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 $R_1, R_2, \ldots, R_\kappa, \ldots$.

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

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:

$$r = \{\langle a_1 a_2 \ldots a_n \rangle \mid a_i \in dom\ (A_i),\ 1 \leq i \leq n\}$$

Элементы $\langle a_1 a_2 \ldots a_n \rangle$ 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 $\alpha = \langle a_1 a_2 \ldots a_n \rangle$ - 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 $\alpha [X] = \langle \alpha [A_{i_1}] \ \alpha [A_{i_2}] \ldots \alpha [A_{i_k}] \rangle = \langle a_{i_1} a_{i_2} \ldots a_{i_k} \rangle$.

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

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

Определение 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 σ, Π, J и именами реляционных схем.

Определение 2.

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) $c \, \theta \, d$, 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 θ 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 a 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_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 имеет смысл только в конкретной таблице, причем расположение переменных a_1 и a_2 в резюме показывает к каким атрибутам относят-

ся 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).

Пример 4.

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:

$$\{a_1a_2 \mid (\exists b_1)(\exists b_2)(\exists b_3)(\exists b_4)(\exists b_5) \text{ такие, что } R_1(b_1a_1b_2) \wedge R_2(b_2a_2b_3) \wedge R_3(b_4b_5c_1) \wedge b_3 < c_1\}.$$

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) $\{a_1a_2 \mid (\exists b_1)(\exists b_2)(\exists b_3)(\exists b_4)(\exists b_5)(\exists b_6)$ такие, что ПОСТАВЩИК $(a_1b_1b_2b_3) \wedge$ КЛИЕНТ $(a_2b_4b_5b_6) \wedge (b_3 = b_6)\}$.

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) $\{a_1a_2 \mid (\exists b_1)(\exists b_2)(\exists b_3)(\exists b_4)(\exists b_5)(\exists b_6)$ такие, что

ПОСТАВЩИК $(b_1b_2a_1b_3) \wedge$ КЛИЕНТ $(b_4b_5a_2b_6) \wedge$

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

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 дслжны 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:

а) $E = \Pi_X (E_1)$, 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 $A_i \in \{A_1, A_2, \ldots, A_\kappa\}$ 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:

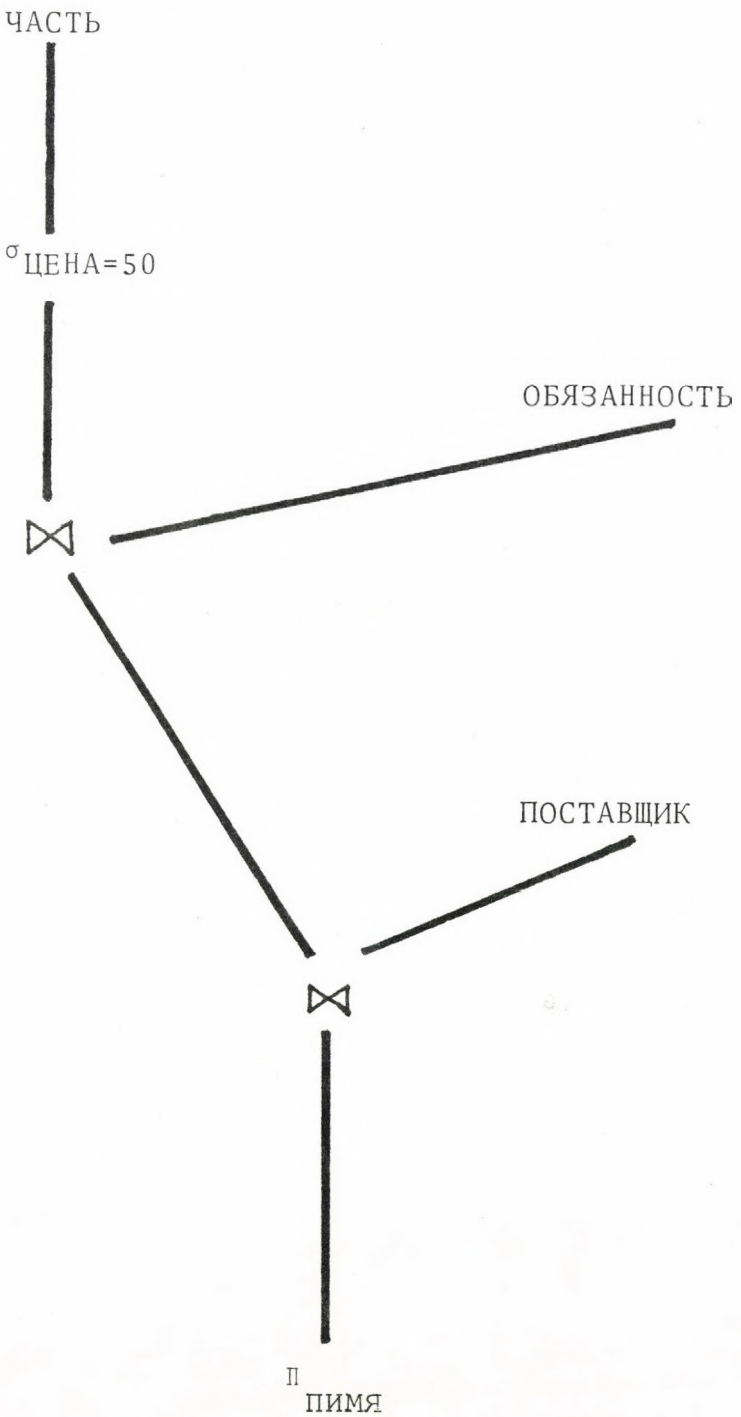
(6) $\Pi_{\text{ПИМЯ}}((\text{ОБЯЗАННОСТЬ} \bowtie \sigma_{\text{ЦЕНА}=50}(\text{ЧАСТЬ})) \bowtie \text{ПОСТАВЩИК})$

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.

