MTA SZTAKI Közlemények 33/1985 p. 7-27

ТАБЛИЦЫ КОНЪЮНКТИВНЫХ ЗАПРОСОВ И ИХ ПРИМЕНЕНИЕ

Галя Ангелова

Лаборатория математической лингвистики

Институт математики с ВЦ, БАН

София, Болгария

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 σ (селекцию), π (проекцию) и J (соединение). 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 сильной и слабой эквивалентности двух заданных SPJ - выражений. Так как между конъюнктивными реляционными запросами и SPJ - выражениями существует взаимно-однозначное соответствие, то при помощи таблиц конъюнктивных запросов можно исследовать проблему сильной и слабой эквивалентности данных конъюнктивных запросов. Понятие таблици играет важную роль и в алгоритме интерпретации запросов пользователя в System/U (/6/ и /7/), где оно применяется для нахождения оптимального внутреннего представления запроса, которое является слабоэквивалентным начальному представлению запроса.

2. ДЕФИНИЦИИ И ПРИМЕРЫ

Будем использовать дефиниции операций селекции, проекции и (естественного) соединения:

Пусть r - отношение над множеством атрибутов X и $A \in X$, $a \in A$, $Y \subseteq X$. Тогда:

- селекция $A \theta a$, обозначаемая через $\sigma_{A\theta a}(r)$, представляет собой:

$$\sigma_{A\theta a}(r) = \{\alpha / \alpha \in r \text{ и } \alpha(A)\ \theta\ a\}.$$

(Здесь θ - одно из $=, <, >, \leq, \geq, \neq$). Таким образом из отношения r берутся только те кортежи, имеющие в атрибуте A значение b и $b \theta a$. Так $\sigma_{A\theta a}(r)$ является отношением над множеством атрибутов X и следовательно представляет собой подмножество отношения r;

- проекция $\pi_Y(r)$ представляет собой:

$$\pi_Y(r) = \{\beta[Y] / \beta \in r\},$$

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

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

Пусть R_1 и R_2 являются реляционными схемами, а r_1 и r_2 - отношения над этими реляционными схемами. Тогда

$$r_1 \bowtie r_2 = \{\gamma / \gamma \text{ является кортежом над атрибутами } R_1 \cup R_2 \text{ и существуют кортежи } v_1 \in r_1 \text{ и } v_2 \in r_2\text{, такие, что } v_1 = \gamma[R_1] \text{ и } v_2 = \gamma[R_2]\}.$$

Дефиниция 1. 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 S, P, J и именами реляционных схем.

Дефиниция 2. Конъюнктивным запросом в реляционном языке запросов будем называть запрос вида:

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

где P_i, $1 \leq i \leq k$ – термы вида а) или вида б):

а) $R(c_1 c_2 \ldots c_s)$, что означает, что кортеж $c_1 c_2 \ldots c_s$ принадлежит отношению над реляционной схемой R. Здесь c_j, $1 \leq j \leq s$ являются константами соответствующего домена или $c_j \in \{a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m\}$ – (т.е. c_j – символы среди $a_1, a_2, \ldots, a_n, b_1, \ldots, b_m$);

б) $c \theta d$, где c и d – либо константы, либо элементы множества $\{a_1, a_2, \ldots, a_n, b_1, \ldots, b_m\}$. Здесь θ – одно из $=, <, >, \leq, \geq$.

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. Отношение ОБЯЗАННОСТЬ дает информацию об обязанностях, присущих каждому поставщику.

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

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

Этот запрос можно представить и следующим образом:

(2) $$\{a_1 / (\exists b_1)(\exists b_2) : \text{ПОСТАВЩИК}(a_1, b_1, b_2, c_1)\}.$$

Как известно, между конъюнктивными запросами и SPJ - выражениями существует взаимно-однозначное соответствие, т.е. каждый конъюнктивный запрос может быть представлен как SPJ - выражение и наоборот, каждому SPJ - выражению соответствует конъюнктивный запрос (см. /7/). По этой причине мы будем строить таблицы для SPJ - выражений и часто будем интерпретироват эти таблицы как конъюнктивные запросы.

