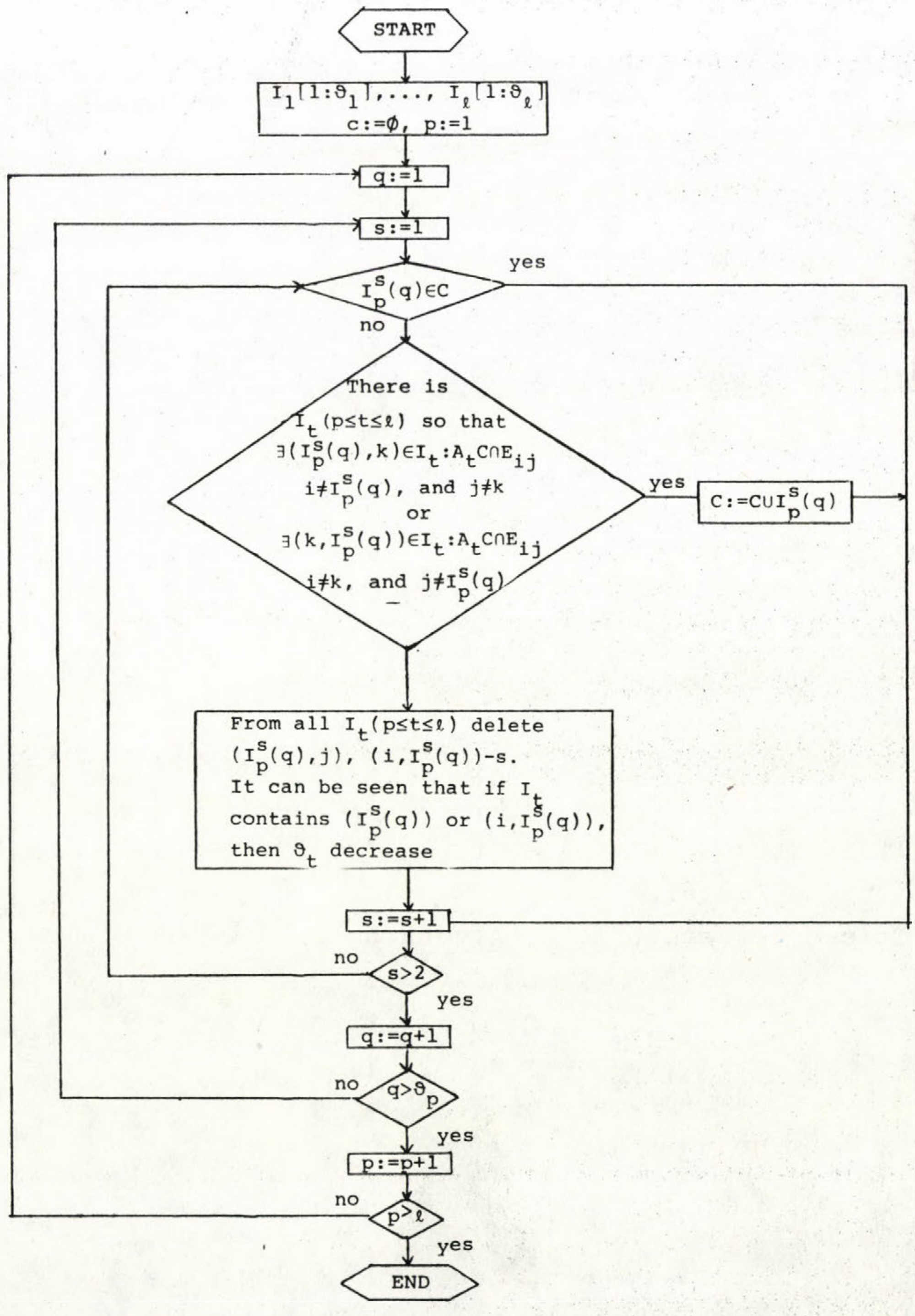
If there is not an E_{ij} so that $E_{ij} \neq \emptyset$ / then we choose any pair h_i, h_j. It is obvious that $R' = \{h_i, h_j\}$ is an irredundant relation such that $S_{R'} = S_R$. Now, we assume that $a_{t_1}, \ldots, a_{t_\ell}$ are attributes such that $E_{ij} : a_{t_q} \in E_{ij}$, where $q = 1, \ldots, \ell$.

Step 2: We construct sets of indexpairs, as follows: $I_1=\{(i,j):\ a_{t_1}\in E_{ij}\},\dots,I_\ell=\{(i,j):\ a_{t_\ell}\in E_{ij}\}$. Let $A_q=\bigcap_{a_{t_q}\in E_{ij}} E_{ij}$,

$$q=1,\dots,\ell.$$

It is obvious that $\ell\le n$. Denote by ϑ_i the number of elements of I_i, $i=1,\dots,\ell$.

Denote $I_i^1(j)$, and $I_i^2(j)$ the first and second indices of j-th pair in I_i, $i=1,\dots,\ell$; $1\le j\le\vartheta_i$. After that we perform the following bloc-scheme.



Then $R'=\{h_i : i\in C\}$ is an irredundant relation such that $R'\subseteqq R$, and $S_{R'}=S_{R'}$ i.e. $S_{R'}$ and S_R are s-equivalent.

Proof. It is clear that by Theorem 2.9., and Lemma 2.5. we have $S_{R'}=S_R$. It can be seen that in Step2 the Algorithm formulated in the bloc-scheme deletes all redundant rows of R. Thus, R' is an irredundant relation.

The proof is complete.

Remark 2.13. It can be seen that Algorithm 2.12 requires time polynomial in the number of rows and columns of R.

Example 2.14. Let $\Omega=\{1,2,3,4,5,6\}$ and

$$R=\{h_1,h_2,h_3,h_4,h_5,h_6,h_7\}$$

be a relation over Ω.

$$\begin{array}{rcccccc} \text{Attributes:} & 1 & 2 & 3 & 4 & 5 & 6 \\ & & & & & & \\ & 1 & 0 & 0 & 2 & 2 & 2 \\ & 2 & 0 & 0 & 3 & 3 & 0 \\ & 3 & 3 & 3 & 2 & 1 & 0 \\ R = & 1 & 1 & 2 & 4 & 3 & 3 \\ & 4 & 2 & 4 & 1 & 0 & 1 \\ & 5 & 2 & 4 & 5 & 4 & 4 \\ & 6 & 4 & 5 & 5 & 5 & 5 \end{array}$$

Clearly, $E_{12}=E_{56}=\{2,3\}$, $E_{23}=E_{67}=\{4\}$, $E_{14}=\{1\}$, $E_{23}=\{6\}$, and $E_{24}=\{5\}$.

Consequently, $I_1=\{(1,4)\}$, $I_2=\{(1,2),(5,6)\}$, $I_3=\{(1,2),(5,6)\}$, $I_4=\{(1,3),(6,7)\}$, $I_5=\{(2,4)\}$, and $I_6=\{(2,3)\}$.

It is obvious that $A_2=\{1\}$, $A_2=\{2,3\}$, $A_3=\{2,3\}$, $A_4=\{4\}$, $A_5=\{5\}$, and $A_6=\{6\}$.

It can be seen that by Algorithm 2.12 we obtain $C=\{1,2,3,4\}$, i.e.

$$R'=\begin{array}{cccccc} 1 & 0 & 0 & 2 & 2 & 2 \\ 2 & 0 & 0 & 3 & 3 & 0 \\ 3 & 3 & 3 & 2 & 1 & 0 \\ 1 & 1 & 2 & 4 & 3 & 3 \end{array}$$

Definition 2.15. Let $I \subseteq P(\Omega)$, and I closed under intersection. Let $M = P(\Omega)$. Denote M^+ the set $\{\cap M' : M' \subseteq M\}$. We say that M generates I if $M^+ = I$.

By convention $\cap\emptyset = \Omega$, i.e. M^+ always contains Ω; so Ω is never required in M. It is obvious that $\Omega \in I$.

J.Demetrovics in [2] showed that for a given family M of subsets of Ω there is exactly one family N, which generates M^+, and has minimal cardinality.

Lemma 2.16. [2] Let $M = P(\Omega)$ be a family over Ω. Let $N = \{A \in M : (\forall B, C \in M)(A = B \cap C \rightarrow A = B \text{ or } A = C)\}$. Then N generates M^+ and if N' generates M^+, then $N \subseteq N'$. It is possible that $\emptyset \in N$.

By Remark 2.3 we obtain $F(A \cup B) = F(A) \cap F(B)$ for $A, B \in P(\Omega)$, where F is a strong operation. Thus, the set $\{F(A) : A \subseteq \Omega\}$ closed under intersection. It is easy to see that

The set $\{F(\{a\}) \neq \Omega : a \in \Omega\}$ generates $\{F(A) : A \subseteq \Omega\}$. It is possible that $a \neq b$, but $F(\{a\}) = F(\{b\})$. It is known (see [1,2]) that if S is an s-family over Ω, then there is a relation R over Ω such that $S_R = S$. However, here we construct for a given s-family S a simple concrete relation R so that R represents S.

Proposition 2.17. Let S be an s-family over $\Omega = \{a_1, \dots, a_n\}$. Let $\{F_S\ \{a\} : i = 1, \dots, n\} = \{A_1, \dots, A_k : A_i \neq A_j,\ 1 \leq i \leq k,\ 1 \leq j \leq k,\ i \neq j,\ k \leq n\}$. We set $T = \{A_1, \dots, A_k\}$, and
$N = \{A \in T : A \neq \Omega,\ (\forall B, C \in T)(A = B \cap C \rightarrow A = B \text{ or } A = C)\}$. Suppose that $N = \{B_1, \dots, B_\ell\}$, $(\ell \leq k)$. Then we set $R = \{h_0, h_1, \dots, h_\ell\}$ as follows:
for all $a \in \Omega$: $h_0(a) = 0$.

$$\text{for each } i\ (i = 1, \dots, \ell) : h_i(a) = \begin{cases} 0 & \text{if } a \in B_i , \\ i & \text{otherwise} . \end{cases}$$

Then R represents S, i.e. $S_R = S$.

Proof. By Theorem 2.9 we prove that for each $a \in \Omega$:

$$F_S(\{a\}) = \bigcap_{a \in E_{ij}} E_{ij} \quad \text{if } \exists\, E_{ij} : a \in E_{ij} ,$$

$$F_S(\{a\}) = \Omega \text{ otherwise.}$$

It is easy to see that if $F_S(\{a\})=\Omega$, then there is not an E_{ij} so that $a\in E_{ij}$ (by the construction of R). If $F(\{a_{i_t}\})=B_t$, then by (ii) in Definition of strong operation we have $\forall B_k: a_{i_t}\in B_k$ implies $F_S(\{a_{i_t}\})=B_t \subseteq B_k$. Consequently,

$$F_S(\{a_{i_t}\}) = \bigcap_{a\in E_{i_t}} E_{i_t} = E_{o_t} = B_t.$$

For $a_{i_t}\in\Omega$: $F(\{a_{i_t}\}) = B_{j_1}\cap\ldots\cap B_{j_t}$. We obtain that for any $B_k (a_{i_t}\in B_k)$ $F(\{a_{i_t}\})\subseteq B_k$ by (ii). Consequently,

$$F(\{a_{i_t}\}) = \bigcap_{a\in E_{ij}} E_{ij} = \bigcap_{q=1}^{t} E_{o j_q} .$$

It can be seen that by (ii) for any a $(a\in\Omega)$ so that $F(\{a\}) = F(\{a_{i_t}\})$. We have

$$F(\{a\}) = \bigcap_{a\in E_{ij}} E_{ij} = E_{o_t} \quad \text{or} \quad F(\{a\}) = \bigcap_{a\in E_{ij}} E_{ij} = \bigcap_{q=1}^{t} E_{o j_q}$$

The proof is complete.

Based on Proposition 2.17, if can be seen that if F is a strong operation over Ω, then there is an effective algorithm (this algorithm requires time polynomial in $/\Omega/$), which determines a relation (it is analogous to R in Proposition 2.17) or that this relation represents F.

Definition 2.18. Let S be an s-family over Ω. Let $Q(S)=\min\{m: S_R=S, /R/=m, R \text{ is a relation over } \Omega\}$. Thus, Q(S) is the number of rows of minimal relation which represents the s-family S.

Corollary 2.19. Let S be an s-family over Ω.
Let

$T=\{F_S(\{a\}): a\in\Omega\}$ and $N=\{A\in T: A\neq\Omega, (\forall B,C\in T)(A=B\cap C\rightarrow A=B \text{ or } A=C)\}$.

Then if $/N/=0$ i.e. $T=\{\Omega\}$, then $Q(S)=2$. If $/N/\geq 1$, then $\sqrt{2\log_2 /N/} < Q(S) \leq /N/+1 \leq /\Omega/+1$.

Proof. According Theorem 2.9 if R represents S, then

$$F_S(\{a\}) = \bigcap_{a\in E_{ij}} E_{ij} \quad \text{if } \exists E_{ij}: a\in E_{ij} \;,$$

$$F_S(\{A\}) = \Omega \text{ otherwise} .$$

It is easy to see that if $T=\{\Omega\}$, then for all E_{ij} we obtain $E_{ij}=\emptyset$. Consequently, $Q(S)=2$.

It is clear that according to the definition of strong operation N determines the family $\{F_S(\{a\}): a\in\Omega\}$. It is obvious that $/\Omega/\geq/N/$ and for $\forall A_i, A_j \in N$ $(A_i\neq A_j)$ we have

$$\{(i,j): 1\leq i<j\leq m,\ /R/ = m,\ A_i \subsetneqq E_{ij}\} \neq$$

$$\{(i,j): 1\leq i<j\leq m,\ /R/ = m,\ A_j \subsetneqq E_{ij}\} \quad .$$

Consequently, $/N/\leq 2^{\binom{m}{2}}$. Thus, $\sqrt{2\log_2/N/}<m$. By Proposition 2.17 $Q(S)\leq/N/+1$. The proof is proved.

The next corollary is obvious.

Corollary 2.20. Let $Q(n)=\max\ \{Q(S): S$ is an s-family over Ω, $/\Omega/=n\}$. Then $Q(n)\leq n+1$.

Definition 2.21. Let

$T\subseteqq P(\Omega)$ and let $N=\{A\in T: A\neq\Omega, (\forall B,C\in T)(A=B\cap C\rightarrow A=B \text{ or } A=C)\}$.

Then we say that T is s-semilattice if T closed under intersection, $\Omega\in T$ and N satisfies (1): for all $A\in N$

$$(\exists a\in A)(A_i\in N \text{ and } A\not\subset A_i \rightarrow a\notin A_i) \ .$$

Theorem 2.22. Let F be a strong operation over Ω. Let

$T_F= \{F(A): A\in P(\Omega)\}$ and

$N_F=\{A\in T_F: A\neq\Omega,\ (\forall B,\ C\in T_F)\ (A=B\cap C \rightarrow A=B \text{ or } A=C)\}$.

Then T_F is a s-semilattice. Conversely, if T is any s-semilattice, then there is exactly one strong operation so that $T=T_F$. Where for each a $(a\in\Omega)$

$$F(\{a\}) = \bigcap_{\substack{a\in A_i \\ A_i\in N}} A_i \qquad \text{if } \exists A_i\in N : a\in A_i\ ,$$
$$F(\{a\}) = \Omega \qquad \text{otherwise} .$$

Proof. It is obvious that for arbitrary strong operation we have $\forall A,B\in P(\Omega): F(A\cup B)= F(A)\cap F(B)$, $F(\emptyset)=\Omega$ and $A\subseteq B\to F(B)\subseteq F(A)$. Consequently $\Omega\in T_F$ and T_F closed under intersection. Now we assume that $A\in N_F$. If there is not attribute a so that $F(\{a\})=A$, then if $A=F(B)$ $(/B/\geq 2)$, then $A = \bigcap_{b_i\in B} F(\{b_i\})$. This contradicts the definition of N_F. Consequently, there is attribute a $(a\in\Omega)$ so that $F(\{a\})=A$. It is obvious that $a\in A$. Clearly, N_F satisfies (1).

Conversely, we assume that T is a s-semilattice over Ω. Then we set for each a $(a\in\Omega)$

$$F(\{a\}) = \bigcap_{\substack{b\in A_i \\ A_i\in N}} A_i \qquad \text{if } \exists A_i\in N : a\in A_i\ ,$$
$$F(\{a\}) = \Omega \qquad \text{otherwise.}$$

Clearly, for all $A\in N$ $(\exists a\in A: A_i\in N \text{ and } A\not\subset A_i \to a\notin A_i)$ we obtain $F(\{a\})=A$ for different set $A(A\in N)$ is easy to see that there is attribute a so that $F(\{a\})=A$. Consequently, $\forall A\in N$: $a\in\Omega$: $F(\{a\})=A$. Now we prove that F is a strong operation. According to the construction of F it is clear that $a\in F(\{a\})$ and if there is $A_i\in N$ so that $a\in A_i$, then $F(\{a\})\in N^+$. If $b\in F(\{a\})$, then

$$F(\{b\}) = \bigcap_{\substack{a\in A_i \\ A_i\in N}} A_i \subseteq \bigcap_{\substack{a\in A_i \\ A_i\in N}} A_i = F(\{a\}).$$

It is obvious that the set $\{F(\{a\}): b\in\Omega\}$ determines the set $\{F(A): A\in P(\Omega)\}$. Consequently, F is a strong operation and $T=T_F$. If we suppose that there is strong operation F′ so that $T=T_{F'}$. Then for all a $(a\in\Omega)$ there is b $(b\in\Omega)$ such that $F(\{a\})=F'(\{b\})$. It is obvious that $a\in F'(\{b\})$. Consequently, we have $F'(\{a\}) \subseteq F(\{a\})$. On the other hand, there is attribute c $(c\in\Omega)$ so that $F'(\{a\})=F(\{c\})$. By $a\in F(\{c\})$ we obtain

$F(\{a\}) \subseteq F'(\{a\})$.

The proof is proved.

Definition 2.23. Let S be a s-family over Ω, and $(A,B) \in S$. We say that (A,B) is a maximal right side dependency of S if $\forall B': B \subseteq B'$, $(A,B') \in S \rightarrow B=B'$.

Denote by M(S) the set of all maximal right side dependencies of S. We say that $B(B\ \Omega)$ is a maximal right side of S if there is an A such that $(A,B) \in M(S)$. Denote U(S) the set of all maximal right sides of S.

Corollary 2.24. Let S be a s-family over Ω. Then U(S) is a s-semilattice. Conversely, if T is a s-semilattice, then there is exactly one s-family so that $T=U(S)$. By Lemma 2.5 and Theorem 2.22 this corollary is obvious.

ACKNOWLEDGEMENTS

The author would like to take this opportunity to express deep gratitude to Professor Dr.J.Demetrovics for his hel, valuable comments and suggestions.