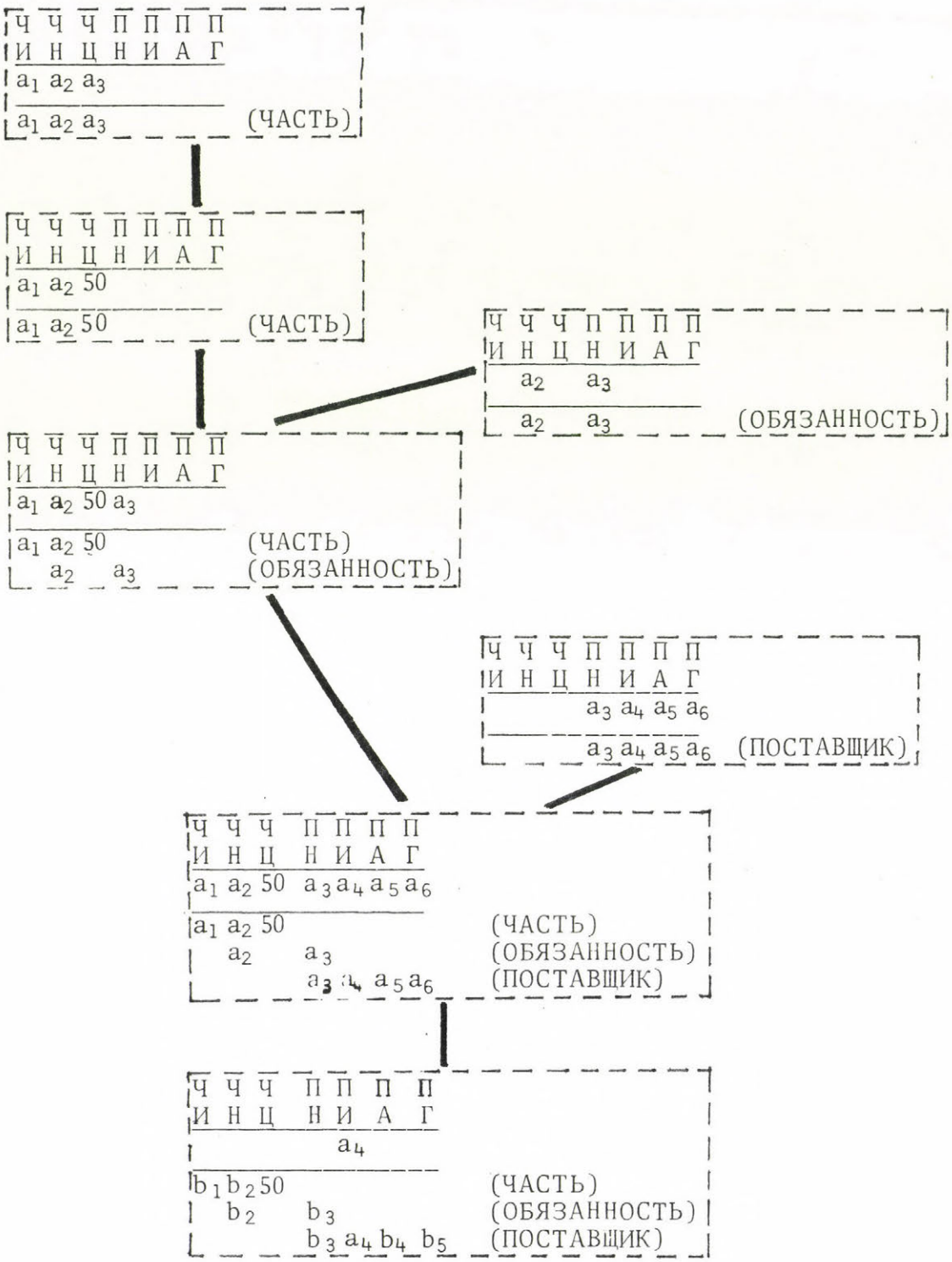
В таблицах на фиг.2 атрибуты обозначены только через две буквы (например, ЧИМЯ обозначено через ЧИ - $\frac{\text{Ч}}{\text{И}}$), а все связанные переменные, встречающиеся только один раз, пропущены (т.е. замещены пустыми символами).

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 T - таблица. Минимальной таблицей для таблицы T будем называть таблицу, содержащую минимальное число строк и эквивалентной таблице T. 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 T будем называть оптимизацией таблицы T.

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 следующих выражениях:

(7) $\Pi_{AB}(AB \bowtie BC)$ и

(8) 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).

Пример 7.

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, реализирующее 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 предполо-

жение - 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 építi 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.