Дефиниция 3. Введем понятие таблицы:
Каждая таблица представляет собой двумерную матрицу, причем к этой матрице можно задавать и список ограничений. Столбцы матрицы соответствуют заданному множеству атрибутов - A_1, A_2 ,...., A_k , порядок которых фиксирован. Таблица может содержить произвольное число строк; ее элементы - символы следующих видов:

а) свободные переменные (distinguished variables) - они соответствуют $a_1,a_2,\dots,a_n$ в (1) и (2). Будем обозначать их через букву a с нижным индексом - a_1 , a_2 ,....;

б) Связанные переменные (nondistinguished variables)- они соответствуют символам $b_1,b_2,\dots,b_m$ в (1) и (2). Будем обозначать их через букву b с нижным индексом - b_1 , b_2 ,....;

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

г) пробелы.

Над таблицей (или в качестве ее первой строки) задаются атрибуты, для которых составлена данная таблица - A_1, A_2 ,...., A_k . В следующей строке (здесь мы будем считать, что именно она является первой строкой таблицы) могут находиться только свободные переменные,константы или пробелы.

Эта строка называется резюме таблицы и представляет собой выражение, находящееся в (1) и (2) слева от косой черты "/". Способ расположения свободных переменных в резюме таблицы показывает к каким атрибутам следует их отнести. Например, a_1a_2 не означает, что a_1 является свободной переменной над атрибутом A_1, а a_2 - свободной переменной над атрибутом A_2. Запись a_1a_2 обретает 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.

Кроме строки резюме таблица содержит и строки, описывающие выражения справа от косой чертой в (1) и (2). Эти строки будем называть просто "строками" таблицы и будем их использовать для описания термов вида $R(c_1,c_2,\dots,c_s)$ в п. а) дефиниции 2. Каждому терму $R(c_1,\dots,c_s)$ отводим одну строку таблицы следующим способом:

- если отношение R задано над атрибутами $A_{i_1},A_{i_2},\dots, A_{i_s}$, то тогда в столбцы, соответствующие этим атрибутом, ставим c_1 для A_{i_1}, c_2 для $A_{i_2},\dots,$ c_s для A_{i_s}. Из п.а) дефиниции 2 видно, что таким образом в строке могут участвовать свободные переменные, связанные переменные и константы;
- в столбцы атрибутов, которые не участвуют в отношении R, ставим пробелы.

Каждой строке ставим маркер с правой стороны таблицы - если строка отведена терму $R(c_1,c_2,\dots,c_s)$, справа ставим маркер (R) и таким образом отмечаем "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.

Таким образом при помощи таблицы мы описали выражения слева от косой черты в (1) и (2) и термы вида $R(c_1, c_2, \ldots, c_s)$, находящиеся справа от косой черты. Так как справа в (1) могут фигурировать и выражения вида $c\,\theta\,d$ (п.б/ дефиниции 2), 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 получаем таблицу

(3')

ПИМЯ	ПНОМЕР	ПАДРЕС	ПГОРОД	
a_1				
a_1	b_1	b_2	b_3	(ПОСТАВЩИК)

$$b_3 = c_1$$

или

(3'')

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

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 3'' отношением-результатом является:

$R' = \{a_1 / a_1 \in$ 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' как "результатом" таблицы 3''.

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)

$$b_3 < c_1$$

Отношение-результат можно записать следующим образом

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

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

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

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 примерной базе данных:

q_2 : Найти имена всех поставщиков и всех клиентов, живущих в одном и том же городе.

Конъюнктивное представление запроса:

(4) $\{a_1a_2 / (\exists b_1)(\exists b_2)(\exists b_3)(\exists b_4)(\exists b_5)(\exists b_6)$ так, что
ПОСТАВЩИК $(a_1b_1b_2b_3)$ и КЛИЕНТ $(a_2b_4b_5b_6)$
и $(b_3 = b_6)\}$.