REFERENCES

[1] Czédli, G.: Függőségek relációs adatbázis modellben. Alkalmazott Matematikai Lapok 6 (1980) 131-143.

[2] Demetrovics, J.: Relációs adatmodell logikai és strukturális vizsgálata. MTA-SZTAKI Tanulmányok, Budapest, 114 (1980) 1-97.

Az erős operációkról

Vu Duc Thi

Összefoglaló

A szerző az erős operációk tulajdonságait vizsgálja. Néhány uj kombinatorikus eredményt is ad, amelyek az erős függőségek családjaira vonatkoznak.

Сильные операции

Ву Дык Тхи

Р е з ю м е

В настоящей работе изучается связь между сильными операциями и сильными зависимостями.

MTA SZTAKI Közlemények 34/1986 pp.105-111

SUR L'EXUSTENCE DES MESURES CONTINUES INVARIANTS POUR TRANSFORMATIONS DE L'INTERVALLE

Nguyen Cong Thanh

Institut de Recherach
Météorologique et Hydrologique,
Hanoi, Vietnam

La notion d'orbites strictement turbulentes est en relation avec celle de trajectoires turbulentes de Ruelle-Takens. Dans [1] Lasota-Yorke ont étudié l'existence des mesures continues invariantes pour une transformation avec les orbites strictement turbulentes. Ici, on expose les résultats de Lasota-Yorke, les applique dans le cas d'une dimension pour obtenir des résultats plus généraux.

1. DÉFINITIONS ET RÉSULTATS

Soient X un espace topologique de Hausdorff et T une application continue (transformation) de X dans lui-même.Pour chaque orbite $\Theta(x_o)$ du point x_o l'ensemble

$$L(\Theta) = \bigcap_{n\geq 0} \overline{(T^n\Theta)}, \text{ ou } \overline{(T^n\Theta)} \text{ est fermeture de } T^n\Theta$$

est dit l'ensemble limit de Θ. Pour insister le rôle de la transformation T on écrit quelque fois $\Theta_T(x)$ et $L_T(\Theta_T)$. L'orbite Θ_T s'appele strictement turbulentes (plus précisement, strictement turbulente par rapport à T) si $L_T(\Theta_T)$ est un ensemble compact non-vide qui ne contient pas de points périodiques.

Une mesure m sur X est invariante pour T si

$$m(A) = m(T^{-1}A) \quad \text{pour tout } A \in \mathcal{B}$$

où $\mathcal{B}$ σ-algèbre de Borel de X.

La mesure m est dite continue si $m(x)=0$ pour tout $x \in X$.

Proposition 1

Soient $T: X \to X$ une application continue et $S=T^p$ avec un certain entier p. Pour chaque $x \in X$ deux affirmations suivantes sont equivalentes

a) $\Theta_T(x)$ est strictement turbulente par rapport à T,

b) $\Theta_S(x)$ est strictement turbulente par rapport à S.

La démonstration est immediate des définitions.

Proposition 2

Soient $T: X \to X$ une transformation et $\Theta(x)$ une orbite strictement turbulente. Alors il existe une mesure continue concentrée sur $L(\Theta)$ et invariante pour T.

La démonstration de la Proposition 2 est simple (Voir [1]).

La résultat principal de [1] est la Théorème suivant qui donne une condition simple assurant l'existence des orbites strictement turbulentes.

Theorem 1 [1]

Soit $T: X \to X$ une transformation. Suppose qu'il existe deux sousensembles copacts, non-vide, disjoints A et B de X tels que

$$T(A) \cap T(B) \supset A \cup B$$

Alors pour T il existe des orbites strictement turbulentes et en consequence des mesures continues invariantes pour T.

2. LE CAS D'UNE DIMENSION

Dans le cas $X=[\alpha,\beta]$, en appliquant Théorème 1 Lasota-Yorke ont démontré le résultat suivant

Théorèm 2

Soit f une transformation de l'intervalle $[\alpha,\beta]$ dans lui-même. Alors l'existence des points periodiques de periode 3N, pour un certain entier N, implique l'existence des mesures continues invariantes pour f.

Le théorème ci-dessous est une généralisation du Théorème 2.

Théorèm 3

Soit f une transformation de l'intervalle fermé dans lui-même. Suppose que f ait des points périodiques de période differente de puissances de 2. Alors il existe des orbites strictement turbulentes, et en consequence, des mesures continues invariantes pour f.

Démonstration

Suppose f ait des points périodiques de période $n\cdot 2^k$, où n est un entier impair. En posant $T=f^{2^k}$ on obtient l'application continue T avec des points périodiques de période n.

Soient Θ une orbite périodique de période n de T et

$$K = [\min\Theta , \max\Theta]$$

Puisque $T(\max\Theta)< \max\Theta$ et $T(\min\Theta)> \Theta$. On peut poser

$$b = \min\{x\in\Theta \text{ tel que } T(x)<x\}$$

et il est facile de voir que $b\neq\min\Theta$.

On prend a comme le point de Θ le plus voisin de b à gauche. Ainssi on obtient

$$T(b) \leq a \quad \text{et} \quad T(a) \geq b \tag{1}$$

et pose

$$A = [a,b]$$

En tenant compte de ce que deux égalités dans (1) ne peuvent avoir lieu simultanement, on voit que $T(A) \supset A$ et $T(A) \neq A$. De cela il résulte que la suite $\{T^i(A)\}$ $i=1,2,...$ est croissante et ensuite $T^n(A) \supset K$.

De l'imparité de n on deduit que les nombres de points de à deux côtés de A ne sont pas égaux. Suppose que les points de Θ d'après l'order naturel sur la ligne réelle soient

$$\Theta = \{x_1,...,x_k,a,b,y_1,...,y_m\}$$

ou $k+m+2=n$.

Pose

$$G = \{x_1...x_k\} \text{ et } D = \{y_1...y_m\}$$

Sans limiter la généralité on peut supposer $m>k$. Puisque $T(b) \leq a$ sous l'action de T l'une des images de y_i doit rester encore à droite de A. C'est à dire il existe un point $c \in D$ tel que $T(c) \geq b$.

Si $T(b)<a$ d'après la continuité de T il existe un point $d \in (b,c)$ tel que $T(c)=a$. En posant $B=[d,c]$ on a $T(B) \supset A$.

Si $T(b)=a$ on considère l'emsemble $E=D \cup \{a,b\}$. Alors

$$T(E) \cap G \neq \emptyset$$

S'il ne l'était pas, l'ensemble E serait invariant sous l'action de T et des points de E seraient de période plus petite que n. C'est une contradiction.

Ainsi il existe un point $y_i \in D$ tel que $T(y_i)<a$. En posant $B=[y_i,d]$ (ou $B=[d,y_i]$) on obtient aussi $T(B) \supset A$.

Donc dans tous les cas il y a toujours un intervalle fermé B tel que

$$B \cap A = \emptyset \text{ et } T(B) \supset A.$$

Vu que

$$T^n(A) \supset K \quad \text{et} \quad T(B) \supset A \quad \text{on a}$$

$$T^{n+1}(A) \supset K \quad \text{et} \quad T^{n+1}(B) = T^n(t(B)) \supset T^n(A) \supset K$$

En somme

$$T^{n+1}(A) \cap T^{n+1}(B) \supset A \cup B.$$

En notant $F=f^p$ avec $p=(n+1)\cdot 2^k$ on voit que l'application F vérifie les conditions du Théorème 1. Donc pour F il exite des orbites strictement turbulentes. Et grâce à la Proposition 2 il existe des mesures continues invariantes pour f.

C.Q.F.D.

Remarque

Les applications continues satisfaisant aux conditions du Théorème 3 ont le compertement chaotique d'après le sens de Li-Yorke. Donc l'existence des mesures continues invariantes est un caractère du phénomène chaotique dans le cas d'une dimension (Voir [2]).

REFERENCES

[1] Lasota, A. - Yorke, J.A.: On the existence of invariant measures for transformations with strictly turbulent trajectories. Bull.Acad.Polon.Scien. XXV.No.31 (1977)

[2] Nguyen Cong Thanh: Dissertation, Budapest, 1985.

Az intervallum leképzések folytonos invariáns mértékeinek létezéséről

Nguyen Cong Thanh

Összefoglaló

A dolgozat fő eredménye a következő tétel: Legyen X egy Hausdorff tér, és $T: X \to X$ egy folytonos leképzés; X-nek létezik két kompakt diszjunkt, nem üres részhalmaza - A és B - ugy, hogy $T(A) \cap T(B) \supset A \cup B$, akkor T-nek létezik folytonos invariáns mértéke.

Következmény: ha f egy zárt intervallum önmagába való leképzése és f-nek van nem 2-hatvány periódusu periódikus pontja, akkor f-nek van folytonos invariáns mértéke.

***

On the continuous invariant measures of intervall mappings

Nguyen Cong Thanh

Summary

The main result of this paper is the following: let X be a Hausdorff space and $T: X \to X$ a continuous mapping; if there exist two disjoint compact subsets - A and B - of X such that $T(A) \cap T(B) \supset A \cup B$, then T has a continuous invariant measure.

If f is a continuous mapping of a closed intervall having periodical point with period $k \neq 2^h$, then f has continuous invariant measure.

О непрерывных инвариантных мерах отображения единичного интервала

Нгуен Конг Тхан

Р е з ю м е

Главный результат рабоыт заключается в следующем: пусть X хаусдорфово пространство и $T:X\to X$ непрерывное отображение; если существуют непересекающиеся компактные подмножества A и B, такие что $T(A)\cap T(B)\supset A\cup B$, то тогда T имеет непрерывную инвариантную меру.

Следствие: если f отображение замкнутого интервала и оно имеет периодическую точку периода $k\neq 2^k$, то тогда f имеет непрерывную инвариантную меру.

MTA SZTAKI Közlemények 34/1986 pp.113-122

REMARKS ON DUAL DEPENDENCIES

Vu Duc Thi

MTA SZTAKI

§0. INTRODUCTION

One of the main concepts in relational database theory is the full family of functional dependencies, that was first axiomatized by W.W.Armstrong [1].

The full family of dual dependencies have also been introduced and axiomatized [2,3].

In this paper, we give some results, that are related to dual dependencies. Some properties of functional dependencies are investigated also.

§1. DEFINITIONS

In this section, we present some necessary definitions.

Definition 1.1. Let Ω be a finite set of attributes, and $R=\{h_1,\dots,h_m\}$ be a relation over Ω, $A,B\subseteq\Omega$. Then we say that B dually depends A in R (denote $A \underset{R}{\overset{d}{\rightarrow}} B$) if

$$(\forall h_i,h_j\in R)((\exists a\in A)(h_i(a)=h_j(a))\rightarrow(\exists b\in B)(h_i(b)=h_j(b)));$$

B functionally depends A in R (denote $A \underset{R}{\overset{f}{\rightarrow}} B$) if

$$(\forall h_i,h_j\in R)((\forall a\in A)(h_i(a)=h_j(a))\rightarrow(\forall b\in B)(h_i(b)=h_j(b))).$$

Let

$$D_R=\{(A,B) : A \underset{R}{\overset{d}{\rightarrow}} B\} \quad\text{and}\quad F_R=\{(A,B) : A \underset{R}{\overset{f}{\rightarrow}} B\} .$$

$D_R(F_R)$ is called the full family of dual (functional) dependencies.

Definition 1.2. Let R be a relation over Ω, and $A \subseteq \Omega$. A is a key of R if $A \xrightarrow[R]{f} \Omega$. The key A is a minimal key of R if for any A' $(A' \subseteq A)$: $A' \xrightarrow[R]{f} \Omega$ implies $A' = A$.

Denote by K_R the set of all minimal keys of R. It is clear that K_R forms a Sperner-system over Ω.

Definition 1.3. Let K be a Sperner-system over Ω. We define the set of antikeys of K, denoted by K^{-1}, as follows:

$$K^{-1} = \{A \subsetneq \Omega : (B \in K) \rightarrow (B \nsubseteq A) \text{ and } (A \subsetneq C) \rightarrow (\exists B \in K)(B \subseteq C)\} .$$

Clearly, K^{-1} is also a Sperner-system over Ω. K_R^{-1} is called the set of antikeys of relation R.

Definition 1.4. Let Ω be a finite set, and denote $P(\Omega)$ its power set. Let $Y \subseteq P(\Omega) \times P(\Omega)$. Then we say that Y satisfies the F-axioms, if for all $A,B,C,D \subseteq \Omega$.

(F_1) $(A,A) \in Y$;

(F_2) $(A,B) \in Y,\ (B,C) \in Y \rightarrow (A,C) \in Y$;

(F_3) $(A,B) \in Y,\ A \subseteq C,\ D \subseteq B \rightarrow (C,D) \in Y$;

(F_4) $(A,B) \in Y,\ (C,D) \in Y \rightarrow (A \cup C, B \cup D) \in Y$.

Y satisfies the D-axioms, if for all $A,B,C,D \subseteq \Omega$.

(D_1) $(A,A) \in Y$;

(D_2) $(A,B) \in Y,\ (B,C) \in Y \rightarrow (A,C) \in Y$;

(D_3) $(A,B) \in Y,\ C \subseteq A,\ B \subseteq D \rightarrow (C,D) \in Y$;

(D_4) $(A,B) \in Y,\ (C,D) \in Y \rightarrow (A \cup C, B \cup D) \in Y$;

(D_5) $(A,\emptyset) \in Y \rightarrow A = \emptyset$.

Definition 1.5. Let $Y \subseteq P(\Omega) \times P(\Omega)$. We say that Y is a d-(f-) family over Ω if Y satisfies the D-(F-) axioms.

§2. THE FAMILIES OF DUAL DEPENDENCIES

Definition 2.1. Let D be a d-family, and R be a relation over Ω. Then we say that R reprezents D iff $D_R = D$.

Definition 2.2. Let D be a d-family over Ω, and $(A,B) \in D$. We say that (A,B) is a maximal left side dependency of D if $\forall A' : A \subsetneq A',\ (A',B) \in D \rightarrow A' = A$.

Denote by M(D) the set of all maximal left side dependencies of D. We say that A is a maximal left side of D if there is a B so that $(A,B) \in M(D)$. Denote G(D) the set of all maximal left sides of D. A family G of subsets of Ω is called d-semilattice iff G contains $\emptyset, \Omega$, and $A,B \in G$ imply $A \cap B \in G$.

In paper [2], the next theorem is proved.

Theorem 2.3. [2]. Let D be a d-family over Ω. Then G(D) is a d-semilattice over Ω. Conversely, if G is any d-semilattice, then there exist exactly one d-family D so that G(D)=G, where

$$D = \{(A,B) : \forall C \in G : A \nsubseteq C \rightarrow B \nsubseteq C\} \quad .$$

Definition 2.4. Let $I \subseteq P(\Omega)$, and I is closed under intersection, i.e. $A,B \in I \rightarrow A \cap B \in I$. Let $M \subseteq P(\Omega)$. Denote M^+ the set $\{\cap M' : M' \subseteq M\}$. We say that M generates I iff $M^+ = I$.

Definition 2.5. Let $R = \{h_1, \ldots, h_m\}$ be a relation over Ω. Let $N_{ij} = \{a \in\ : h_i(a) \neq h_j(a),\ 1 \leq i < j \leq m\}$. We call N_{ij} the non-equality set of R. Denote by N the family of all non-eqiality sets of R.

Practically, it is possible that $\Omega \in N$, there are some N_{ij} , which are equal to each other. According to the definition of relation. We obtain $\emptyset \notin N$.

We assume that

$$N = \{N_{ij} : 1 \leq i < j \leq m\} = \{A_1, \ldots, A_k : A_i \neq A_j \text{ for } i \neq j\ ;\ i,j = 1, \ldots, k\}$$

We set $S = \{A_1, \ldots, A_k\}$.

Based on the non-equality sets of R we give a necessary and sufficient condition for $D_R = D$. We can consider the set of non-equality sets as a characterization of dual dependencies in relations.

Theorem 2.6. Let D be a d-family, and R be a relation over Ω. Then R represents D if and only if $G(D)=(S\backslash\Omega)^{+}\cup\{\emptyset\}$.

Proof. By Theorem 2.3, it is easy to see that R represents D iff $G(D_R)=G(D)$. Consequently, we only must prove that $G(D_R)=$ $=(S\backslash\Omega)^{+}\cup\{\emptyset\}$. Clearly, $G(D_R)$ is a d-semilattice over Ω. It is obvious that $G(D_R)$ contains $\emptyset,\Omega$, and $(S\backslash\Omega)^{+}$ contains Ω (by convention $\cap\emptyset=\Omega$). Now, we suppose that $N_{ij}\neq\Omega$ (it is obvious that $N_{ij}\neq\emptyset$). Because for any $a\in\Omega\backslash N_{ij}$ we obtain $h_i(a)=h_j(a)$, but $\forall b\in N_{ij}: h_i(b)\neq h_j(b)$, $aUN_{ij} \overset{d}{\underset{R}{\not\to}} N_{ij}$. Hence $N_{ij}\in G(D_R)$ holds. Consequently, $S\subseteq G(D_R)$ holds, i.e. we have $(S\backslash\Omega)^{+}\cup\{\emptyset\}\subseteq G(D_R)$. Conversely, if $A\in G(D_R)\backslash\{\emptyset,\Omega\}$, then if we assume that $\forall h_i,h_j\in R:$ $a\in A$ so that $h_i(a)=h_j(a)$. So $\Omega \overset{d}{\underset{R}{\to}} A$, this contradicts to the definition of A. Consequently, there is an index pair (i,j) so that $A\subseteq N_{ij}$. We set $H=\{N_{ij}: A\subseteq N_{ij}\}$. If there is a N_{ij} such that $A=N_{ij}$, then $A\in S$. If

$$A\notin \bigcap_{N_{ij}\in H} N_{ij} \quad ,$$

then for all $N_{ij}\notin H$ we have $A\notin N_{ij}$. So

$$\bigcap_{N_{ij}\in H} N_{ij} \overset{d}{\underset{R}{\to}} A$$

holds. This contradicts $A\in G(D_R)\backslash\{\emptyset,\Omega\}$. Consequently, we have

$$A=\bigcap_{N_{ij}\in H} N_{ij} \ .$$

Thus according to the definition S we obtain $A\in S^{+}$. So $G(D_R)=(S\backslash\Omega)^{+}\cup\{\emptyset\}$ holds. The theorem is proved.

The next proposition shows that from given any d-family D, we can construct one simple non-empty relation R such that $D_R=D$.

