Представление замыканий на множестве функциональных и соединенных зависимостей в реляционных схемах*

Я. ДЕМЕТРОВИЧ и НГУЕН СУАН ЗУЙ (Будапешт)

К пятидесятилетнему юбилею профессора Золтана Дароци

1. Введение

Изучение зависимостей данных играет важную роль в уменьшении избыточностей и повышении точностей данных при проектировании баз данных.

Понятие функциональных зависимостей (*ФЗ*) впервые было описано в [6] и затем использовано в теории нормализации отношений [7]. Система вывода для *ФЗ* впервые была аксиоматизирована в работе [8], ее полнота и корректность были доказаны в [2].

Понятие многозначных зависимостей (*МЗЗ*) впервые было введено в [11]. Первая полная аксиоматизация *ФЗ* и *МЗЗ* была получена в [5] в 1977 г.

Понятие соединенных зависимостей (*СЗ*) впервые было дано в работе [16] и затем активно развито в [1]. В работе [17] описана система аксиом для класса зависимостей, повидимо, шире класса *СЗ*. В этой же работе была высказана гипотеза о том, что не существует конечная система аксиом для *СЗ*.

В теории реляционных баз данных проблему членства можно формулировать следующим: Дана совокупность зависимостей C (*ФЗ* и/или *МЗЗ* и/или *СЗ*) и некоторая зависимость x. Требуется построить алгоритм распознавания выводимости зависимости x из совокупности C. Различные алгоритмы были построены для различных классов зависимостей [1, 3, 4, 7, 8, 9, 14]. Вообще, сложности этих алгоритмов зависят от природы данных зависимостей.

В случае, когда C является совокупностью одних *ФЗ*, а x — данная *ФЗ*, был получен алгоритм решения проблемы членства с временной сложностью пропорциональной длине входа [4]. Существуют алгоритмы с полиномиальной сложностью по длине входа для случая, когда C является совокупностью *ФЗ* и/или *МЗЗ* и/или *СЗ* и x — данная *ФЗ* или *МЗЗ* [12].

В работе [15] доказана, что проблема членства является *NP*—hard если C состоит из одной *СЗ* и функциональных зависимостей, а x — данная *СЗ*.

При поисках решений к проблеме членства используются различные подходы и инструменты, каждый из которых выгоден определенным классам за-

* Research supported by Hungarian Foundation for Scientific Research Grant 1066.

висимостей: дерева происхождения [13], пропозициональные формулы [8], табло и их следы [1, 14], трансляции реляционных схем [9, 10].

Известно, что если $x = Y \to Z$ является *ФЗ*, то совокупность зависимостей C выводит x тогда и только тогда, когда Z является подмножеством замыкания Y^+ [2]. Это означает, что проблема членства для *ФЗ* равносильна проблеме вычисления замыкания, поэтому задача о представлении замыканий имеет определенный интерес в решении поставленной проблемы.

В общем случае, если C является совокупностью *ФЗ* и $x - ФЗ$, то сложность алгоритма решения проблемы членства с помощью табло имеет порядок $\mathcal{O}(l^3)$ по длине входа l [12].

Понятие трансляций реляционных схем впервые было описано в работе [10] и использовано для поиск представления замыканий относительно совокупности *ФЗ*.

Данная работа посвящена рассмотрению нового варианта — расширения понятия трансляций реляционных схем на совокупность *ФЗ* и *СЗ*.

В пункте 2 излагаются нужные сведения и определения.

Главная теорема о представлении замыкания множества атрибутов реляционных схем со совокупностью *ФЗ* и *СЗ* формулируется и доказывается в пункте *3*.

В пункте 4 описывается природа замыкания пустых множеств в реляционных схемах.

Наконец, в пункте 5 предлагается схема вычисления замыкания множества атрибутов относительно совокупности *ФЗ* и *СЗ*. В этом же пункте исследуются сложности вычисления по времени нескольких этапов данной схемы.