Соответствующая таблица имеет вид:

T_2:

ПИМЯ	ПНОМЕР	ПАДРЕС	ПГОРОД	КИМЯ	КНОМЕР	КАДРЕС	КГОРОД	
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 этой примерной базе данных 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 напр. АДРЕС и ГОРОД, как это сделано в примере 8.6 в /7/. Тогда для q_2 мы бы 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_3 :

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

Его конъюнктивная запись имеет вид:

(5) $\{a_1 a_2 / (\exists b_1)(\exists b_2)(\exists b_3)(\exists b_4)(\exists b_5)$ так, 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 (5) 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 дефиниции 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 т.наз. предположение о единственной роли (unique role assumption - см. /3/). В этом случае можем быть уверены, что данная выше дефиниция таблицы позволить нам сопоставлять каждому конъюнктивному запросу соответствующую ему таблицу.

3. ПОСТРОЕНИЕ ТАБЛИЦ ПО ДАННЫМ SPJ – ВЫРАЖЕНИЯМ

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

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. Достаточно показать способ сопоставления таблиц выражениям r , $\sigma_{A_i=c}(E_1)$, $\pi_X(E_1)$ и $E_1 \bowtie E_2$, где E_1 и E_2 – SPJ – выражения, для которых мы получили таблицу.

Предположим, что $E = r$. 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. Таблица для отношения ЧАСТЬ примерной базы данных дана на фиг. 2.

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:

а) если столбец A_i в резюме T_1 – пустой, то выражение E не имеет смысла и таблица T – неопределена.

б) 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 $T = \emptyset$.

в) 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 встечается a в T_1.

Получение таблицы для операции селекцию илюстрирано на фиг. 2.

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 символи" в столбцы, непринадлежащие X. Во всех остальных строках для этих столбцов свободные переменные заменяются разными связанными переменными.

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:

а) 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;

б) свободная переменная в одной из таблиц или в обеих, то в резюме T ставится та же переменная;

в) пустые символы в обеих таблицах T_1 и T_2, то в таблицу T ставим также пустой символ.

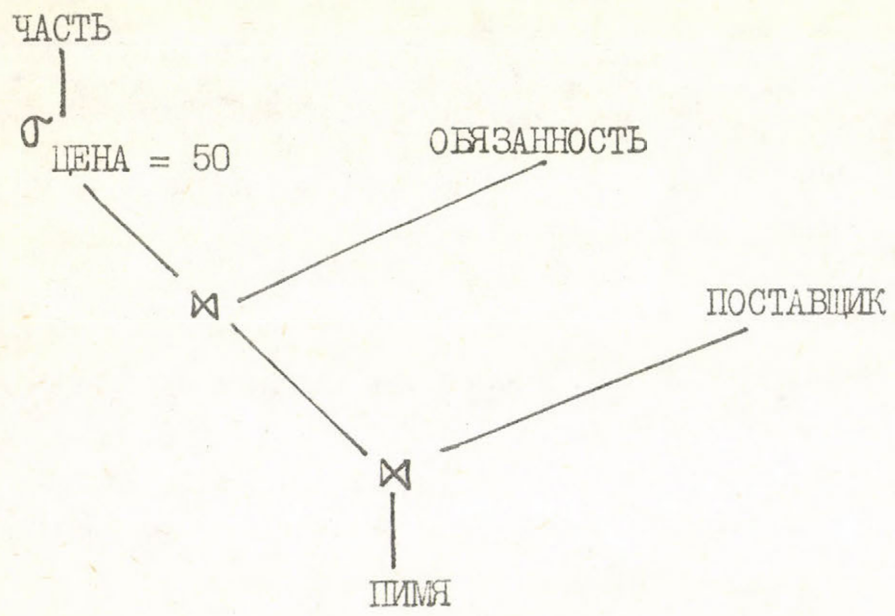
Пример 4. Проиллюстрируем процесс конструирования таблицы для SPJ - выражения на следующем примере:

q_4 : Найти имена поставщиков, поставляющие части ценой 50.