Proposition 2.7. Let D be a d-family over Ω, $G(D)$ be a set of all maximal left sides of D, and let

$$N_2 = \{A \in G(D) : A \neq \Omega, (\forall B, C \in G(D))(A = B \cap C \to A = B \text{ or } A = C)\}$$

Then we assume that $N_2 = \{A_1, \ldots, A_k\}$, we set $R = \{h_1, h_2, \ldots, h_{2h-1}, h_{2h}\}$ as follows:

$$\text{for } i=1,\ldots,k : \forall a \in \Omega \; h_{2i-1}(a) = 2i-1$$

$$h_{2i}(a) = \begin{cases} 2i-1 & \text{if } a \in \Omega - A_i \,, \\ 2i & \text{other wise.} \end{cases}$$

Then R represents D.

Proof. It is obvious that $|N_2| \geq 1$. Consequently, $R \neq \emptyset$ holds. Clearly, $N = \{N_{ij} : N_{ij}$ is a non-equality set of $R\} = N_2 \cup \{\Omega\}$. On the other hand $N_2^+ = G(D)$. By Theorem 2.6 we obtain $D_R = D$. The proposition is proved.

Remark 2.8. Let R be a relation over Ω.
Let $F_R' = F_R - \{(\emptyset, A) \in F_R : A \neq O\}$, $D(F_R) = \{(A,B) : (B,A) \in F_R'\}$. The paper [3] showed that F_R' $(D(F_R))$ satisfies the F_-(D_-) axioms.

If $F_R' \neq F_R$ holds, i.e. $\{(\emptyset, A) \in F_R : A \neq \emptyset\} \neq \emptyset$, then we assume that $R = \{h_1, \ldots, h_m\}$, let $R' = \{h_1, \ldots, h_m, h_{m+1}\}$, where for all $a \in \Omega$, and all $i (1 \leq i \leq m) : h_{m+1}(a) \neq h_i(a)$. It is obvious that $F_{R'} = F_R'$, and $D_R = D_{R'}$.

It is known that usually $D(F_R) \neq D_R$. We give a necessary and sufficient condition for $D(F_R) = D_R$.

First we investigate the f-families.

Definition 2.9. Let F be a f-family over Ω, and $(A,B) \in F$. Then we say that (A,B) is a maximal right side dependency of F if $\forall B' : B \subseteqq B'$, $(A,B') \in F \to B' = B$.

Denote by $M(F)$ the set of maximal right side dependencies of F. We say that B $(B \subseteqq \Omega)$ is a maximal right side of F iff there is an A so that $(A,B) \in M(F)$. Denote $I(F)$ the set of maximal right sides of F. Thus, $I(F) : \{B : \exists A : (A,B) \in M(F)\}$.

W.W.Armstrong in [1] proved an important following theorem.

Theorem 2.10. [1]. Let F be a f-family over Ω. Then $I(F)$ is closed under intersection. Conversely, if I is any family of subsets of Ω, which is closed under intersection, then there exists exactly one f-family F such that $I(F)=I$, where $F=\{(A,B): \forall C\in I: A\subseteq C \rightarrow B\subseteq C\}$.

By convention $\cap\emptyset=\Omega$ we have that if I is closed under intersection, then $\Omega\in I$ holds.

In [3] it is showed that for given family M of subsets of Ω there is exactly one family N, which generates M^+, and has minimal cardinality.

Lemma 2.11. [3] Let $M\subseteq P(\Omega)$ be a family over Ω. Let $N=\{A\in M: \forall B,C\in M: A=B\cap C \rightarrow A=B \text{ or } A=C\}$. Then N generates M^+ and if N' generates M^+, then $N=N'$.

(It is possible $\emptyset\in N$ and by convention $\emptyset=\Omega$ we have that Ω is never required in M.)

Definition 2.12. In [3] the equality sets of the relation are defined as follows: Let $R=\{h_1,\ldots,h_m\}$ be a relation over Ω. for $1\leq i<j\leq m$, we denote by E_{ij} the set $\{a\in\Omega: h_i(a)=h_j(a)\}$. We set $E=\{E_{ij}: 1\leq i<j\leq m\}$.

Practically, it is possible that there are some E_{ij}, which are equal to each other.

Let $T=\{E_{ij}: \text{if } E_{pq},E_{st}\in T, \text{ then } E_{pq}\neq E_{st}\}=$

$$=\{B_1,\ldots,B_\ell : B_i\neq B_j \text{ for } i\neq j\} .$$

It is obvious that ℓ is the number of elements of T, and all elements of T are not equal to each other. It is obvious that $E_{ij}\neq\Omega$.

Lemma 2.13. Let $R=\{h_1,\ldots,h_m\}$ be a relation over Ω. Then $T^+=I(F_R)$.

Proof. It is clear that $I(F_R)$ is closed under intersection, and $E_{ij}\in I(F_R)$, where $1\leq i<j\leq m$. Consequently, $T^+=I(F_R)$. Conversely, it is obvious that $\Omega\in T^+$ and $\Omega\in I(F_R)$. If $A\in I(F_R)\setminus\{\Omega\}$, then there is a E_{ij} $(E_{ij}\in T)$ such that $A\subseteq E_{ij}$. We

set $S=\{E_{ij}\in T\colon A\subseteq E_{ij}\}$. If there is an E_{ij} ($E_{ij}\in T$) so that $E_{ij}=A$, then $A\in T^{\ddagger}$ holds. If $A\subsetneqq \bigcap_{E_{ij}\in S} E_{ij}$, then for all inderpairs (p,q) such that $E_{pq}\in S$ or there is an E_{ij} ($E_{ij}\in S$) so that $E_{pq}=E_{ij}$ we obtain that:

$$(\forall a\in A)(h_p(a)=h_q(a)) \rightarrow (\forall b\in \bigcap_{E_{ij}\in S} E_{ij})(h_p(b)=h_q(b)) .$$

On the other hand, for all other inderpairs (p,q) (if exist) there is an $a\in A$ such that $h_p(a)\neq h_q(a)$. Consequently,

$$A \xrightarrow[R]{f} \bigcap_{E_{ij}\in S} E_{ij}$$

holds. This contradicts to the definition of A. Thus, we have

$$A = \bigcap_{E_{ij}\in S} E_{ij} .$$

Definition 2.14. Let R be a relation, and F be a f-family over Ω. We say that R represents F iff $F_R=F$.

Based on Theorem 2.10 and Lemma 2.13 a following corollary is obvious.

Corollary 2.15. Let R be a relation, and F be a f-family over Ω. Then R represents F if and only if $T^{+}=I(F)$. Now, we give a necessary and sufficient condition for $D(F_R)=D_R$.

Theorem 2.16. Let R be a relation over Ω.
Let

$N_1=\{A\in E\colon A\neq\emptyset,\ (\forall B,C\in E)(A=B\cap C\rightarrow A=B \text{ or } A=C)$
E is the set ofall equality sets of R$\}$.

and

$N_2= \{D\in N\colon D\neq\Omega\ (\forall X,Y\in N)(D=X\cap Y\rightarrow D=X \text{ or } D=Y)$,
N is the set of all non-equality sets of R$\}$.

Thus, N_2 is the unique family, which has minimal cardinality, and generates N.

Then $D(F_R)=D_R$ holds if and only if $N_2=N_2$ holds.

Proof. By the definition of $D(F_R)$ we have $I(F'_R)=G(D(F_R))$, and $\emptyset \in I(F'_R)$. By Theorem 2.3 we obtain that $D(F_R)= D_R$ holds if and only if $G(D_R)=G(D(F_R))$ holds. It is obvious that $\emptyset \in N$. From Theorem 2.6 we have $G(D_R)=N_2^+ \cup \{\emptyset\}$.

By Lemma 2.13 we obtain that $I(F_R) \cup \{\emptyset\}=I(F'_R)$ and $N_1^+ \cup \{\emptyset\}=I(F'_R)$ holds. Consequently, $N_1=N_2$ holds iff $G(D_R)= = G(D(F_R))$ holds. The proof is complete.

It can be seen that from E, and N we can construct the N_1 and N_2. Consequently, the next corollary is obvious.

Corollary 2.17. Let R be a relation over Ω. Then there is an effective algorithm, that decide whe there $D_R=D(F_R)$ holds or not.

It is clear that the time complexity of this algorithm is polynomial in $|R|$, and $|\Omega|$.

Definition 2.18. Let $R=\{h_1,\dots,h_m\}$ be a relation over Ω, and E be a family of all equality sets of R.

Let

$$T=\{A \in P(\Omega):\ \exists E_{ij} \in E:\ A=E_{ij}\},\ \text{and}\quad M=\{A \in T:\ \bar{\exists} B \in T:\ A \subsetneq B\}$$

M is called the family of maximal equality sets of R.

Theorem 2.19. Let R be a relation over Ω, and K_R^{-1} be a set of antikeys of R. Then $K_R^{-1}=M$.

Proof. It is obvious that F_R is a f-family. Now, we suppose that A is an antikey of R. It is clear that $A \neq \Omega$. If there exists exists a B such that $A \subsetneq B$, and $A \xrightarrow[R]{f} B$, then by the definition of antikey $B \xrightarrow[R]{f} \Omega$. Hence $A \xrightarrow[R]{f} \Omega$ holds. This contradicts $\forall C \in K_R$: $C \nsubseteq A$. So we have $A \in I(F_R)$. If there is a B' such that $B' \in I(F_R)\setminus\Omega$, and $A \subset B'$, then B' is a key. This contradicts $B' \neq \Omega$. Consequently, $A \in I(F_R)\setminus\Omega$ and $\bar{\exists} B'(B' \in I(F_R)\setminus\Omega): A \subset B'$. On the other hand, according to the definition of relation $\Omega \neq M$. It is easy to see that $E_{ij} \in I(F_R)$. Thus, $M \subseteq I(F_R)$. If D is a set such that $\forall E_{ij} \in M: D \nsubseteq E_{ij}$, then by the definition of functional dependency we have $D \xrightarrow[R]{f} \Omega$, i.e. D is a key. Consequently, M is the set of distinct of $I(F_R)$. So we have $A \in M$.

Conversely, we assume that $A \in M$. Then according to the definition of relation and M we have $A \not\xrightarrow[R]{f} \Omega$, i.e. $\forall B \in K_R : B \nsubseteq A$. On the other hand, because A is a maximal equality set, we obtain that if $A \subset D$ then $D \xrightarrow[R]{f} \Omega$. Consequently, by the definition of antikey $A \in K_R^{-1}$. The theorem is proved.

Based on Theorem 2.19 the next corollary is obvious.

Corollary 2.20. Let K be a non-empty Sperner-system over Ω, and let R be a relation. Then a necessary and sufficient condition for $K=K_R$ is that $K^{-1}=M$.

ACKNOWLEDGEMENT

The author would like to express deep gratitude to Professor J.Demetrovics for his help, valuable comments and suggestions.

REFERENCES

[1] W.W.Armstrong; Dependency Structures of Data Base Relationships, Information Processing 74, North-Holland Publ.Co. (1974) 580-583.

[2] G.Czédli; Függőségek relációs adatbázis modellben, Alkalmazott Matematikai Lapok 6 (1980) 132-143.

[3] J.Demetrovics; Relációs adatmodell logikai és strukturális vizsgálata, MTA-SZTAKI Tanulmányok, Budapest, 114 (1980) 1-97.

Megjegyzések duális függőségekről

Vu Duc Thi

Összefoglaló

Ebben a cikkben néhány uj eredményt adunk, amelyek duális függőségekre vonatkoznak. A funkcionális függőségek néhány tulajdonságait is vizsgáljuk.

Замечания об дуальных зависимостях

Ву Дык Тхи

Р е з ю м е

В настоящей работе изучаются дуальные зависимости. Рассматривается также связь между дуальными зависимостями и функциональными зависимостями.

MTA SZTAKI Közlemények 34/1986 pp.123-149

ЭКВИВАЛЕНТНОСТЬ ОДНОГО КЛАССА РЕЛЯЦИОННЫХ ВЫРАЖЕНИЙ

До Суан Тхо

В В Е Д Е Н И Е

На основе реляционной алгебры /relational algebra/ введенной доктором Коддом /1.3./, запросы использователей базы данных /БД/ могут быть выражены в виде реляционных выражений. Поэтому в области оптимизации обработки запросов в БД большое внимание уделялось вопросу представления реляционных выражений и создания эквивалентных преобразований реляционных выражений.

В /4/, /5/ было использовано табло /tableaux/ как двухверное представление одного класса запросов использователей соответствующего ограниченным реляционным выражениям /restricted relational expressions/. Ограниченным реляционным выражением называется выражение обладающее тремя операциями реляционной алгебры: выбор /Select/, проектирование /Project/ и соединение /Join/. /В /4/ его называют выражением SPJ./

Табло /обозначенное T/ - двухмерная матрица, каждый столбец которой соответствует одному из атрибутов универсального отношения в фиксированном порядке. Первая строка называется сводкой. Символы в табло выражены из:

1/ - Характерных переменных /distinguished variables/,

2/ - Нехарактерных переменных /nondistinguished variables/,

3/ - Констант,

4/ - Бланков.

Каждое ограниченное реляционное выражение может быть представлено в виде соответствующего табло. В /5/ были предложены методы эквивалентных преобразований любого табло к виду табло T_o с минимальным числом строк. Как известно, число строк в табло больше чем число операций соединения в ограниченных реляционных выражениях на единицу и, осуществление операции соединения в вычислительной машине требует много времени и объема памяти. Поэтому этим методам было уделено большое значение для оптимальной обработки реляционного вырежения. Несмотря на то, понятие табло написано в /4/, /5/ имеет некоторое ограничение. С его помощью лишь только представит в виде табло реляционное выражение обладающее операцией выбора вида $\sigma_{A=c}(r)$.

Практически нам приходилось обрабатывать наиболее сложные запросы. Рассмотрим следующие примеры /см. /6//.

Пусть в БД хранятся сведения о служащих и их учреждениях. Универсальное отношение R определено на атрибутах Н /номер служащего/, З /зарплата/, У /номер учреждения/, Ф /фамилия служащего/, Д /должность/, В /название учреждения/, А /адрес/, т.е.

R = R (Н,Ф,Д,З,У,В,А)

R1=R1 (Н,Ф,Д,З,У) - схема отношения СЛУЖАЩИЕ

R2=R2 (У,В,А) - схема отношения УЧРЕЖДЕНИЕ.

Первый запрос: получить номер, фамилию, должность тех служащих, которые работают в учреждении № 19 и имеют зарплату больше 1500. 1-й запрос представим в виде реляционного выражения:

$$\Pi_{Н,Ф,Д}(\sigma_{З>1500 \wedge У=19}(R1))$$

Второй вопрос: получить номер, фамилию, должность тех служащих, работающих в учреждениях № 17, 19, 32.

2-ой запрос представим в виде:

$$\Pi_{Н,Ф,Д}(\sigma_{У=17 \vee У=19 \vee У=32}(R1))$$

Легко видеть, что 1-й, 2-ой запросы не могут буть представлены в виде табло.

В этой работе мы введем новое понятие - обобщенное табло /обозначение τ /.

Введенное нами понятие τ позволяет развивать результаты опубликованные в /4/, /5/ для ограниченного реляционного выражения обладающего операцией выбора таких видов:

а/ $\sigma_{A\theta c}(r)$ б/ $G_{A\in}\{a_1,a_2,\ldots,a_k\}$

где A - атрибут, c - константа, $\theta \in \{=,\neq,<,\leq,>,\geq\}$, $a_i \in dom(A)$

Будем обозначать в общем виде $\sigma_{A\in\xi}$ для случаев а/ и б/, ξ - множество значений и $\xi \subseteq dom(A)$.

Работа состоит из 4 основных разделов. В пером разделе повторены основные понятия реляционной модели. Во втором разделе введены понятия обобщенного табло τ и правила построения табло соответствующего ограниченному реляционному выражению E с операцией выбора $\sigma_{A\in\xi}$. В третьем разделе будут описаны понятия эквивалентности и эквивалентное преобразование двух обобщенных табло. В четвертом разделе будем показывать связь между табло T и обобщенным табло τ .

1. Основные понятия реляционной модели

В реляционном БД объекты описываются при помощи характеризующих признаков, которые называются атрибутами. Каждый атрибут имеет имя и множество значений. Множество значений атрибута A называется его доменом и обозначается dom(A).

$R=\{A_1,A_2,\ldots,A_n\}$ - множество имен атрибутов отношения r. R называют схемой отношения r. Под понятием отношения понимают любое подмножество декартова произведения доменов, т.е.

$$r(A_1,\ldots,A_n) \subseteq dom(A_1) \times \ldots \times dom(A_n)$$

Элементы отношения $(a_1,\ldots,a_n)$, $a_i \in dom(A_i)$ является кортежем.

Значение атрибута A в кортежи μ обозначено $\mu[A]$.

Будем обозначать операцию выбора через $\sigma_{A\in\xi}$, A – атрибут схемы отношения r, $\xi \subseteq dom(A)$.

$$\sigma(r)_{A\in\xi} = \{\mu\in r \mid \mu[A] \in\xi\}.$$

Пусть $X \subseteq R$. Операция проектирования отношения r на X /обозначена $\Pi_X(r)$/ определяется следующим образом:

$$\Pi_X(r) = \{\mu[X] \mid \mu\in r\}.$$

Если отношения r_1 и r_2 имеются, схемы R_1, R_2 соответственно. Тогда операция естественного соединения /natural join/ отношения r_1 и r_2 обозначена $r_1 \times r_2$.

$$r_1 \times r_2 = \{\mu \mid /\mu \text{ - кортеж отношения со схемой } R_1 \text{ и } R_2/$$

$$\wedge(\exists\upsilon_1\in r_1 \wedge \upsilon_2\in r_2 : \upsilon_1[R_1] = \mu[R_1]\wedge\upsilon_2[R_2] = \mu[R_2])\}$$

Пусть отношения r_1 и r_2 имеют общую схему отношения. Будем обозначать операцию объединения двух отношений r_1 и r_2:

$$r_1 \cup r_2 = \{\mu \mid \mu\in r_1 \vee \mu\in r_2\}.$$

Операция пересечения отношений r_1 и r_2 обозначается $r_1 \cap r_2$

$$r_1 \cap r_2 = \{\mu \mid \mu\in r_1\wedge\mu\in r_2\}.$$

З А М Е Ч А Н И Е: Если r_1 и r_2 имеют общую схему, то $r_1 \times r_2 = r_1 \cap r_2$. Поэтому считаем, что операция пересечения – частным случаем операции естественного соединения.

Операция вычитания отношения r_1 и r_2 /обозначим $r_1 - r_2$/ определяется:

$$r_1 - r_2 = \{\mu \mid \mu\in r_1 \wedge \mu\bar{\in} r_2\}.$$

Под понятием реляционного выражения понимают выражение, обладающее операндами - реляционными схемами, операторами - операциями реляционной алгебры. В /3/ Кодд определил, что множество операции реляционной алгебры обладает реляционной полнотой.