2. Определения

Определение 2.1. Пусть R — конечное множество символов, называемых атрибутами $\{A_1, A_2, \ldots, A_n\}$. Каждому атрибуту A_i сопоставим множество $\mathrm{DOM}(A_i)$, $1 \leq i \leq n$, называемое доменом атрибута A_i. Множества $\mathrm{DOM}(A_i)$, $1 \leq i \leq n$ могут быть конечными, счетными или не различными между собою. Требуется только, чтобы они были непустыми.

Пусть $D = \bigcup_{i=1}^{n} \mathrm{DOM}(A_i)$. Отношением $r(R)$ на множестве атрибутов R называется конечное множество отображений $\{t_1, t_2, \ldots, t_p\}$ из R в D таких, что для каждого

$$t \in r(R), \quad t(A_i) \in \mathrm{DOM}(A_i), \quad 1 \leq i \leq n.$$

Элементы из $r(R)$ называются кортежами.

Мы часто пишем r вместо $r(R)$, когда R уже известно. Мы и полагаем существование специального значения e такого, что $t(\emptyset) = e$ для любого кортежа t в любом отношении. Из этого следует $t_1(\emptyset) = t_2(\emptyset)$ для всяких t_1 и t_2 из любого отношения.

По традиции, принятой в теории реляционных моделей данных мы будем писать XY вместо $X \cup Y$ для множеств атрибутов X и Y. Кроме того, вместо записи одноатрибутного множества $\{A\}$ мы будем писать просто A.

Определение 2.2. Пусть $t \in r(R)$ и $X \subseteq R$. X-значение кортежа t, записываемое так $t(X)$ и понимается как ограничение отображения t на X.

Определение 2.3. Проекцией отношения $r(R)$ на подмножестве атрибутов $X \subseteq R$ называется отношение

$$r[X] = \{t(X) | t \in r(R)\}.$$

Определение 2.4. Пусть $r(R)$ и $s(S)$ отношения. Пусть $T = RS$. Соединением (естественным) отношений $r(R)$ и $s(S)$ называется отношение $r * s$ на множестве атрибутов T, составляющее из всех таких кортежей t, что существуют кортежи u и v для которых $u = t(R)$ и $v = t(S)$.

Заметим, что в этом случае мы имеем $u(R \cap S) = v(R \cap S)$, так как $R \cap S$ является подмножеством и R и S. Таким образом, каждый кортеж из $r * s$ является комбинацией кортежа из r и кортежа из s с равными $(R \cap S)$-значениями.

Определение 2.5. Пусть $r(R)$ отношение. Пусть x является последовательностью подмножеств $X_1, X_2, \ldots, X_p$ множества R, где $R = X_1 X_2 \ldots X_p$. Проекционно-соединенным отображением по x называется отображение $m_x(r) = r[X_1] * r[X_2] * \ldots * r[X_p]$.

Определение 2.6. Пусть $r(R)$ отношение и пусть $X, Y \subseteq R$. Говорят, что отношение r удовлетворяет функциональной зависимости (*ФЗ*) $X \to Y$, если для всяких двух кортежей u и v из r таких, что $u(X) = v(X)$ всегда следует, что $u(Y) = v(Y)$.

Определение 2.7. Пусть $r(R)$ отношение и пусть $X_1, X_2, \ldots, X_p$ подмножества R, где $R = X_1 X_2 \ldots X_p$. Говорят, что отношение r удовлетворяет соединенной зависимости (*СЗ*) $x = {}^*[X_1, X_2, \ldots, X_p]$ если $r = m_x(r)$.

Во всех местах этой работы C будет обозначать множество функциональных и соединенных зависимостей, заданных на множестве атрибутов R. Элементы из C будем называть ограничениями.

Определение 2.8. Ограничение x, заданное на множестве атрибутов R является тривиальным если x удовлетворяется в любом отношении $r(R)$.

Из определений 2.6 и 2.7 известно, что для любого отношения тривиальными будут такие и только такие *ФЗ* и *СЗ* следующих форм:

— $X \to Y$, где $X \supseteq Y$, и

— ${}^*[X_1, X_2, \ldots, X_p]$, где $R = X_i$ для некоторого i, $1 \leq i \leq n$.