Ответ запроса можно представить при помощи SPJ - выражения:

$$\pi_{\text{ПИМЯ}} ((\text{ОБЯЗАННОСТЬ} \bowtie \sigma_{\text{ЦЕНА} = 50} (\text{ЧАСТЬ})) \bowtie \bowtie \text{ПОСТАВЩИК}). \quad (6)$$

Дерево разбора /2/ этого выражения дано на фиг. 1.



Фиг. 1. Дерево разбора для SPJ- выражения (6).

При помощи этого дерева представляется последовательность конструирования таблицы для SPJ - выражения (6). В таблицах на фиг. 2 атрибуты обозначены только через две буквы (например, ЧИМЯ обозначено через ЧИ - $\frac{Ч}{И}$), а все связанные переменные, встечающиеся только один раз, пропущены (т.е. замещены пустыми символами).

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 дефинициях:

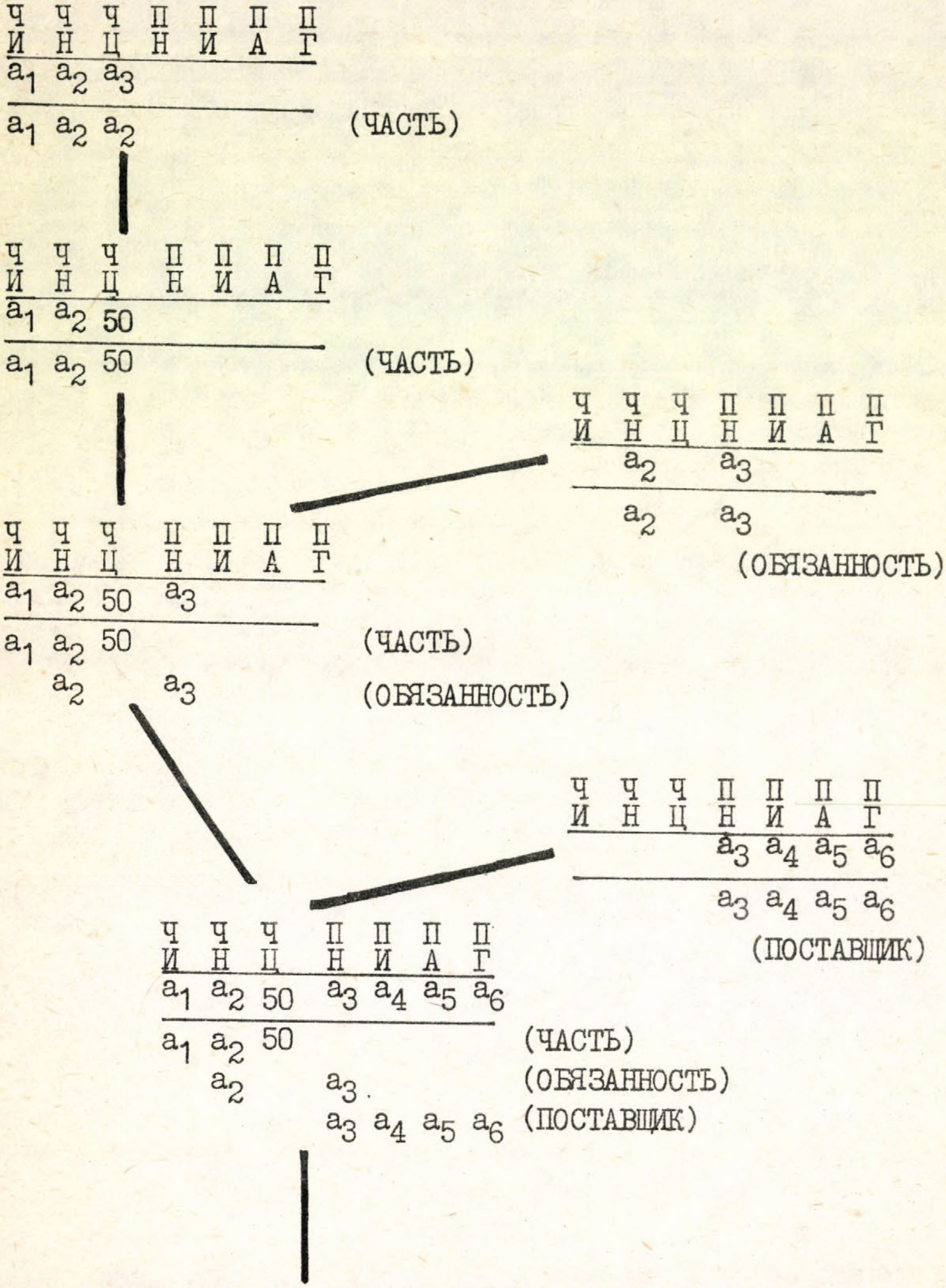
Дефиниция 4. Пусть $d = \{r_1, r_2, .., r_n\}$ представляет собой состояние базы данных, т.е. множество отношений над реляционными схемами $\{R_1, R_2, .., R_n\}$. Тогда отношение - результат, сопоставленное данной таблице T при помощи вышеописанной интерпретации, будем означать через $T(d)$ (естественно, значение этого отношения является различным для различных состояний d).