Нам известно, что реляционное выражение может описывать много довольно сложных и разнообразных запросов пользователей, в которых операции выбора, проектирования и соединения имеют охватывающее значение.

Поэтому в этой работе мы будем рассматривать ограниченное реляционное выражение, в котором имеются только операции выбора, проектирования и соединения.

В /4/, /5/ были введены понятия о усиленной эквивалентности /strong equivalence/ и слабой эквивалентности /weak equivalence/ двух реляционных выражений.

Пусть E реляционное выражение с операндами-схемами отношений $R_1, R_2, \ldots, R_n$. Присваиванием α называется замена каждой схемы R_i $(i = \overline{1,n})$ в E ее соответствующим содержанием r_i, т.е.

$$\alpha = R_i \to r_i, \quad i = (\overline{1,n}).$$

Будем обозначать $V_\alpha(E)$ - значение реляционного выражения E соответствующее присваиванию α . А V(E) есть отображение присваивания α для операндов в выражении E на значение выражения $V_\alpha(E)$, т.е.

$$V(E) : \alpha \to V_\alpha(E).$$

Тогда два реляционных выражения E_1 и E_2 считаются усиленно эквивалентными, если $V_\alpha(E_1) = V_\alpha(E_2)$, $\forall\alpha$ или $V(E_1) = V(E_2)$.

Понятие о слабой эквивалентности основано на предположении существования универсального отношения на множестве атрубутов $\bigcup_{i=1}^{n} R_i$, и для определенного состояния I универсального отношения каждой схеме R_i присваивает соответственным значениям $r_i = \Pi_{R_i}(I)$.

Обозначить $V_I(E)$ - значение реляционного выражения E при состоянии I. $V_I(E)$ определяется индуктивно по числу реляционной операции в выражении E следующим образом:

1. Если в E имеется только схема отношения R_i, то:

$$V_I(E) = \Pi_{R_i}(I)$$

2а. Если $E = \sigma_{A_i \in \xi}(E_1)$, то

$$V_I(E) = G_{A_i \in \xi}(V_I(E_1)).$$

2б. Если $E = \Pi_X(E_1)$, то

$$V_I(E) = \Pi_X(V_I(E_1)).$$

2в. Если $E = E_1 \times E_2$, то

$$V_I(E) = V_I(E_1) \times V_I(E_2).$$

2г. Если $E = E_1 \cup E_2$, то

$$V_I(E) = V_I(E_1) \cup V_I(E_2).$$

2д. Если $E = E_1 - E_2$, то

$$V_I(E) = V_I(E_1) - V_I(E_2).$$

Рассматривается как отображение из состояний I универсального отношения в отношение $V_I(E)$.

Если $\forall I$, $V_I(E_1) = V_I(E_2)$, то говоря, что E_1 слабо эквивалентно /или эквивалентно/ E_2. Обозначают $E_1 \equiv E_2$.

В этой работе исследуется случай слабой эквивалентности. Но полученные результаты могут обобщать и на случай усиленной эквивалентности.

2. Понятие обобщенного табло

Обобщенным табло /обозн. τ / называется двухмерная матрица, каждый столбец которой соответствует одному из атрибутов универсального отношения в фиксированном порядке. Первая строка называется сводкой.

Символы матрицы могут быть:

1. Характерные переменные /обозначаются a_i/
2. Нехарактерные переменные /обозначены b_i/
3. Символы множеств значений ξ , принадлежащих доменам атрибутов универсального отношения /обозн. $\tilde{\xi}$ /
4. Бланки.

З А М Е Ч А Н И Е: В случае, когда каждое множество значений ξ имеет только один элемент, то получается табло T. Поэтому можно рассматривать T как частный обобщенного табло τ .

Пусть τ обобщенное табло , множество символов появляющихся в τ . Под понятием оценки /Evaluation/ ρ для τ понимают присваивание соответственного значения каждому символу из , и это присваивание значения удовлетворяет слудующим условиям:

а/ Если $\tilde{\xi} \in$ - символ множества значения, то $\tilde{\xi}$ присваивается одно значение $c \in \xi$. Будем обозначать $\rho(\tilde{\xi}) = c$.

Если $\xi = \emptyset$, то $\rho(\tilde{\xi}) = \emptyset$ и табло τ обладающее пустым множеством - пустое. Пустое табло /обозн. $\emptyset$/ отображает из любого состояния I универсального отношения в пустое отношение.

б/ Если $w = v_1 \, v_2 \, \dots \, v_n$ - одна строка табло τ , то $\rho(w) = \rho(v_1) \dots \rho(v_n)$. Множество атрибутов соответствующих столбцам не имеющим бланк на сводке τ называются целевой схемой отношения /target relation scheme/.

Значение табло τ соответствующее одному состоянию I универсального отношения /обозн. τ (I)/ опредяется следующим образом:

$\tau(I) = \{\rho(w_o) \mid$ для некоторой оценки ρ имеют $\rho(w_i) \in I$, $i = \overline{1,m}\}$, где w_o - сводка, w_i - строки табло τ.

Правила построения обобщенного табло

Пусть E ограниченное реляционное выражение. Тогда обобщенное табло τ для E строится индуктивно по числу операции в E следующим образом:

1. Если в E нет никакой операции, то E - схема отношения R. Тогда табло τ состоит из сводки и одной строки.

   а/ Если A_i - атрибут схемы R, то в столбце A_i на сводке и на строке имеется одинаковая характерная переменная a_i.

   б/ Если A_i не является атрибутом схемы R, то в столбце на сводке состоит бланк, и на строке ставить новую нехарактерную переменную.

2. Пусть $E = \sigma_{A_i \in \xi}(E_1)$. τ_1 обобщенное табло построенное для E_1. Тогда $\tau = \rho_{A_i \in \xi}(\tau_1)$ получается из τ_1 следующим образом:

   а/ Если в столбце A_i сводки τ_1 имеется бланк, то выражение E не имеет никакой смысл, и табло τ для E не определено.

   б/ Если в столбце A_i сводки τ_1 имеется символ множества значения $\bar{\xi}_1$ тогда:

   (i) Если $\xi_1 \cap \xi = \emptyset$, то $\forall I$:

   $V_I(E) = G_{A_i \in \xi}(V_I(E_1)) = \emptyset$, отсюда $\tau = \emptyset$.

   (ii) Если $\xi_1 \cap \xi \neq \emptyset$, то табло τ получается из τ_1 заменой $\bar{\xi}_1$ на $\bar{\xi}_2 (\xi_2 = \xi_1 \cap \xi)$ в местах где $\bar{\xi}_1$ появляется в столбце A_i.

   в/ Если τ_1 имеет характерную переменную a_i в столбце A_i сводки, то τ получается из τ_1 заменой a_i на $\bar{\xi}$ в местах, где a_i появляется в столбце A_i.

3. Пусть имеется $E = \Pi_X(E_1)$ и τ_1 - обобщённое табло для E_1. Табло $\tau = \Pi_X(\tau_1)$ получается из τ_1 следующим образом: в столбцах соответствующих атрибутам невходящих в X все символы сводки заменяются бланками, а характерные переменные на строках заменяются новыми нехарактерными переменными.

4. Пусть $E = E_1 \times E_2$ и τ_1, τ_2 два обобщенных табло для E_1, E_2 соответственно. Тогда $\tau = \tau_1 \times \tau_2$ для E строится следующим образом:

   а/ Если в столбце A_i сводки табло τ_1 имеет символ множества $\bar{\xi}_1$, а табло τ_2 имеет $\tilde{\xi}_2$ и $\xi_1 \cap \xi_2 = \emptyset$. Тогда $\forall I$, $V_I(E) = \Phi$, отсюда $\tau = \emptyset$.

   б/ Пусть $_1$ и $_2$ множества символов двух табло τ_1 и τ_2 соответственно. Ненарушая общность мы предполагаем, что τ_1 и τ_2 имеют одинаковые характерные переменные в столбцах соответствующих одним атрибутам, τ_1 и τ_2 имеют непересекающие множества нехарактерных переменных. Тогда табло τ включает в себя все строки табло τ_1 и τ_2 с символами определяющими по следующим правилам:

   (i) Если в столбце A_i сводки обе τ_1, τ_2 или одно из них имеется символ множества значений $\tilde{\xi}$ то в столбце A_i сводки табло τ ставит символ $\tilde{\xi}$. Все характерные переменные в столбце A_i табло τ заменяются символом $\tilde{\xi}$.

   (ii) Если в столбце A_i сводки τ_1 имеет символ множества значений $\tilde{\xi}_1$, τ_2 имеет символ множества $\tilde{\xi}_2 \cdot \xi_1 \cap \xi_2 \neq \emptyset$. Тогда в соответственном столбце A_i табло τ на сводке поставить символ множества $\tilde{\xi}$, $\xi = \xi_1 \cap \xi_2$, и во всех местах где появляются $\tilde{\xi}_1$, $\tilde{\xi}_2$ заменяют символом $\tilde{\xi}$.

   (iii) Если в столбце A_i обе τ_1, τ_2 или одно из них имеет характерную переменную a_i и два предыдущих правила не используется, то на сводке τ поставить характерную переменную a_i.

(iv) Для остальных случаев на сводке τ поставить бланки.

Пример:

1-й запрос /см. введение/ представить в виде реляционного выражения

$$E = \Pi_{Н,Ф,Д}(\sigma_{З\in\xi_1 \wedge У\in\xi_2}(R_1)),$$

где $\xi_1 = \{C/C > 1500 \wedge C \in dom(З)\}$, $\xi_2 = \{19\}$, тогда по правилам построения τ для E имеем:

$$\tau = \begin{array}{|ccccccc|} \multicolumn{1}{c}{Н} & Ф & Д & З & У & В & \multicolumn{1}{c}{А} \\ \hline a_1 & a_2 & a_3 & & & & \\ \hline a_1 & a_2 & a_3 & \tilde{\xi}_1 & \tilde{\xi}_2 & b_1 & b_2 \\ \hline \end{array}$$

Лемма 1: Пусть τ_1, τ_2 два обобщенных табло. Для любого состояния I универсального отношения имеют место следующие равенства:

1^o/ $\sigma_{A_i\in\xi}(\tau(I)) = (\sigma_{A_i\in\xi}(\tau))(I)$,

2^o/ $\Pi_X(\tau(I)) = (\Pi_X(\tau))(I)$,

3^o/ $\tau(I) \times \tau_1(I) = (\tau \times \tau_1)(I)$.

Доказательство:

1. Доказательство 1-ого равенства производится по двух шагам:

a/ $\sigma_{A_i\in\xi}(\tau(I)) \subseteq (\sigma_{A_i\in\xi}(\tau))(I)$.

Пусть $\sigma_{A_i\in\xi}(\tau(I)) \neq \emptyset$. В этом случае надо доказать, что $\forall\mu \neq \emptyset$, если $\mu\in \sigma_{A_i\in\xi}(\tau(I))$, то $\mu\in(\sigma_{A_i\in\xi}(\tau))(I)$.

Обозначаем множество символов табло τ и $\mu = (\mu_1 \ldots \mu_i \ldots \mu_n)$. Тогда существует оценка ρ для τ :

$$\rho(w_o) = \rho(v_1)\ldots \rho(v_i)\ldots \rho(v_n) = (\mu_1 \ldots \mu_i \ldots \mu_n),$$

$\rho(w_j) \in I, \quad j = \overline{1,m},$

$w_o = (v_1 \ldots v_i \ldots v_n)$ - сводка, w_j - строка табло τ.

По определению табло символы появляющиеся на сводке могут быть характерными переменными, символами множества значений или бланками. Так как $\sigma_{A_i \in \xi}(\tau(I)) \neq \emptyset$, поэтому v_i не может быть бланком. Рассмотрим два случая: (i) v_i - характерная переменная a_i. Тогда:

$$\rho(v_i) = \mu[A_i] = \mu_i \in \xi.$$

(ii) v_i - символ множества значений $\tilde{\xi}_1$. Тогда

$$\rho(v_i) = \rho(\tilde{\xi}_1) = \mu[A_i] = \mu_i \in \xi \cap \xi_1.$$

Пусть - множество символов табло $\tau' = \sigma_{A_i \in \xi}(\tau)$,

$w'_o = (u_1 \ldots u_i \ldots u_n)$ - сводка табло τ'.

По правилам построения τ' из τ очевидно что У′ отличается от У только одним символом v_i и u_i. Будем выбирать оценку ρ' для τ' следующим образом:

-$\forall d \in$ У′ и $d \in$ У /т.е. $d \neq u_i$/, тогда $\rho'(d) = \rho(d)$

$$\rho'(u_i) = \begin{cases} \rho'(\tilde{\xi}) = \rho(v_i) = \mu[A_i] = \mu_i \in \xi, & \text{если } v_i \text{ характерная переменная } a_i \\ \rho'(\tilde{\xi}_2) = \rho(\tilde{\xi}_1) = \mu[A_i] = \mu_i \in \xi_2, & \text{где } \xi_2 = \xi \cap \xi_1, \text{ если } v_i \text{ символ множества } \tilde{\xi}_1. \end{cases}$$

По способу построения ρ' и ρ имеем:

$$\rho'(w'_o) = \rho(w_o) = \mu; \quad \rho'(w'_j) = \rho(w_j) \in I, \quad j = \overline{1,m}$$

Следует $\mu \in \sigma_{A_i \in \xi}(\tau))(I)$.

Если $\sigma_{A_i \in \xi}(\tau(I)) \neq \emptyset$, видно, что а/ - удовлетворено.

б/ - аналогично можем доказать, что

$$\sigma_{A_i \in \xi}(\tau(I)) \supseteq (\sigma_{A_i \in \xi}(\tau))(I) \qquad (1.б)$$

Таким образом равенство 1^o/ доказано.

2/ По определению обобщенного табло и правила построения $\Pi_X(\tau)$ из τ легко показать, что: $\Pi_X(\tau(I)) = (\Pi_X(\tau))(I)$, $\forall I$.

3/ Для доказательства 3-его равенства, сначала мы покажем, что

а/ $\tau(I) \times \tau_1(I) \subseteq (\tau \times \tau_1)(I)$, $\forall I$.

Пусть $\tau(I) \times \tau_1(I) \neq \emptyset$. Надо доказать, что

$\forall \nu \neq \emptyset$, $\nu \in \tau(I) \times \tau_1(I) \rightarrow \nu \in (\tau \times \tau_1)(I)$.

По предложению $\nu = \mu \bowtie \mu$, где $\mu \in \tau(I)$, $\mu_1 \in \tau_1(I)$.

Это означает, что $\exists$ оценка ρ для τ и ρ_1 для τ_1 такие, что:

$$\rho(w_o) = \mu, \quad \rho(w_j) \in I, \quad j = \overline{1,m}$$

w_o - сводка, w_j - строка табло τ.

$$_1(w_o^1) = \mu_1, \quad \rho_1(w_j^1) \in I, \quad j = \overline{1,m_1}$$

w_o^1 - сводка, w_j^1 - строки табло τ_1.

Пусть $У$, $У_1$ множества символов двух табло τ, τ_1 соответственно.

$У'$ множество символов $\tau' = \tau \times \tau_1$. Выбираем оценку ρ' для τ' следующим образом:

если $d \in У'$, $d \in У$, $d \notin У_1$, то $\rho'(d) = \rho(d)$

если $d \in У'$, $d \notin У$, $d \in У_1$, то $\rho'(d) = \rho_1(d)$

если $d \in У'$, $d \in У$ $d \in У_1$, то $\rho'(d) = \rho(d) = \rho_1(d) = \mu[A_i] = \mu_1[A_i]$.

/Это имеет место, когда d является характерной переменной или символом множества $\tilde{\xi}$ в $У$ и $У_1$./

Если $d \in У'$, $d \notin У$, $d \notin У_1$, то по правилу построения $\tau \times \tau_1$, d может быть только символом множества значений в каком-нибудь столбце A_j сводки табло $\tau \bowtie \tau_1$. В соответственном столбце сводки табло τ имеет символ множества $\tilde{\xi}$, а табло τ_1 - $\tilde{\xi}_1$, $\xi \cap \xi_1 \neq \xi$, $\xi \cap \xi_1 \neq \xi_1$.

Мы имеем:

$$\rho'(d) = \rho'(\tilde{\xi}_2) = \rho(\tilde{\xi}) = \rho_1(\tilde{\xi}_1) = \mu[A_i] = \mu_1[A_j],\ \xi_2 = \xi \cap \xi_1.$$

На основе выбирания ρ' для $\tau \times \tau_1$ имеем:

$$\rho'(w'_o) = \nu\ ,\quad \rho'(w'_j) \in I,\quad j = \overline{1, m+m_1}\ ,$$

w'_o - сводка, w'_j - строка табло $\tau \times \tau_1$.

Отсюда следует, что $\nu \in (\tau \bowtie \tau_1)(I)$.

Если $\tau(I) \bowtie \tau_1(I) = \emptyset$, видно, что а/ - удовлетворено.

Так и требовалось доказать.

в. Аналогично можно доказать:

$$\tau(I) \bowtie \tau_1(I) \supseteq (\tau \times \tau_1)(I)\ .$$

Таким образом равенство 3/ доказано.

<u>Т е о р е м а 1</u>

Пусть E ограниченное реляционное выражение, использование правила 1 - 4 позволяет создать соответствующее обобщенное табло τ дли E:

$$V_I(E) = \tau(I)\quad \forall I\ .$$

<u>Д о к а з а т е л ь с т в о :</u>

Доказательство теоремы проводится индуктивно по числу операции в E.

1/ Пусть E - схема отношения R. Тогда из определения $\tau(I)$ и $V_I(E)$ непосредственно следует, что $V_I(E) = \tau(I)$.

2/ Пусть второе правило было использовано. $E = \sigma_{A_i \in \xi}(E_1)$ и τ_1 обобщённое табло E_1. Индуктивным образом имеем:

$V_I(E_1) = \tau_1(I)$. Обозначаем $\tau = \sigma_{A_i \in \xi}(\tau_1)$ - обобщенное табло для E построено с использованием правила 2/. Тогда на основе определения отображения V_I /правило 2а/ и леммы 1 получается:

$$V_I(E) = V_I\ \sigma_{A_i \in \xi}(E_1)) = \sigma_{A_i \in \xi}(V_I(E_1))$$
$$= \sigma_{A_i \in \xi}(\tau_1(I)) = (\sigma_{A_i \in \xi}(\tau_1))(I)$$
$$= \tau(I).$$

3/ Пусть третье правило было использовано $E = \Pi_X(E_1)$. По индуктивно, то $V_I(E_1) = \tau_1(I)\ \forall I$. Обозначение $\tau = \Pi_X(\tau_1)$ табло построенное для E использованием 3-его правила. Тогда на основе определения V_I /правило 2б/ и леммы 1 имеем:

$$V_I(E) = V_I(\Pi_X(E_1)) = \Pi_X(V_I(E_1))$$
$$= \Pi_X(\tau_1(I)) = (\Pi_X(\tau_1))(I)$$
$$= \tau(I).$$

4/ Пусть $E = E_1 \bowtie E_2$, τ_1 и τ_2 два обобщенных табло для E_1, E_2 соответственно. По предположению индукции

$$V_I(E_1) = \tau_1(I),\quad V_I(E_2) = \tau_2(I).$$

Обозначаем $\tau = \tau_1 \bowtie \tau_2$ табло построенное использованием 4-ого правила.

$$\text{Тогда: } V_I(E) = V_I(E_1 \bowtie E_2)$$
$$= V_I(E_1) \bowtie V_I(E_2) = \tau_1(I) \times\ _2(I)$$
$$= (\tau_1 \bowtie \tau_2)(I)$$
$$= \tau(I) \text{ и т.д. и т.п.}$$

Таким образом для любого ограниченного реляционного выражения E можем построить его соответствующее табло τ. Поэтому τ используется как способ представления реляционных выражений. В третьем разделе будем использовать эквивалентности обобщенных табло.

3. Эквивалентность обобщенных табло

Табло τ_1 и τ_2 называются эквивалентными /обозн. $\tau_1 \equiv \tau_2$/, если при любом состоянии универсального отношения I:

$$\tau_1(I) = \tau_2(I).$$

Табло τ_1 называется включенным в τ_2 /обозн. $\tau_1 \subseteq \tau_2$/, если $\forall I$

$$\tau_1(I) \subseteq \tau_2(I).$$

З а м е ч а н и е : Необходимым условием для $\tau_1 \equiv \tau_2$ или $\tau_1 \subseteq \tau_2$ является то, что τ_1 и τ_2 имеют общую целевую схему отношения.

Пусть τ_1 и τ_2 два обобщенных табло имеющих множество символов $У_1$ и $У_2$ соответственно. Отображение $\Psi : У_1 \to У_2$ называют гомоморфизмом, если Ψ удовлетворяет слудующим условиям:

1/ Есди $\tilde{\xi}_1 \in У_1$, $\tilde{\xi}_1$ - символ множества значения, то
$\Psi(\tilde{\xi}_1) = \tilde{\xi}_2$ - символ множества значений и $\xi_2 \subseteq \xi_1$

2/ Если $a_i \in У_1$ характерная переменная, то $\Psi(a_i)$ является или характерной переменной или символом множества значений $\tilde{\xi}_k$, который появляется на сводке в соответственном столбце табло τ_2.

3/ Если w_j какая-то строка табло τ_1, то $\Psi(w_j)$ тоже строка τ_2.

Аналогично в /4/ для обобщенного табло мы имеем место следующей теоремы.

Т е о р е м а 2 :

Пусть τ_1 и τ_2 два обобщенных табло с общей целевой схемой отношения, имеющего множества символов $У_1$ и $У_2$ соответственно. Необходимыми достаточными условиями для $\tau_2 \subseteq \tau_1$ является существование гомоморфизма

$$\Psi : У_1 \to У_2.$$

Доказательство:

Необходимость - Если $\tau_2 \subseteq \tau_1$, то существует гомоморфизм. Пусть ρ_2 оценка табло τ_2, которая взаимоднозначно отображает $У_2$ в множество значений С. Выберем I состояние универсального отношения составляющегося из таких кортежей: $\rho_2(w_j^2)$, $j=\overline{1,m_2}$, w_j^2 - строки табло τ_2.

С таким выбором I и ρ_2 - оценка для τ_2 имеем:

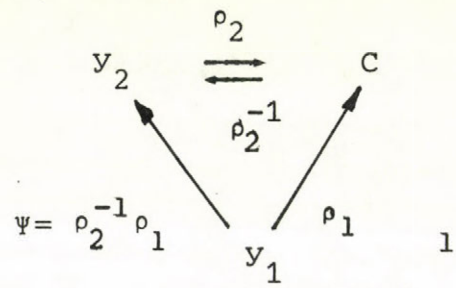
$$\rho_2(w_o^2) = \mu \in \tau_2(I), \quad w_o^2 \text{ - сводка } \tau_2.$$

По предположению $\tau_2 \subseteq \tau_1 \to \mu \in {}_1(I)$, отсюда следует, что существование оценки ρ_1 для τ_1 таково, что:

$$\rho_1(w_o^1) = \mu$$

$$\rho_1(w_j^1) \in I, \quad j = \overline{1,m_1}$$

где w_o^1 - сводка, w_j^1 - строки табло τ_1.



(р.1)

Построим отображение /рис. 2./ $\psi = \rho_2^{-1} \rho_1$, и докажем, что ψ гомоморфизм. То есть ψ нужно удовлетворять условиям 1-3.

Пусть $w_o^1 = v_1 \ldots v_i \ldots v_n$

$$w_o^2 = u_1 \ldots u_i \ldots u_n.$$

Исходя из того, что τ_1 и τ_2 имеют общую целевую схему отношения и $\rho_1(w_o^1) = \rho_2(w_o^2) = \mu$, поэтому $\psi : v_i \to u_i$, где v_i, u_i являются либо характерными переменными либо символами множества значений. Таким образом ψ удовлетворяет 2-ому условию гомоморфизма.

Если v_i символ множества значений $\tilde{\xi}_1$, то $\psi(v_i)$ не может быть характерной переменной. Действительно в обратном случае $\psi(\tilde{\xi}_1) = a_i$ и с - значение присвоенное характерной переменной, $c \in \mathrm{dom}(A_i)$, $c \notin \xi_1$.

Строим I' из I с заменой $\rho_2(a_i)$ в столбце i значением c и выберем оценку $\rho'_2 : \rho'_2(d) = \rho_2(d)$, для $d \in У_2$ и $d \neq a_i$; $\rho'_2(a_i) = c$.

Тогда $\mu' = \rho'_2(w_o^2) \in \tau_2\ (I')$, но $\mu' \notin \tau_1\ (I')$. Это противоречит предположению $\tau_2 \subseteq \tau_1$.

И так $\Psi(\tilde{\xi}_1)$ является символом множества значений $\tilde{\xi}_2$. Более этого $\xi_2 \subseteq \xi_1$. Действительно в обратном случае $\exists c \in \xi_2 - \xi_1 \neq \emptyset$ и аналогичным рассуждением предыдущему пришли к противоречию $\tau_2 \nsubseteq \tau_1$.

Таким образом Ψ удовлетворяет 1-ому условию гомоморфизма.

Сейчас нам нужно доказать, что Ψ удовлетворят 3-ему условию гомоморфизма, где $\Psi = \rho_2^{-1}\ \rho_1$.

Если w_j^1 - строка τ_1, то $\rho_1(w_j^1)$ - один кортеж состояния I.
ρ_2 - взаимнооднозначное отображение из $У_2$ в I. Тогда
$\rho_2^{-1}(\rho_1(w_j^1))$ - строка τ_2.

<u>Д о с т а т о ч н о с т ь :</u> Пусть существует гомоморфизм $\Psi : У_1 \to У_2$, I - любое состояние универсального отношения, $\mu \in \tau_2(I)$. Тогда оценка ρ_2 для τ_2: ρ_2 отображает множество $У_2$ табло τ_2 в множество значений C, т.е.

$$\rho_2 : У_2 \to C$$

$$\rho_2(w_o^2) = \mu;\quad \rho_2(w_j^2) \in I\ ,\ j = \overline{1,m_2}$$

w_o^2 - сводка, w_j^2 - строки τ_2.

Для иллюстрации $\mu \in \tau_1(I)$ выберем оценку ρ_1 для τ_1 таким образом /р.2/

$$\forall d \in У_1 \to \rho_1(d) = \rho_2(\Psi(d))$$

$$\begin{tikzcd} У_1 \arrow[r, "\Psi"] \arrow[dr, "\rho_1 = \rho_2(\Psi)"'] & У_2 \arrow[d, "\rho_2"] \\ & C \end{tikzcd}$$

/р.2/

По определению гомоморфизма, если $\tilde{\xi}_1 \in Y_2$ - символ множества значений, то

$$\rho_1(\tilde{\xi}_1) = \rho_2(\Psi(\tilde{\xi}_1)) = \rho_2(\tilde{\xi}_2) = c \in \xi_2$$

С другой стороны $\tilde{\xi}_2 = \Psi(\tilde{\xi}_1)$, $\xi_2 \subseteq \xi_1 \to c \in \xi_1$.

Из условия гомоморфизма следует

$$\rho_1(w_o^1) = \rho_2(\Psi(w_o^1)) = \rho_2(w_o^2) = \mu$$

w_o^1, w_o^2 - сводки τ_1 и τ_2 соответственно.

$$\rho_1(w_j^1) = \rho_2(\Psi(w_j^1)) = \rho_2(w_k^2) \in I$$

w_j^1, w_k^2 - строки τ_1 и τ_2 соответственно.

Таким образом $\mu \in \tau_1(I)$ и $\tau_2 \subseteq \tau_1$. Теорема доказана.

Пусть τ_1 и τ_2 обобщенные табло, Θ отобразает строки табло τ_1 в строки τ_2.

Θ называется включенным отображением, если оно удовлетворяет следующим условиям:

а/ Если в столбце A_k на строке i табло τ_1 имеется характерная переменная, то в столбце A_k на строке $\Theta(i)$ табло τ_2 также имеется характерная переменная.

б/ Если в столбце A_k на строке i табло τ_1 имеется символ множества значений $\tilde{\xi}$, то в столбце A_k на строке $\Theta(i)$ табло τ_2 имеется символ множества значений $\tilde{\xi}' : \xi' \subseteq \xi$.

в/ Если столбец A_k на строках i и j табло τ_1 имеют одинаковые нехарактерные переменные, то в столбце A_k на строках $\Theta(i)$ и $\Theta(j)$ табло τ_2 имеют одинаковые символы. Этот символ может быть символом множества значений, характерной или нехарактерной переменной.

Включенное отображение является основой для проверки эквива-

4. Связь между табло и обобщенным табло

в /7/ введены понятие монотонного реляционного выражения и объединения табло. Под понятием монотонного реляционного выражения понимают вырашения обладающие операциями выбора, проектирования, соединения и объединения.

Объединением табло называется выражение $\bigcup\limits_{i=1}^{n} T_i$, где $T_1, T_2, \ldots, T_n$ табло имеющие общую целевую схему отношения. Общая целевая схема отношения табло T_i, $i = \overline{1,n}$ является целевой схемой объединения табло $\bigcup\limits_{i=1}^{n} T_i$. Объединение табло - форма представления монотонного реляционного выражения.

При одном состоянии I универсального отношения значения объединения табло определеяются следующим образом: $(\bigcup\limits_{i=1}^{n} T_i)(I) = \bigcup\limits_{i=1}^{n} T_i)(I)$.

Аналогично предыдущему мы имеем понятие объединения обобщенных табло. Пусть $\tau_1, \ldots, \tau_n$ обобщенные табло имеющие общую целевую схему отношения.

Тогда $\bigcup\limits_{i=1}^{n} \tau_i$ объединение обобщенных табло. $\forall I$ - значение объединения обобщенных табло определяется таким же образом:
$(\bigcup\limits_{i=1}^{n} \tau_i)(I) = \bigcup\limits_{i=1}^{n} \tau_i(I)$.

Для выявления связи между табло (T) и обобщенным табло (τ) мы рассмотрим правило разложения обобщенного табло τ на объединение табло.

Пусть У - множество символов τ и табло τ имеет только один символ множества значений $\tilde{\xi}$ в У /обозначаем $\tau = \tau\ |\xi|$/. Мы будем обозначать $S_c^{\tilde{\xi}}\ \tau(|\xi|)$ замена $\tilde{\xi}$ в табло τ одним элементом $c \in \xi$ в местах, где $\tilde{\xi}$ появляется. $\tau[c]$ - табло, полученное после замены. Видно, что $\tau[c] = T[c]$ так как табло (T) является частным случаем обобщенного табло когда все множества имеют один элемент, т.е.

$$S_c^{\tilde{\xi}}\,(\tau\,[\xi]) = \tau\,[c] = T[c].$$

Если τ имеет k символов множества значений $\tilde{\xi}_1, \tilde{\xi}_2, \ldots, \tilde{\xi}_k$, т.е. $\tau = \tau\,[\xi_1, \ldots, \xi_k]$. Обозначаем через $S_{c_1 \ldots c_k}^{\tilde{\xi}_1 \ldots \tilde{\xi}_k}(\tau\,[\xi_1 \ldots \xi_k])$ - замены в обобщенном табло каждого символа значения $\tilde{\xi}_i$ одним элементом $c_1 \in \xi_i$.

Табло полченное после замен имеет вид:

$$S_{c_1 \ldots c_k}^{\tilde{\xi}_1 \ldots \tilde{\xi}_k}(\tau\,[\xi_1 \ldots \xi_k]) = \tau\,[c_1 \ldots c_k] = T[c_1 \ldots c_k].$$

<u>Л е м м а 2 :</u> Пусть τ - обобщенное табло, имеющее только один символ множества значений $\tilde{\xi}$ в У, т.е. $\tau = \tau\,[\xi]$. Тогда $\tau\,[\xi] = \bigcup\limits_{c \in \xi} T[c]$.

<u>Д о к а з а т е л ь с т в о :</u>

Надо доказать, что $\forall I$ имеет место равенства:

$$(\tau\,[\xi])\,(I) = (\bigcup_{c \in \xi} T[c])$$

пусть $\mu \in ([\xi])(I)$ и $\mu \neq \emptyset$. Это означает существование оценки ρ для $\tau\,[\xi]$, такой, что: $\rho\,(w_o) = \mu$, $\rho\,(w_j) \in I$, $j = \overline{1,m}$; $\rho\,(\tilde{\xi}) = c_k \in \xi$,

w_o - сводка, w_j - строки табло τ.

По правилу разложения табло τ, в $\bigcup\limits_{c \in \xi} T(c)$ имеется табло $T[c_k] = S_{c_k}^{\xi}\,\tau\,[\xi]$. Если S - множество символов табло T, то S и У отличаются только одним символом c_k и $\tilde{\xi}$.

Выберем оценку ρ' для $T[c_k]$ следующим образом:

$$\rho'(d) = \rho(d), \ \forall d \in S, \ \ d \neq c_k; \ \ \rho'(c_k) = \rho(\tilde{\xi}) = c_k$$

Тогда $\rho'(w_o^T) = \mu$, $\rho'(w_j^T) \in I$, $j = \overline{1,m}$

w_o^T - сводка, w_j^T - строки табло T.

лентности табло и установления эквивалентных преобразований.

Т е о р е м а 3: Пусть τ_1, τ_2 - обобщенные табло имеющие общую целевую схему отношения. Необходимым и достаточным условиями для $\tau_2 \subseteq \tau_1$ является существование включенного отображения $\Theta : \tau_1 \to \tau_2$.

Д о к а з а т е л ь с т в о :

Необходимость: Если $\tau_2 \subseteq \tau_1$, то по теореме 3 существует гомоморфизм $\Psi : У_1 \to У_2$. Надо доказать, что Ψ является включенным отображением.

Из 1-ого и 2-ого условия гомоморфизма следует, что Ψ удовлетворяет условиям а/ и б/ включенного отображения. Из 3-его условия гомоморфизма следует, что Ψ отображает строки табло τ_1 в строки табло τ_2, одновременно Ψ удовлетворяет условию с/ включенного отображения.

Д о с т а т о ч н о с т ь :

Пусть существует включенное отобразение Θ надо доказать, что $\tau_2 \subseteq \tau_1$.

Пусть отобразение $\Psi : У_1 \to У_2$ удовлетворяет следующему условию: если d_1 - символ в столбце A_k на строке w_j^1 табло τ_1 и d_2 символ в столбце A_k на строке $\Theta(w_j^1)$ табло τ_2, то $\Psi(d_1) = d_2$.

Докажем, что Ψ - гомоморфизм.

Известно, что Θ - включенное отображение. Из условия б/ отображения Θ следует Ψ удовлетворяет 1-ому условию. Из условия а/ отображения Θ следует, что Ψ удовлетворяет 2-ому условию. Ψ тоже удовлетворяет 3-ему условию, так как Ψ было определено на основе отображения Θ. Таким образом Ψ гомоморфизм. По теореме 3, $\tau_2 \subseteq \tau_1$.

С л е д с т в и е 1: τ_1, τ_2 два обобщенных табло имеющих общую целевую схему отношения. Необходимым и достаточным условием для $\tau_1 \equiv \tau_2$ является существование включенного отображения из τ_1 в τ_2 и включенного отображения из τ_2 в τ_1 .

И так $\mu \in (T[c_k])(I) \Rightarrow \mu \in (\bigcup_{c \in \xi} T[c])(I)$.

Аналогичным образом мы можем доказать, что если

$$\mu \in (\bigcup_{c \in \xi} T[c])(I) \Rightarrow \mu \in (\tau[\xi])(I) .$$

Лемма доказана.

Из леммы 2 доказать индуктивно по числу множество значений в табло τ имеем место следующей теоремы:

Т е о р е м а 4 : Пусть τ обобщенное табло, имеющее k символов множеств значений $\tilde{\xi}_1, \tilde{\xi}_2, \ldots, \tilde{\xi}_k$ в k столбцах соответствующих k атрибутам табло τ . Тогда имеет место следующее равенство:

$$\tau[\xi_1, \ldots, \xi_k] = \bigcup_{c_1 \in \xi_1} \ldots \bigcup_{c_k \in \xi_k} T[c_1 \ldots c_k].$$

З а м е ч а н и е : Не любое произвольное реляционное выражение представленное в виде объединения табло может быть представлено в виде обобщенного табло.

П р и м е р : Реляционное выражение:

$$\Pi_{Н,Ф,Д,З}(\sigma_{У=19 \wedge (\text{`ДИРЕКТОР'} \vee З=2000)}(R_1))$$

Может быть представлено в виде объединения, но не может быть представлено в виде эквивалентного обобщенного табло.

З а к л ю ч е н и е: С понятием табло T в [4], [5], [7] запросы обладающие операцией выбора вида $\sigma_{A \in \{a_1 \ldots a_n\}}(r)$ обычно

подлежат разложению на объединения n подзапросов и будут представлены в виде объединения табло, каждое из которых соответствует одному запросу. Например, выражение $\Pi_{Н,Ф,У}(\sigma_{У \in \{17,19,32\}}(R_1))$ может быть представлено в виде объединения трех подвыражений:

$$\Pi_{Н,Ф,У}(\sigma_{У=17}(R_1)) \cup \Pi_{Н,Ф,У}(\sigma_{У=19}(R_1)) \cup \Pi_{Н,Ф,У}(\sigma_{У=32}(R_1)) .$$

Тем самым одно может быть представлено в виде $\bigcup_{i=1} T_i$, каждое T_i соответствует подвыражению. С точки зрения осуществления на практике, такой подход будет дорогостоящим из-за большой запоминающей емкости и вычислительного времени ЭВМ. Это объясняется тем, что количество табло, находящихся в памяти, и, с которыми все время должны обращаться, равно n. В случае достаточно большого n работа будет слишком сложной. В то время как показали выше, такие запросы могут быть представлены в виде эквивалентного обобщенного табло τ . Поэтому обработка таких выражений будет более удобной и эффективной. Кроме этого введение понятия обобщенного табло еще позволяет расширить результаты в /4/, /5/, /7/ для реляционных выражений с операцией выбора $\sigma_{A\theta C}$, где $\theta = \{\neq,<,\leq,>,\geq\}$.