Дефиниция 5. Будем говорить, что $T_1 \subseteq T_2$, если для каждого d в силе $T_1(d) \subseteq T_2(d)$.

Дефиниция 6. 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 эквивалинтности двух данных таблиц (см. /1/ и /7/), лежит понятие "отображения" (mapping) между символами и строками таблиц.

Дефиниция 7. Пусть T_1 и T_2 - таблицы. Отображение h символов T_1 на символы T_2 называется "содержащим отображением" (containment mapping), если:



Фиг. 2. Конструирования таблицу для SPJ – выражения (6).

Ч И	Ч Н	Ч Ц	П Н	П И	П А	П Г	
				a_4			
b_1	b_2	50					(ЧАСТЬ)
	b_2		b_3				(ОБЯЗАННОСТЬ)
			b_3	a_4	b_4	b_5	(ПОСТАВЩИК)

Фиг. 2. (Продолжение). Конструирование таблицу для SPJ- выражения (6).

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 отображает 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. (см. /7/). $T_1 \geq T_2$ тогда и только тогда, когда существует "содержащее отображение" h из T_1 на T_2 .

Следствие. $T_1 \equiv T_2$ 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.

Дефиниция 8. Пусть T - таблица. Минимальной таблицей для таблицы T будем называть таблицу, содержащую минимальное число строк и эквивалентной таблице T. Процесс нахождения минимальной таблицы для данной таблицы T будем называть оптимизацией таблицы T.

В рассматриваемых до сих пор таблицах маркеры (tags) показывают из какого отношения берется данная строка. Определенная выше эквивалентность, где в п.б) дефиниции 7 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 - complete). Эта минимальная таблица, сильно эквивалентная данной таблице, должна содержить строки с такими же маркерами, как и выходная таблица. Такая минимальная таблица предполагает, что при обработке началного конъюнктивного запроса будут применяться J - операции 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 U над атрибутами

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

такого, что $r_i = \pi_{R_i}(U)$ для $1 \leq i \leq n$. Это предположение известно под именем "предположение существования универсума" (universal instance assumption- см. /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 откуда 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 определении понятия слабой эквивалентности.

Дефиниция 9. Пусть E_1 и E_2 - два запроса над данным состоянием d и пусть U - универсальное отношение для d . E_1 и E_2 являются слабо эквивалентными (записываем $E_1 \equiv_w E_2$), если $E_1(U) = E_2(U)$ для каждого возможного состояния U .