В общем для достижения этих расширенных результатов требуются более сложные и тонкие техники доказательства.

Автор благодарит доц. Хо Тхуан, Нгуен Кат Хо и Ле Тиен Вьонг за полезные обсуждения.

Л И Т Е Р А Т У Р А

[1] Codd, E.F., A Relational Model of Data for Large Shared Data Banks, Comm. Ac., 1970, v.13, No.6, p. 337-387.

[2] Codd, E.F., Further Normalization of the Database Relation Relational Model. In: Database Systems, Englewood Cliffs, N.Y.: Prentice-Hall, 1972, p. 33-64.

[3] Codd, E.F., Relational Completeness of Database Sublanguages. In: Database Systems. Englewood Cliffs, N.Y. :Prentice-Hall, 1972, p. 79-90.

[4] Aho A.V., Sagiv, Y., and Ullman, J.D., Efficient Optimization of a class of Relational Expressions. ACM Trans. Database Syst., 4,4 :Dec, 1979, p. 435-454.

[5] Aho, A.V., Sagiv, Y., and Ullman, Y.D., Efficient Optimization of a Relational Expressions. ACM Trans. Database Syst., 4,4 :Dec. 1979. p. 435-454.

[6] Astrahan, M.M., Chamberlin, D.D., Implementation of a Structured English Query Language. Communication of the ACM Oct. 1975 Vol. 18, Number 10.

[7] Sagiv, A., and Yannakakis, M., Equivalence Among Relational Expressions with the Union and Difference Operators., Y, if the Association for Computing Machinery, Vol. 27, No. 4, Oct. 1980.

[8] Aho A.V., Beeri, C., and Ullman, Y.D., The Theory of Joins in Relational Database Proc. 18-th IEEE Symposium on Foundations of Computer Science, 1977, p. 107-113.

[9] Деит К.: Введение в системы баз данных. М. Наука. 1980

[10] Мартин Дж.: Организация баз данных в вычислительных системах. М. Мир. 1980.

[11] Горчинская О.Ю.: Теоретический аспект построения реляционных моделей. А. и Т. № 1, 1983.

The equivalence of a class of relational expressions

Do Suan Tho

Summary

Many database queries can be formulated in terms of relational expressions using the relational operations select, project, join. The equivalence problem for these queries is studied with query optimization in mind.

In this paper we introduce the so-called generalized tableaux, two-dimensional representations of queries. Every relational expression over the operations select of the $\sigma_{A\in\xi}(r)$ form (A-attribute of the relation r, $\xi \subseteq \mathrm{dom}(A)$), project and join can be represented by generalized tableaux. We consider the equivalence problem for generalized tableaux, then we discuss the relationship between tableaux and generalized ones.

A relációs kifejezések egy osztályának ekvivalenciájáról

Do Suan Tho

Összefoglaló

Több adat-bázis lekérdezést relációs kifejezések formájában lehet megfogalmazni, használva a kiválasztás, projekció és összekapcsolás operációkat. Ezen lekérdezések ekvivalencia problémájáról van szó, különös tekintettel az optimalizációra. A cikkben a szerző egy u.n. általánositott táblát vezet be, a lekérdezések kétdimenziós reprezentációját.
Minden olyan relációs kifejezés, amelynek $\sigma_{A\in\xi}(r)$ formája van, reprezentálható általánositott táblával. A cikkben az ekviva-

lencia problémát a szerző egyrészt az általánositott táblákra és az általánositott és közönséges táblákra vonatkozóan vizsgálja.

MTA SZTAKI Közlemények 34/1986 pp.151-162

SOME INVARIANTS OF COVERS FOR FUNCTIONAL DEPENDENCIES

Ho Thuan

MTA SZTAKI

ABSTRACT

The nonredundant and minimum covers have been investigated by different approaches in [1], [2], and several useful properties of them have been proved and used in various problems in the logical design of databases. But few attention is paid to the study of invariants concerning the attribute sets of the left and right sides of these covers. In this paper, we prove some additional invariants for covers and nonredundant covers.

Most of these results have been presented in [3] but the proofs given there are too long and the required assumptions are too strict. We will provide them in a weakened form and with much shorter proofs.

§1. INTRODUCTION

In most studies concerning covers for functional dependencies, we usually start from a set F of FDs on $\Omega=\{A_1,A_2,\dots,A_n\}$

$$F = \{L_i \to R_i \mid i = \overline{1,k} \quad L_i, R_i \subseteq \Omega\}$$

and try to find a shorter representation for F, i.e. a new set F′ of FDs with either a fewer number of FDs or a less total size such that F and F′ imply the same set of FDs. So doing, several algorithms concerning relational databases which start with a smaller cover will run faster.

But few attention is paid to the study of invariants concerning the attribute sets of the left and right sides of these covers.

In the following, we prove some additional invariants for covers and nonredundant covers.

Most of these results have been presented in [3] but the proofs given there are too long and the required assumptions are too strict. We will provide them in a weakened form and with much shorter proofs. These invariants can be used, for instance, as a simple criterion to check whether two sets of FDs are not equivalent.

In this section we recall some notions and results in the theory of relational database needed in the sequel.

Definition 2.1. A set $F=\{L_i \to R_i \mid L_i, R_i \subseteq \Omega,\ i=\overline{1,k}\}$ of FDs on Ω is in natural reduced form if

(i) $L_i \neq L_j$ for all $i \neq j$

(ii) $L_i \cap R_j = \emptyset \quad \forall i=\overline{1,k}$.

Through this paper we will only consider sets of FDs in natural reduced form (since this form is the most natural and concise representation), and we assume that all attributes are chosen from some fixed universe Ω.

Given a set F of FDs, we denote by F^+ the closure of F, i.e. the set of all FD that can be inferred from the FDs in F by repeated application of Armstrong's axioms [5].

Definition 2.2. The sets of FDs over Ω

$$F_1=\{L_i^{(1)} \to R_i^{(1)} \mid i=\overline{1,k_1}\} \quad \text{and}$$

$$F_2=\{L_i^{(2)} \to R_i^{(2)} \mid i=\overline{1,k_2}\}$$

are said equivalent, written $F_1 \equiv F_2$, if $F_1^+=F_2^+$. If $F_1 \equiv F_2$ then F_1 is a cover for F_2.

Definition 2.3. A set F of FDs is nonredundant if there is no proper subset F' of F with $F' \equiv F$. If such F' exists, F is redundant. F_1 is a nonredundant cover for F_2 if F_1 is a cover for F_2 and F_1 is nonredundant.

Let F be a set of FD over Ω and let $X \to Y$ be an FD in F. Attribute A is said extraneous in $X \to Y$ if A can be removed from the left side or right side of $X \to Y$ without changing the closure of F.

Definition 2.4. Let F be a set of FDs over Ω and let $X \to Y$ be in F.

$X \to Y$ is left reduced if X contains no attribute A extraneous in $X \to Y$.

$X \to Y$ is right reduced if Y contains no attribute A extraneous in $X \to Y$.

$X \to Y$ is reduced if it is left-reduced and right reduced and $Y \neq \emptyset$.

A set F of FDs is left-reduced (right-reduced, reduced) if every FD in F is left-reduced (respectively right-reduced, reduced).

Lemma 1.1. Given sets of FDs F_1 and F_2 over Ω. $F_1 \equiv F_2$ if and only if $F_1 \subseteq F_2^+$ and $F_2 \subseteq F_1^+$.

Lemma 1.2. [4] Let F be a set of FDs on Ω,

$$F = \{L_i \to R_i \mid i = \overline{1,k}\} .$$

If $(X \to Y) \in F^+$ and $A \in L(F)$ where $L(F) = \bigcup_{i=1}^{k} L_i$, then

$$(X \setminus \{A\} \to Y \setminus \{A\}) \in F^+ .$$

Let F be a set of FDs on Ω

$$F = \{L_i \to R_i \mid L_i, R_i \subseteq \Omega,\ i = \overline{1,k}\}.$$

Let us denote by

$$AF = \{L_i \to R_i \mid (L_i \to R_i) \in F \wedge |L_i| = 1\},$$

the set of all FDs in F with left side consists of only one attribute, and

$$\mathcal{L}(AF)=\{L_i \mid (L_i\to R_i) \in AF\} \ .$$

We have the following lemma:

Lemma 3.1. Let F_1 and F_2 be two equivalent sets of FDs on Ω

$$F_1=\{L_i^{(1)}\to R_i^{(1)} \mid i=\overline{1,k_1}\} \ ,$$

$$F_2=\{L_i^{(2)}\to R_i^{(2)} \mid i=\overline{1,k_2}\} \ .$$

Then

$$\mathcal{L}(AF_1)=\mathcal{L}(AF_2) \ .$$

Proof. Without loss of generality, suppose that there exists $L_j^{(1)}=A\,i_j \in \mathcal{L}(A\,F_1)\setminus\mathcal{L}(A\,F_2)$. It is easy to show that $L_j^{(1)}\to R_j^{(1)} \in F_2^{\,+}$.

Infact, it is obvious that

$$(L_j^{(1)})^+_{F_2} = L_j^{(1)} \ .$$

On the other hand we have

$$L_j^{(1)} \cap R_j^{(1)} = \emptyset \ ,$$

(F_1 is in natural reduced form).

Hence

$$R_j^{(1)} \not\subseteq (L_j^{(1)})^+_{F_2} \ ,$$

showing that $L_j^{(1)}\to R_j^{(1)} \notin F_2^+$, a contradiction. The lemma is proved.

Example 3.1. Let be given

$$\Omega = A\,B\,C\,D\,E$$

$$F_1=\{A\to BC,\ AD\to CE\}$$

$$F_2=\{A\to B,\ B\to C,\ AD\to CE\}.$$

We have

$$\mathcal{L}(AF_1)= \{A\}$$

$$\mathcal{L}(AF_2)= \{A,B\} \neq \mathcal{L}(AF_1)$$

Hence

$$F_1 \not\equiv F_2$$

Remark 3.1. Lemma 3.1 is equivalent to the assertion that for a given FD-graph $G_F = <V,E>$, all covers of G_F have the same set of simple nodes without outgoing arcs [2].

Lemma 3.2. Let be given two equivalent sets of FDs in natural reduced form on Ω:

$$F_1 = \{L_i^{(1)} \rightarrow R_i^{(1)} \mid i=\overline{1,k_1}\}$$
$$F_2 = \{L_i^{(2)} \rightarrow R_i^{(2)} \mid i=\overline{1,k_2}\}.$$

Then

$$R(F_1) = R(F_2)$$

where
$$R(F_j) = \bigcup_{i=1}^{k_j} R_i^{(j)}, \ j=1,2 .$$

Proof. We first show that $R(F_1) \subseteq R(F_2)$. Let $A \in R(F_1)$. It follows that there exists

$$L_i^{(1)} \rightarrow R_i^{(1)} \text{ with } R_i^{(1)} = AX, \ X \neq R_i^{(1)} .$$

Since $F_1 \equiv F_2$, we have

$$(L_i^{(1)} \rightarrow AX) \in F_2^+$$

or, equivalently,

$$AX \subseteq (L_i^{(1)})^+_{F_2} = L_i^{(1)} \cup \left(\bigcup_{j\in\alpha} R_j^{(2)}\right)$$
$$\alpha \subseteq \{1,2,\ldots,k_2\}$$

On the other hand: $AX \cap L_i^{(1)} = \emptyset$.

This shows that $A \in R(F_2)$.

Similarly, we can show that $R(F_2) \subseteq R(F_1)$.

Hence

$$R(F_1) = R(F_2).$$

Remark 3.2. It is obvious that lemma 3.2 holds in the case F_1, F_2 are equivalent, nonredundant and right reduced, and this is the content of one theorem in [3].

Example 3.2. Let be given

$$\Omega = A\,B\,C\,D\,E$$

$$F_1 = \{A \to BC,\ AD \to CE\}$$
$$F_2 = \{A \to BD,\ AD \to CE\}$$

We have

$$R(F_1) = BCE \neq R(F_2) = BCDE$$

Hence

$$F_1 \not\equiv F_2 .$$

Theorem 3.1. Let F_1 and F_2 be two equivalent and non-redundant sets of FDs on Ω.

$$F_j = \{L_i^{(j)} \to R_i^{(j)} \mid i = \overline{1,k_j}\} \qquad j=1,2$$

Then
$$L(F_1) \setminus R(F_1) = L(F_2) \setminus R(F_2) ,$$

where
$$L(F_j) = \bigcup_{i=1}^{k_j} L_i^{(j)} , \qquad j=1,2 .$$

$$R(F_j) = \bigcup_{i=1}^{k_j} R_i^{(j)} , \qquad j=1,2 .$$

Proof. First we prove that

$$L(F_1) \setminus L(F_2) \subseteq R(F_1).$$

Suppose $A \in L(F_1) \setminus L(F_2)$, i.e.

$$A \in L(F_1) \quad \text{and} \quad A \bar{\in} L(F_2).$$

Then, there exists

$$(L_i^{(1)} \to R_i^{(1)}) \in F_1$$

with $L_i^{(1)} = AX$, $X \neq \emptyset$. (This follows from $A \in L(F_2)$ and lemma 3.1)
Since F_1 is nonredundant,

$$L_i^{(1)} \rightarrow R_i^{(1)}$$

must participate in some derivation sequence for some $(L_h^{(2)} \rightarrow R_h^{(2)}) \in F_2$. [6].

So we have:

$$L_h^{(2)} \supseteq L_{i_1}^{(1)}$$

$$L_h^{(2)} R_{i_1}^{(1)} \supseteq L_{i_2}^{(1)}$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$L_h^{(2)} R_{i_1}^{(1)} R_{i_2}^{(1)} \ldots R_{i_t}^{(1)} \supseteq L_i^{(1)} = AX$$

$$L_h^{(2)} R_{i_1}^{(1)} R_{i_2}^{(1)} \ldots R_{i_t}^{(1)} R_i^{(1)} \supseteq L_{i_{t+2}}^{(1)}$$

$$\cdots\cdots\cdots\cdots\cdots$$

So

$$A \in L_h^{(2)} R_{i_1}^{(1)} \ldots R_{i_t}^{(1)} .$$

Since $A \bar{\in} L(F_2)$, it is obvious that $A \in R(F_1)$. Thus we have proved that

$$L(F_1) \setminus L(F_2) \subseteq R(F_1).$$

Similarly, we can prove that

$$L(F_2) \setminus L(F_1) \subseteq R(F_2).$$

On the other hand, by lemma 3.2,

$$R(F_1) = R(F_2) .$$

Consequently,

$$\begin{aligned} L(F_1)\setminus R(F_1) &= \{[L(F_1)\setminus L(F_2)]\setminus R(F_1)\} \cup \{[L(F_1) \cap L(F_2)]\setminus R(F_1)\} = \\ &= [L(F_1) \cap L(F_2)]\setminus R(F_1) = [L(F_2) \cap L(F_1)]\setminus R(F_2) \\ &= L(F_2)\setminus R(F_2). \end{aligned}$$

The theorem is completely proved.

Remark 3.3. In [3] theorem 3.1. have been proved under the additional assumption that F_1 and F_2 are both R-reduced.

Example 3.3. We take up again the example 3.2.

$$\Omega = A\,B\,C\,D\,E$$
$$F_1 = \{A \to BC,\ AD \to CE\}$$
$$F_2 = \{A \to BD,\ AD \to CE\}$$

We have $L(F_1) \setminus R(F_1) = AD \neq L(F_2) \setminus R(F_2) = A$

Hence

$$F_1 \not\equiv F_2 \; .$$

Theorem 3.2. Let F_1 and F_2 be any two equivalent and nonredundant sets of FDs over Ω

$$F_j = \{L_i^{(j)} \to R_i^{(j)} \mid i = \overline{1,k_j}\} \qquad j=1,2 \; .$$

Then

$$L(F_1) \cup R(F_1) = L(F_2) \cup R(F_2).$$

Proof. We first prove that

$$L(F_1) \cup R(F_1) \subseteq L(F_2) \cup R(F_2).$$

By lemma 3.2., we have

$$R(F_1) = R(F_2) \subseteq L(F_2) \cup R(F_2) \; .$$

We have to prove

$$L(F_1) \subseteq L(F_2) \cup R(F_2) \; .^{*)}$$

Suppose $A \in L(F_1)$. Then there exists $(L_i^{(1)} \to R_i^{(1)}) \in F_1$ with $L_i^{(1)} = AX$. Since F_1 is nonredundant, $L_i^{(1)} \to R_i^{(1)}$ must participate in some derivation sequence for some $(L_h^{(2)} \to R_h^{(2)}) \in F_2$, such that

---

*) It is much better to argue as follows:
By the proof of theorem 3.1 we have $L(F_1) \setminus L(F_2) \subseteq R(F_1)$.
But $R(F_1) = R(F_2)$. Hence $L(F_1) \subseteq L(F_2) \cup R(F_2)$ ...

$$L_h^{(2)} \supseteq L_{i_1}^{(1)}$$

$$L_h^{(2)} R_{i_1}^{(1)} \supseteq L_{i_2}^{(1)}$$

$$L_h^{(2)} R_{i_1}^{(1)}\ R_{i_2}^{(1)} \supseteq L_{i_3}^{(1)}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$L_h^{(2)} R_{i_1}^{(1)} R_{i_2}^{(1)} \ldots R_{i_t}^{(1)} \supseteq L_i^{(1)} = AX$$

$$L_h^{(2)} R_{i_1}^{(1)} \ldots R_{i_t}^{(1)} R_i^{(1)} \supseteq L_{i_{t+2}}^{(1)} .$$

It follows that either $A \in L_h^{(2)} \subseteq L(F_2)$, or $A \in R_{i_1}^{(1)} R_{i_2}^{(1)} \ldots R_{i_t}^{(1)} \subseteq R(F_1) \subseteq R(F_2) \cup L(F_2)$.
In both cases, $A \in L(F_2) \cup R(F_2)$, showing

$$L(F_1) \cup R(F_1) \subseteq L(F_2) \cup R(F_2) .$$

Similarly, we can prove that

$$L(F_2) \cup R(F_2) \subseteq L(F_1) \cup R(F_1) .$$

Q.E.D.

Theorem 3.3. Let F_1 and F_2 be any two equivalent, non-redundant and left-reduced sets of FDs on Ω

$$F_j = \{L_i^{(j)} \rightarrow R_i^{(j)} \mid i=\overline{1,k_j}\} , \qquad j=1,2.$$

Then

$$L(F_1) = L(F_2) .$$

Proof. Suppose $L(F_1) \neq L(F_2)$. Without loss of generality, assume that there exists $A \in L(F_1)$ and $A \bar{\in} L(F_2)$. It follows that, there is

$$L_i^{(1)} \rightarrow R_i^{(1)} \text{ with } L_i^{(1)} = AX,\ X \neq \emptyset .$$

Since $A \bar{\in} L(F_2)$, by lemma 1.2. and from $AX \rightarrow R_i^{(1)} \in F_2^+$, we have

$$X \rightarrow R_i^{(1)} \in F_1^+ \quad (\text{since } F_2^+ = F_1^+) .$$

This means that in F_1 we can replace $AX \to R_i^{(1)}$ by $X \to R_i^{(1)}$ without altering F_1^+.

We arrive to a contradiction because, by the hypothesis, F_1 is left reduced. Thus, we have: $L(F_1) = L(F_2)$,

Q.E.D.

Remark 3.4. Proofs of theorems 3.2. and 3.3. that we gave above are substantial improvements and are much shorter than the proofs given in [3].

ACKNOWLEDGEMENT

The author would like to thank Prof.Dr.Janos Demetrovics for his valuable remarks and suggestions.

REFERENCES

[1] Maier,D. Minimum covers in the relational database model. J.ACM 27 (Oct 1980), 664-674.

[2] Ausiello, G. et al. Graphs algorithms for functional dependency manipulation. J.ACM 30 (Oct.1983), 752-766.

[3] Tran Thai Son and Dinh ngoc Thanh. Some results about the structure of minimum covers in the relational database model (private communication).

[4] Ho Thuan and Le van Bao. Some results about keys of relational schemas, Acta Cybernetica, Tom.7. Fasc.1, Szeged 1985, 99-113.

[5] Ullman,J.D. Principles of Database systems Computer Science Press, Potomac, Md, Second edition, 1982.

[6] Armstrong, W. On the generation of dependency structures of relational databases. Pub. 272 Université de Montréal 1977.

Funkcionális függőségek lefedéseire vonatkozó invariánsok

Ho Thuan

Összefoglaló

A cikkben a lefedésekre ill. nem-redundáns lefedésekre vonatkozó néhány ujabb invariánsról van szó. Az eredmények többsége már [3]-ban megvolt, de itt a szerző az eredményeket gyengébb feltételekkel bizonyitja és bizonyitásai egyszerübbek is.

Некоторые инварианты покрытий для функциональных зависимостей

Хо Тхуан

Р е з ю м е

В статье доказываются некоторые инварианты для покрытий и не лишних покрытий. Большинство этих результатов были изложены в [3], но здесь доказываются те самые результаты с ослабленными предположениями и данные доказательства более простые.

MTA SZTAKI Közlemények 34/1986 pp.163-173

DIRECT DETERMINATION AND FD-GRAPH

HO THUAN

MTA SZTAKI

ABSTRACT

The notions of direct determination and FD-graph were introduced by D. Maier and by G. Ausiello et al. respectively [1,2] to study the structure of minimum covers. However, the relationship between these two notions are not explicitly presented.

In this paper, we establish the relation between FD-graph and direct determination and prove some well-known and new properties concerning direct determination. A remark on the definition of FD-graph is also given.

1. INTRODUCTION

The notion of direct determination was introduced by D. Maier in [1] to study the structure of minimum covers. Using direct determination he showed it is possible to find covers with the smallest number of functional dependencies (FD) in polynomial time.

In [2], G. Ausiello et al. presented an approach which is based on the representation of the set of FDs by FD-graph (a generalization of graphs). Such a representation provides a unified framework for the treatment of various properties and for the manipulation of FDs.

However, the notion of direct determination in its relationship with FD-graph is not explicitly presented.

In this paper we establish the relation between FD-graph and direct determination, and prove some well-known and new properties concerning direct determination.

A few comments on the definition of FD-graph are also given.

The reader is required to know the basic notation of the relational model and FDs [3].

§2.

In this section we recall some definitions and results which will be needed in the sequel.
Without loss of generality, we use only sets of FDs in the natural reduced form [4].

Throughout this paper we assume that all attributes are chosen from a fixed universe Ω.

Definition 2.1. The set of attributes X and Y are equivalent under a set of FDs F, written $X \leftrightarrow Y$, if $X \to Y$ and $Y \to X$ are in F^+.

Definition 2.2 [1] Given a set of FDs F with $X \to Y$ in F^+, X directly determines Y under F, written $X \overset{\bullet}{\to} Y$, if $X \to Y \in [F \backslash E_F(X)]^+$, where $E_F(X)$ is the set of all FDs in F with left sides equivalent to X.
That is, no FDs with left sides equivalent to X are used to derive $X \to Y$.

Definition 2.3 [2] Given a set of FDs F on Ω, the FD-graph $G_F = \langle V, E \rangle$ associated with F is the graph with node labeling function w: $V \to P(\Omega)$ and arc labeling function

w': $E \to \{0,1\}$ such that

(i) for every attribute $A \in \Omega$, there is a node in V labeled A (called a simple node);

(ii) for every dependency $X \to Y$ in F where $\|X\| > 1$, there is a node in V labeled X (called a compound node);

(iii) for every dependency $X \to Y$ in F where $Y = A_1 \ldots A_k$ there are arcs labeled 0 (full arcs) from the node labeled X to the nodes labeled $A_1, \ldots, A_K$;

(iv) for every compound node i in V labeled $A_1 \ldots A_k$ there are arcs labeled 1 (dotted arcs) from the node i to all simple nodes (component nodes of i) labeled $A_1, \ldots, A_k$.

The set of full arcs (dotted arcs) is denoted E_o (E_1).

Definition 2.4 [2] Given an FD-graph $G_F = <V,E>$ and two nodes $i,j \in V$, a (directed) FD-path $<i,j>$ from i to j is a *minimal subgraph* $\bar{G}_F = <\bar{V},\bar{E}>$ of G_F such that $i,j \in \bar{V}$ and either $(i,j) \in \bar{E}$ or one of the following possibilities holds:

(a) j is a simple node and there exists a node k such that $(k,j) \in \bar{E}$ and there is an FD-path $<i,k>$ included in $\bar{G}_F$ (graph transitivity);

(b) j is a compound node with component nodes $m_1, \ldots, m_r$ and there are dotted arcs $(j,m_1), \ldots, (j,m_r)$ in $\bar{G}_F$ and r FD-paths $<i,m_1>, \ldots, <i,m_r>$ included in $\bar{G}_\Sigma$ (graph union).

Furthermore an FD-path $<i,j>$ is dotted if all its arcs leaving i are dotted; otherwise it is full.

Definition 2.5. [2] The closure of an FD-graph $G_F = \langle V,E\rangle$ is the graph $G_F^+ = \langle V,E^+\rangle$, labeled on the nodes and on the arcs, where the set V is the same as in G_F, while the set $E^+ = (E^+)_0 \cup (E^+)_1$ is defined in the following way:

$(E^+)_1 = \{(i,j) \mid i,j \in V$, and there exists a dotted FD-path $\langle i,j\rangle\}$

$(E^+)_0 = \{(i,j) \mid i,j \in V,\ (i,j) \notin (E^+)_1$ and there exists a full FD-path $\langle i,j\rangle\}$.

Definition 2.6 [2] Two nodes i,j in an FD-graph are said equivalent if the arcs (i,j) and (j,i) both belong to the closure of G_F. Furthermore a node i of G_Σ is said to be equivalent to a node j of $G_{\overline{\Sigma}}$ where $G_{\overline{\Sigma}}$ is a cover of G_Σ (i.e. $\Sigma^+ = \overline{\Sigma}^+$), if i,j are equivalent in some cover of G_Σ.

Theorem 2.1 [2] Let $G_F = \langle V,E\rangle$ be the FD-graph associated with the set F of FDs, and let $G_F^+ = \langle V,E^+\rangle$ be its closure. An arc (i,j) is in E^+ if and only if $w(i) \to w(j)$ is in F^+.

Remark 2.1 The definition 2.3. is reasonable and concise in the sense that the FD-graph G_F includes all the "meaning part" of the closure of the set of FDs. On the other hand, with the FD-graph we can provide a simple and unified treatment of all properties of sets of FDs.

By the definition of a FD-graph, it is clear that *every compound mode has at least one outgoing full arc.*

However in[2, page 755] we found the following observation:

"Finally we may observe that by definition of FD-path, a *compound node without outgoing full arcs* can only be either a source or a target node of FD-paths to which it belongs"!

Part (ii) of lemma 1 [2, p. 757] arises the same problem. Let us see it:

"(ii) If $G_{\overline{\Sigma}}$ is a subgraph of G_Σ such that all arcs in

$E-\bar{E}$ are dotted (i.e., G_Σ may contain compound nodes not in $G_{\bar{\Sigma}}$ but no more full arcs) and (i,j) is in $(E^+)_0[(E^+)_1]$, then (i,j) is in $(\bar{E}^+)_0\ [(\bar{E}^+)_1]$". (where i,j are two nodes belonging to both V and $\bar{V}$).

It is obvious that, strictly following the definition 2.3, there is no possibility that G_Σ may contain compound nodes not in $G_{\bar{\Sigma}}$ but no more full arcs. And it is easy to show that under these conditions the subgraph $G_{\bar{\Sigma}}$ coincides with G_Σ. In that case, part (ii) of lemma 1 is trivial. How to overcome theses difficulties? A natural way is to think that a FD-graph $G_F = <V,E>$ associated with F is defined precisely to an arbitrary finite number of different compound nodes which are not correspond to the left side of any FD in F, together with the dotted arcs from each of them to its corresponding component nodes.

In our opinion, the view just presented above must be mentioned after introducing the definition of an FD-graph, In so doing, according to the necessity, we can freely add to an FD-graph some new compound nodes without outgoing full arcs if it makes easy the proof of a certain required propriety. In fact, this technique was often used by the authors of [2].

By the above reasons, it would be better to remove part (ii) of lemma 1 in [2], changing it into a remark.

§3.

We are now in a position to establish the relationship between direct determination and FD-graph, and to prove some well-known and new properties concerning direct determination.

Definition 3.1. Given an FD-graph $G_F = <V,E>$ and a node $i \in V$ with at least a full outgoing arc. A strong component of G_F with representative node i is a maximal set of pair-wise equivalent nodes which contains i, denoted by SC(i).

Notice that every node in SC(i) has at least one full outgoing arc.

The following lemma is obvious.

Lemma 3.1. Given an FD-graph $G_F = \langle V,E \rangle$, a node $i \in V$, its corresponding strong component SC(i) and two nodes j,k such that j is equivalent to i. (j is not necessary belong to SC(i), i.e. j can be a compound node without outgoing full arc that we add it to the FD-graph. The same situation can be happen to the node k too).

Then $w(j) \dot{\rightarrow} w(k)$ if and only if there exists a dotted FD-path $\langle j,k \rangle$ containing *no full outgoing arc from any node of SC(i)*.*)

In that case, for sake of simplicity, we write $\langle j \overset{SC(i)}{\not\longrightarrow} k \rangle$.

Example 3.1. Given Ω = ABCDEI

$$F = \{A \rightarrow BC, \quad BC \rightarrow A, \quad AD \rightarrow EI, \quad EA \rightarrow ID\}$$

It is easy to verify that

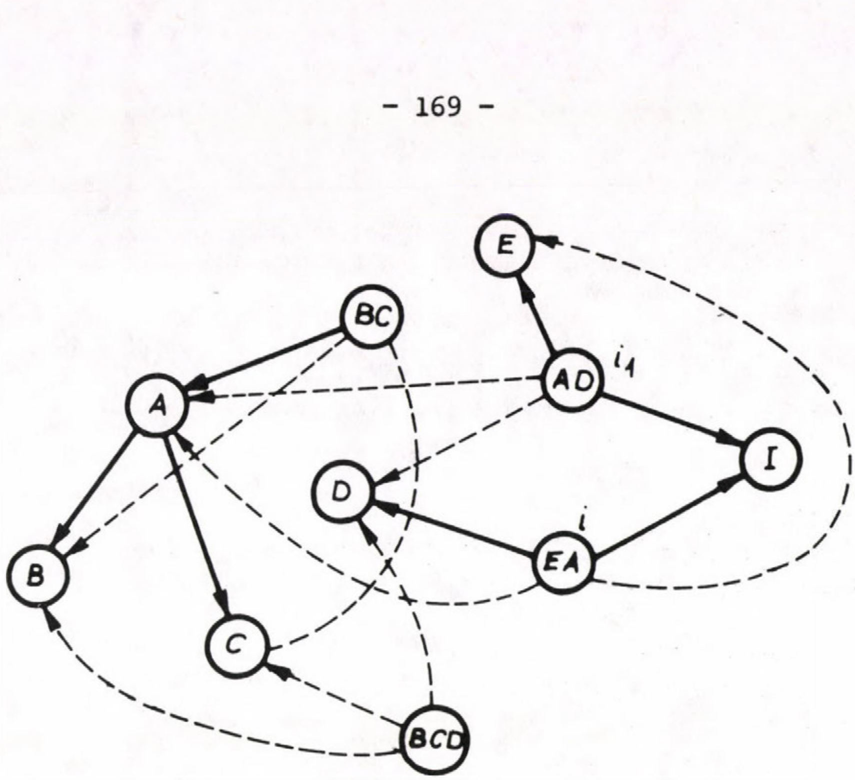
$$E_F(AD) = \{AD \rightarrow EI, \quad AE \rightarrow DI\}$$

and

$$BCD \longleftrightarrow AD$$

The corresponding FD-graph with an added node BCD (without outgoing full arc) is shown in the following figure:

*) In other words, the dotted FD-path $\langle j,k \rangle$ contains no intermediate nodes that are nodes in SC(i).



$SC(i_1) = \{i_1, i_2\}$ where $w(i_1) = AD$, $w(i_2) = EA$.

We find that $BCD \dot{\rightarrow} A$ and

$BCD \dot{\rightarrow} AD$

Lemma 3.2. Given an FD-graph $G_F = \langle V,E \rangle$, two equivalent nodes $i,j \in V$ and i_1, j_1 are two nodes equivalent to i and j respectively.

If $\langle i_1 \overset{SC(i)}{\not\longrightarrow} j_1 \rangle$ and $\langle j_1 \overset{SC(j)}{\not\longrightarrow} k \rangle$

then

$$\langle i_1 \overset{SC(i)}{\not\longrightarrow} k \rangle.$$

Proof. Since i and j are equivalent nodes, we have

$$SC(i) = SC(j)$$

Merge two FD-paths $\langle i_1 \overset{SC(i)}{\not\longrightarrow} j_1 \rangle$ and $\langle j_1 \overset{SC(i)}{\not\longrightarrow} k \rangle$ appropriately at component nodes of j_1 which are intermediate nodes of the FD-path $\langle j_1,k \rangle$, we obtain the FD-path $\langle i_1 \overset{SC(i)}{\not\longrightarrow} k \rangle$.

In the other words, from

$$w(i) \leftrightarrow w(i_1) \leftrightarrow w(j_1),\ w(i_1) \dot{\rightarrow} w(j_1),\ w(j_1) \dot{\rightarrow} w(k)$$

we have $w(i_1) \overset{\bullet}{\rightarrow} w(k)$.

Notice that the above lemma corresponds to [1, Lemma 5].

Lemma 3.3. Given an FD-graph $G_F = \langle V,E \rangle$, $i \in V$ is a node having at least one outgoing full arc and i_o is equivalent to i (i_o can be an added node to the FD-graph and without outgoing full arc).
Then there exists $j \in SC(i)$ such that $\langle i_o \xrightarrow{SC(i)} j \rangle$

Proof. Suppose that $i_o \notin SC(i)$. Otherwise, take $j \equiv i_o$ and the lemma is proved.

Consider the dotted FD-path $\langle i_o, i \rangle$. In the case, there is no intermediate node in $\langle i_o, i \rangle$ that is node of $SC(i)$ then i is the node to be found. Otherwise, suppose $i_1 \in SC(i)$ is an intermediate node of $\langle i_o, i \rangle$. Now we have only to consider the FD-path $\langle i_o, i_1 \rangle$. Repeat the above reasoning for $\langle i_o, i_1 \rangle$. Finally, we will find the required j such that $\langle i_o \xrightarrow{SC(i)} j \rangle$. Q.E.D.

Notice that the above lemma corresponds to [1, lemma 6].

Lemma 3.4. Let $G_F = \langle V,E \rangle$ be a minimum FD-graph (i.e. F is minimum), and $i \in V$ is a node with at least one outgoing full arc. Then in SC(i) there exists no $j_1, j_2, j_1 \neq j_2$ such that $\langle j_1 \xrightarrow{SC(i)} j_2 \rangle$.

Proof. Assume the contrary that there exists $j_1, j_2 \in SC(i)$, $j_1 \neq j_2$ such that there is a dotted FD-path from j_1 to j_2. Since j_2 is equivalent to j_1, j_1 is a superfluous node. We arrive to a contradiction [2, Theorem 3].

Definition 3.2. [2] An FD-graph G_F is nonredundant if F is nonredundant.

Given two FD-graphs G_{F_1} and G_{F_2}, G_{F_2} is a cover of G_{F_1} if

F_2 is a cover of F_1 (i.e. $F_1^+ = F_2^+$).

Lemma 3.5. Given two nonredundant FD-graphs G_{F_1} and G_{F_2} where G_{F_2} is a cover of G_{F_1},

$$G_{F_1} = \langle V_1, E_1 \rangle, \quad G_{F_2} = \langle V_2, E_2 \rangle$$

Let i_1 and i_2 be two equivalent nodes in E_1 and E_2 respectively with at least one outgoing full arc, (p_2, q_2) be a full arc of E_2 with $p_2 \notin SC_2(i_2)$.

If $(i_1, p_2) \in E_2^+$ then $\langle p_2 \xrightarrow{\;SC_1(i_1)\;}\!\!\!\!\!\!\!\!/\;\;\; q_2 \rangle$. *)

Proof. Since $(i_1, p_2) \in E_2^+$, by theorem 2.1, there is a FD-path in G_{F_1} from i_1 to p_2.
Now, assume the contrary that the FD-path in G_{F_1} from p_2 to q_2 has an immediate node $j_1 \in SC(i_1)$. The presence of the FD-path $\langle j_1, i_1 \rangle$ shows that p_2 is equivalent to i_1, i.e. $p_2 \in SC_2(i_2)$, a contradiction. Q.E.D.