Аналогичную дефиницию вводим и для таблиц.

Дефиниция 10. Пусть T_1 и T_2 - таблицы соответственно для конъюнктивных запросов E_1 и E_2. T_1 и T_2 являются слоба эквивалентными (записываем $T_1 \equiv_w T_2$) тогда и только тогда, когда $E_1 \equiv_w E_2$.

В /7/ показано, что необходимое и достаточное условие слабой эквивалентности двух таблиц T_1 и T_2 - данное в теореме 1 условие. 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).

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 дла 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 критерие оптимальности проблема оптимизирования данного запроса сводиться к нахождению таблицы, слабо эквивалентной данной таблице.

Пример 5. Предположим, что для базы данных из примера 1 выполнено предположение о существовании универсума. Ищем ответ для следующего запроса:

q_5 : Найти имена всех поставщиков, поставляющие (или уже поставшие) части, количество которых - 500.

SPJ - выражение, реализирующее ответ:

(7) $\pi_{\text{ПИМЯ}}$ (ПОСТАВЩИК $\bowtie$ ОБЯЗАННОСТЬ $\bowtie$ ПОСТАВКА).

Соответствующая этому запросу таблица T_5:
(здесь пропущены столбцы атрибутов, не участвующих в описании q_5):

T_5 :

ПИМЯ	ПНОМЕР	ПАДРЕС	ПГОРОД	ЧНОМЕР	КНОМЕР	КОЛИЧЕСТ.
a_1						
a_1	b_1					
	b_1			b_2		
	b_1			b_2		500

Сразу видно, что T_6 является оптимальной слабо эквивалентной таблицей для таблицы T_5:

T_6:

ПИМЯ	ПНОМЕР	ПАДРЕС	ПГОРОД	ЧНОМЕР	КНОМЕР	КОЛИЧЕСТВО
a_1						
a_1	b_1					
	b_1					500

так как существуют отображение h_1 символов строк T_5 в символы строк T_6 и отображение h_2 символов строк T_6 в символы строк T_5. h_1 отображает первую строку T_5 в первую строку T_6, вторую строку T_5 в первую строку T_6 и последную строку T_5 во вторую строку T_6; h_2 отображает первую строку T_6 в первую строку T_5 и вторую строку T_6 в последную строку T_5.

Таблица T_6 представляет выражение

(8). $\pi_{\text{ПИМЯ}}$ (ПОСТАВЩИК $\bowtie$ ПОСТАВКА) .

Таким образом ясно, что выражения (7) и (8) являются слабо эквивалентными.

Нужно отметить, что здесь допущение о существовании универсума является существенным. (Таблицы T_5 и T_6 не являются сильно эквивалентными).

Пример 5 иллюстрирует и применение понятия таблицы в System/U (см. /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 /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.

Konjuktiv lekérdezési táblázatok és alkalmazásaik

G. Angelova

Összefoglaló

A szerző bevezeti a "konjuktiv lekérdezési táblázat" fogalmát és megadja az ilyen táblázatokra vonatkozó alapvető definiciókat és jellemzéseket. Leir néhány alkalmazási lehetőséget is pl. az adott SPJ-kifejezésnek megfelelő táblázat konstruálására vonatkozó algoritmust vagy az adott táblázattal ekvivalens minimális táblázat megkeresését.
A táblázatok ekvivalenciájának feltételeivel is foglalkozik.

Tableaux and their applications

G. Angelova

Summary

The paper presents the notion of conjunctive query tableau. The basic definitions and characteristics of tableaux are given. The basic possibilities for applications are described: the algorithm for constructing tableau corresponding to a given SPJ - expression, the process of tableaux optimization in order to find a minimal tableau equivalent to a given one. The conditions for tableaux equivalence are described and thus an algorithm for finding weak or string equivalent conjunctive queries is presented.