Theorem 3.1. With the same assumptions as in lemma 3.5, if we replace in G_{F_1} all nodes belonging to $SC_1(i_1)$ together with their corresponding outgoing arcs by all nodes in $SC_2(i_2)$ together with their corresponidng outgoing arcs, then the new FD-graph is a cover of G_{F_1}.

Proof. We have only to prove that for every full arc $(j_1, k_1) \in E_1$ with $j_1 \in SC_1(i_1)$ there is a FD-path $\langle j_1, k_1 \rangle$ in the new FD-graph. By the lemma 3.5 we have just the required result.

Remark 3.1. Theorem 3.1 can be formulated in an another form as follows:

If F_1, F_2 are nonredundant and equivalent set of FD_s, then

$$\{F_1 \setminus E_{F_1}(X)\} \cup E_{F_2}(X) \equiv \{F_2 \setminus E_{F_2}(X)\} \cup E_{F_1}(X).$$

And this is the content of a theorem in [5].

*) The subcripts 1,2 correspond to G_{F_1}, G_{F_2} respectively

ACKNOWLEDGEMENT

The author would like to thank *Prof. Dr. J. DEMETROVICS* for his help and encouragement.

REFERENCES

[1] Maier, D. Minimum covers in the relational database model. J.ACM 27, 4(October 1980),664-674.

[2] Ausiello, G. et al. Graph algorithms for functional dependency manipulation. J.ACM 30, 4(Oct. 1983), 752-766.

[3] Ullman, J.D. Principles of Database Systems. Computer Science Press, Potomac, Md. Second edition, 1982.

[4] Ho Thuan, Some invariants of covers for functional dependencies. Közlemények 34/1986 MTA SZTAKI, Budapest.

[5] Tran thai Son and Dinhngoc Thanh, Some results about the structure of minimum covers in the relational database model. (Private communication).

Direkt meghatározás és FD-gráf

Ho Thuan

Összefoglaló

A cikk a direkt meghatározás és FD-gráfok közötti kapcsolatra mutat rá és néhány ismert és uj eredményt bizonyit. A cikkben az FD-gráfok definiciójára vonatkozó néhány megjegyzés is megtalálható.

Непосредственная детерминация и FD-граф

Хо Тхуан

Р е з ю м е

В статье устанавливается связь между понятиями FD-графа и непосредственной детерминации и доказываются некоторые связанные знакомые и новые результаты. Тоже задано замечание к определению FD-графа.

MTA SZTAKI Közlemények 34/1986 pp.175-184

LATTICE POINTS IN DIFFERENCE SETS

B. UHRIN

MTA SZTAKI

1. Let A be any Lebesgue-measurable set in R^n. By the difference set of A we mean the algebraic difference of A with itself. We shall denote this set by DA, i.e. $DA := A - A$. DA is clearly symmetric (about the origin), i.e. $DA = -DA$. The most familiar example for a difference set is any symmetric (about Θ) convex set K, because $K = D(\frac{1}{2}K)$. Denote by $\Lambda \subset R^n$ the set of points having integer coordinates (lattice points). The quotient space R^n/Λ is usually identified with the set $P := \{x \in R^n : 0 \leq x_i < 1, \quad i = 1,\ldots,n\}$. The cardinality of the finite set $S \subset R^n$ will be denoted by $|S|$.
The present note studies the question: what is the connection between $|DA \cap \Lambda|$ and the volume (L-measure) $V(A)$ of A? It is intuitively clear that if $V(A)$ is too small then we cannot in general expect that DA will contain non-zero lattice points. For example the open unit hypercube $C := \{x \in R^n: |x_i| < 1, \quad i = 1,\ldots,n\}$ has the volume $V(C)=2^n$, but the only lattice point in C is the origin Θ. However if we increase this set a little, i.e. take the set $C' := \{x: |x_i| < 1+\varepsilon, \quad i=1,\ldots,n\}$, where $\varepsilon > 0$ is arbitrarily small, then $V(C') > 2^n$ and $|C' \cap \Lambda| = 3^n$.
The latter example reflects a more general rule: if for any symmetric (about Θ) convex set K we have $V(K) > 2^n$ then $|K \cap \Lambda| > 1$. (This is the well known Minkowski's convex body theorem, see [1].) Taking into account that $V(K)/2^n = V(\frac{1}{2}K)$, we can formulate this statement in terms of $D(\frac{1}{2}K)$:
$V(\frac{1}{2}K) > 1 \Rightarrow |D(\frac{1}{2}K) \cap \Omega| > 1$.

AMS Classification Numbers (1980): 10E05, 10E30.

This result proved to be true for any measurable set A: if $V(A) > 1$ then $|DA \cap \Lambda| > 1$. (This is the generalization of Minkowski's theorem due to Blichfeldt, see [1]).
On the other hand, let us look at the set S on Figure 1.

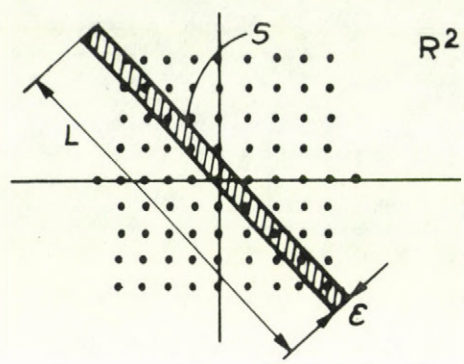


Figure 1

It is symmetric and convex, its volume is εL, but it contains $\sim L/\sqrt{2}$ lattice points. So we have a set of "almost zero volume" containing "almost infinitely many lattice points".
The basic aim of this note is to give a rule which sharpens the Blichfeldt's result and which works also in the above case of "thin sets of small volume containing many lattice points".
The second aim is to present a new method of proof which seems to be at most natural and simple when compared with known ones.

2. The basic tool in our investigations will be the so called lexicographic ordering of points in R^n, denoted by "$\succ$". It is defined as follows: $x \succ \Theta$ if and only if either $x_1 > 0$ or $x_1 = x_2 = .. = x_i = 0$ and $x_{i+1} > 0$ for some $1 \leq i < n$.
Of course, $x \succ y$ means $x-y \succ \Theta$. This is a total (linear) ordering in R^n consistent with the addition of vectors.
Let $H \subset R^n$, be any finite set containing the zero vector Θ and symmetric about Θ (i.e. $H = -H$). If we want to list all elements of H, we can proceed in the following way:

First take Θ. Secondly, take $h_1 \in H$ that is the first positive (in the ordering $\succ$) element of H (i.e. $h_1 \succ \Theta$ and there is no $h \in H$ such that $h_1 \succ h \succ \Theta$). It is clear that $-h_1 \in H$ and $-h_1$ is the first negative element of H ($-h_1 \prec \Theta$ and there is no $h \in H$ s.t. $-h_1 \prec h \prec \Theta$). Take the second positive element of H, h_2, (i.e. $h_2 \succ h_1$ and there is no $h \in H$ s.t. $h_2 \succ h \succ h_1$). Again $-h_2$ is the second negative element of H. And so on. In this way we list all elements of H and we get

$$H = \{-h_q \prec -h_{q-1} \prec \dots -h_1 \prec h_o = \Theta \prec h_1 \prec h_2 \prec \dots \prec h_q\}.$$

At the same time we have just also proved that $|H|$ is an odd number. Now, let $A \subset R^n$ be an arbitrary measurable and bounded set. Applying the previous remark, we have for some $p \geq 1$

$$(1) \quad DA \cap \Lambda = \{-b_{p-1} \prec -b_{p-2} \prec \dots \prec -b_1 \prec \Theta \prec b_1 \prec b_2 \prec \dots \prec b_{p-1}\},$$

showing that

$$(2) \qquad |DA \cap \Lambda| = 2p-1.$$

The relation (2) implies that

$$(3) \qquad |A \cap \Lambda| \leq p.$$

Indeed, assume that $q = |A \cap \Lambda| > p$. Writing $A \cap \Lambda$ in the order $\succ$, we have $A \cap \Lambda = \{a_1 \prec a_2 \prec \dots \prec a_q\}$. The elements $a_1-a_q \prec a_1-a_{q-1} \prec \dots \prec a_1-a_2 \prec \Theta \prec a_2-a_1 \prec a_3-a_1 \prec \dots \prec a_q-a_1$ are mutually different and all belong to $DA \cap \Lambda$. This implies that $|DA \cap \Lambda| \geq 2q-1 > 2p-1$ that contradicts to (2).
In general $A \cap \Lambda$ may be empty (hence $|A \cap \Lambda| = 0$) but (2) may hold with some $p > 1$. The second "extreme" case of (3) is when $|A \cap \Lambda| = p$. This implies quite strict conditions on the structure of $A \cap \Lambda$, namely we have

Proposition 1. Assume (2) holds and $|A \cap \Lambda| = p$. Then $A \cap \Lambda = \{a, a+d, a+2d, \dots, a+(p-1)d\}$ for some $a \in \Lambda$ and $d \succ \Theta$ (i.e. $A \cap \Lambda$ is an arithmetic progression in the lexicographic ordering). □

The proposition will be a simple consequence of the following.

Lemma. Let $S, H \subset R^n$ be two finite non-empty sets. Let $S+H$ be their algebraic sum. Then

$$|S+H| \geq |S| + |H| - 1 \tag{5}$$

and equality is in (5) if and only if S and H are of the following form: $S = \{s, s+d, \dots, s+(r-1)d\}$, $H = \{h, h+d, \dots, h+(q-1)d\}$, where $d \succ \Theta$, $r = |S|$, $q = |H|$. □

Proof. Write S and H in the order $\succ$, say, $S = \{s_o \prec s_1 \prec \dots \prec s_{r-1}\}$, $H = \{h_o \prec h_1 \prec \dots \prec h_{q-1}\}$. Now we have

$$s_o+h_o \prec s_o+h_1 \dots \prec s_o+h_{q-1} \prec s_1+h_{q-1} \prec \dots \prec s_{r-1}+h_{q-1}, \tag{6}$$

which proves (5).

The "if" part of the equality statement is clear. As to the "only if" part, denote $s_{ij} := s_i + h_j$ and $L_j := \{s_{oo}, s_{o1}, \dots, s_{oj}, s_{1j}, s_{2j}, \dots, s_{r-1,j}, s_{r-1,j+1}, \dots, s_{r-1,q-1}\}$, $j = 0, 1, \dots, q-1$. Each L_j consists of lexicographically increasing sequence of elements, begins with s_{oo} and ends with $s_{r-1,q-1}$. This implies, using the assumption $|S+H| = r+q-1$, that each L_j represent the whole set $S+H$. This further implies that the k-th elements of any two L_j-s are equal, $k = 1, 2, \dots, r+q-1$. So we have for L_{j-1}, L_j and L_{j+1}: $s_i + h_{j-1} = s_{i-1} + h_j$, $s_{i+1} + h_{j-1} = s_i + h_j$, $s_{i+1} + h_j = s_i + h_{j+1}$. This implies: $d = s_i - s_{i-1} =$
$= s_{i+1} - s_i = h_j - h_{j-1} = h_{j+1} - h_j$.

Here i and j were arbitrary, so the lemma is proved. ■

Proof of the proposition: It is clear that

(6) $$\mathcal{D}A \cap \Lambda \supseteq \mathcal{D}(A \cap \Lambda).$$

Using (5) with $S := A \cap \Lambda$, $H := -(A \cap \Lambda)$, we have

(7) $$|\mathcal{D}A \cap \Lambda| \geq |\mathcal{D}(A \cap \Lambda)| \geq 2p - 1,$$

hence

(8) $$|\mathcal{D}(A \cap \Lambda)| = 2\ |A \cap \Lambda| - 1.$$

The lemma now implies the result. ■

The relation (2) implies a more general inequality, containing (3) as a special case. Namely, it is cleat that $\mathcal{D}A$ is invariant under the translations of A, i.e. $\mathcal{D}(A-x) = \mathcal{D}A$ for all $x \in R^n$. So (2) implies

(9) $$|\mathcal{D}A \cap \Lambda| = |\mathcal{D}(A-x) \cap \Lambda| = 2p - 1,$$

hence all results proved above hold for $A-x$ instead of A, say:

(10) $$|(A - x) \cap \Lambda| \leq p.$$

The function $|(A-x) \cap \Lambda|$ (of x) is periodic modΛ, i.e. $|(A-x+u) \cap \Lambda| = |(A-x) \cap \Lambda|$, $\forall x$, $\forall u$, hence when dealing with it we can restrict ourselves to the set P (a basic cell of Λ). (9) and (10) give

Proposition 2. For any set $A \subset R^n$ we have

(11) $$|\mathcal{D}A \cap \Lambda| \geq 2 . \max_{x \in P} |(A-x) \cap \Lambda| - 1. \qquad \square$$

3. Let us study now the volume (L-measure) $V(A)$ of a L-measurable bounded set $A \subset R^n$. Using Λ, we get two decompositions of A:

(12) $$A = \bigcup_{u \in \Lambda} ((P+u) \cap A) \quad ,$$

(13) $$A = \bigcup_{x \in P} (A \cap (\Lambda+x)) \quad .$$

In both decompositions the sets in the union are mutually disjoint. In (12) only finite number of sets are non-empty, hence

(14) $$V(A) = \sum_{u \in \Lambda} V((P+u) \cap A)$$

(where the sum is finite).

In (13) each $A \cap (\Lambda+x)$ is finite, so we have to collect many of them to get sets of positive measure. This can be done, say, in such a way that we put together the sets $A \cap (\Lambda+x)$ having the same cardinality.
For this denote

(15) $$A_i := \{x \in P : |A \cap (\Lambda+x)| = i\}, \quad i=0,1,2,\ldots \; .$$

For $x \in A_i$, $i > 0$, the set $A \cap (\Lambda+x)$ is of the form

(16) $$\{a_1(x) = u_1(x) + x,\ a_2(x) = u_2(x)+x, \ldots, a_i(x) = u_i(x)+x\}.$$

Let us partition A_i further according to the rule: $x,y \in A_i$ belong to the same set if $u_j(x) = u_j(y)$ for all $j=1,2,\ldots,i$. This will give a finite partition of A_i, say A_{ik}, $k = 1,2,\ldots,N(i)$.
Hence we can write

$$(17)\qquad A = \bigcup_{i=1}^{\infty} \bigcup_{j=1}^{N(i)} \bigcup_{x \in A_{ij}} (A \cap (\Lambda + x)).$$

Clearly, there are $u_{1j}, u_{2j}, \ldots, u_{ij} \in \Lambda$ such that

$$(18)\qquad \bigcup_{x \in A_{ij}} (A \cap (\Lambda + x)) = \bigcup_{k=1}^{i} (A_{ij} + u_{kj}).$$

Substituting (18) into (17) we get

$$(19)\qquad V(A) = \sum_{i=1}^{\infty} i \cdot V(A_i).$$

The set

$$(20)\qquad \varphi(A) := \bigcup_{i=1}^{\infty} A_i = \{x \in P: A \cap (\Lambda + x) \neq \emptyset\}$$

plays an important role in the algebraic theory of R^n. It is nothing else then the canonical map of A into the tours group R^n/Λ (after identifying R^n/Λ with P). This mapping is important also in our investigations. The quantity

$$(21)\qquad V(\varphi(A)) = \sum_{i=1}^{\infty} V(A_i)$$

is the measure of the projection of A into R^n/Λ.

4. Now we put together the results of previous two sections to get main results of this note. It is clear that $|(A-x) \cap \Lambda| = |A \cap (\Lambda + x)|$ for all $x \in P$.

Proposition 3. For any bounded L-measurable set $A \subset R^n$ we have

$$(22)\qquad |DA \cap \Lambda| \cdot V(\varphi(A)) \geq 2\, V(A) - V(\varphi(A)).$$

Proof. (19) and (21) shows that $V(A) = 0$ if and only if $V(\varphi(A)) = 0$. Let $V(A) > 0$. Inequality (11) implies that

$$(23) \qquad |DA \cap \Lambda| \geq 2i - 1$$

holds for all i such that $V(A_i) > 0$. Hence

$$(24) \qquad |DA \cap \Lambda| \cdot V(A_i) \geq 2i\, V(A_i) - V(A_i).$$

Summing up both sides of (24) we get (22). ■

Proposition 4. Let A be as in Prop.3. If for some positive integer m we have $V(A)/V(\varphi(A)) > m$, then there are $u_i \in \Lambda$, $i = 1,\dots,m$, such that

$$(25) \qquad \{-u_m \prec -u_{m-1} \prec \dots \prec -u_1 \prec \Theta \prec u_1 \prec u_2 \dots \prec u_m\} \subseteq DA \cap \Lambda. \quad \square$$

Proof. Using (22) and (2) we see that $p-1 \geq m$, so (1) gives the result. □

This result is clearly sharper and more "exact" than that of Blichfeldt. Moreover, sets of the type seen on Fig.1 are succesfully treated by this rule. For such sets $V(A)$ may be very small, but $V(\varphi(A))$ is necessarily smaller (see (19) and (21)), so that their ratio may be quite big. For the investigation of this ratio the representations (19) and (21) are very useful.

5. In the course of our proofs we have used two structural properties of R^n only: it is an Abelian group having a total (linear) ordering consistent with the group operation.
Hence all results are true in the same form for any topological Abelian group G and its discrete subgroup Λ, assuming that Λ can be totally ordered. (Of course we have also to assume that the quotient group G/Λ is compact.) In this case V means the Haar-measure in G.

The method sketched in this note can be succesfully developped to prove some estimations in geometry of numbers which are both more general and sharper than the known ones, see [2],[3].

REFERENCES

[1] C.G. Lekkerkerker, "Geometry of Numbers", Wolters-Noordhoff, Gronningen, North-Holland, Amsterdam, 1969.

[2] B. Uhrin, Some useful estimations in geometry of numbers, Period. Math. Hungar., 11 (2), (1980), 95-103.

[3] B. Uhrin, On a generalization of Minkowski's convex body theorem, J. of Number Theory, 13 (2), (1981), 192-209.

Differencia-halmazokban levő rácspontok

Uhrin Béla

Összefoglaló

A cikkben az $(A-A)\cap\Lambda$ halmaz számosságára vonatkozó alsó becslésektől van szó, ahol $A \subset R^n$ L-mérhető halmaz és Λ egy pontrács (diszkrét részcsoport) az R^n-ben. Az eredmények élesitik a klasszikus Minkowski-Rlichfeldt tételt a geometriai számelméletben.

Решеточные точки в разностных множествах

Б. Ухрин

Р е з ю м е

В статье изучаются нижние оценки для мощности множества $(A-A)\cap\Lambda$, где $A \subset R^n$ измеримое множество и Λ есть решетка /дискретная подгруппа/ в R^n. Результаты уточняют классический результат Минковского-Блихфелдта в геометрии чисел.