Определение 2.9. Реляционной схемой (*PC*) называется парой $\langle R, C \rangle$, где R — множество атрибутов и C — множество *ФЗ* и *СЗ* на R.

Пусть дана *PC* $a = \langle R, C \rangle$, обозначим $\mathrm{SAT}(a)$ совокупность всех отношений определенных на множестве атрибутов R и удовлетворяющих всем ограничениям из C.

Определение 2.10. Пусть $a = \langle R, C \rangle$ — заданная *PC* и x заданное ограничение. Говорят, что x выводимо из C если

$$\mathrm{SAT}(R, C) \subseteq \mathrm{SAT}(R, x).$$

Если x выводимо из C, мы пишем $C \models x$.

Проблема членства. Даны $PC\langle R, C\rangle$ и ограничение x. Требуется выяснить $C\models x$?

Определение 2.11. Пусть $R=\{A_1, A_2, \ldots, A_n\}$ множество атрибутов. Рассмотрим два следующих множества:

— $V_d=\{a_1, a_2, \ldots, a_n\}$ множество характерных или различающих переменных,

— $V_n=\{b_1, b_2, \ldots\}$ множество нехарактерных или не различающих переменных, где $V_d\cap V_n=\emptyset$.

Табло $T(R)$ на множестве атрибутов R может рассматриваться как отношение, где для каждого $A_i\in R$, $\mathrm{DOM}\,(A_i)=\{a_i\}\cup V_n$.

Из определения видно, что если характерная переменная содержится в столбце A_i, $1\leqq i\leqq n$, то она должна быть только a_i.

Пусть даны PC $a=\langle R, C\rangle$ и табло T на R. Для каждого ограничения $x\in C$, рассмотрим табло $T(x)$, получаемое из T применением правила соответствующего типа ограничения x: Φ-правило: Это правило применяется для каждого табло T и каждого ограничения $x=Y\to Z$, являющегося $\Phi З$ из C.

Если в T существуют две строки u и v такие, что $u(Y)=v(Y)$ то будем уравнивать значения $u(A)$ и $v(A)$ для каждого атрибута $A\in Z$ следующим образом:

— Если только одна из двух переменных $u(A)$ и $v(A)$ является характерной, то другой переменной будем присваивать соответствующим символом характерной переменной.

— Если $u(A)=b_i$, $v(A)=b_j$, то будем сравнивать их индексы i и j:

если $i<j$, то присвоить $v(A):=b_i$

если $i>j$ то присвоить $u(A):=b_j$

C-правило: Это правило применяется для каждого табло T и каждого ограничения $x={}^*[X_1, X_2, \ldots, X_p]$ являющегося $СЗ$ из C по формуле

$$T(x) = m_x(T),$$

где T рассматривается как отношение на R.

Определение 2.12. Пусть дана PC $a=\langle R, C\rangle$ и дано табло T на R. Применим правила типа Φ и C, соответствующего с $\Phi З$ и $СЗ$ из C до тех пор, пока T изменяется мы получим так называемую порождающую последовательность табло $T_0, T_1, T_2, \ldots,$ где $T_0=T$.

Известны следующие результаты [1, 16, 17]:

— Последовательность $T_0, T_1, T_2, \ldots$ конечна, т. е. существует шаг $k\geqq 0$, такой что $T_k=T_{k+1}=\ldots$. Такой T_k принято назвать следом табло T на PC $a=\langle R, C\rangle$ и обозначать $T^*=\mathrm{chase}_a\,(T)$,

— след табло T не зависит от порядка применения Φ- и C-правил.

Другими словами, для каждого табло T, его след относительно заданной PC всегда существует и единствен.

Определение 2.13. Пусть $r(R)$ отношение и пусть $X, Y\subseteq R$. Говорят, что отношение r удовлетворяет многозначной зависимости ($МЗЗ$) $X\twoheadrightarrow Y$, если r удовлетворяет $СЗ$ ${}^*[XY, XZ]$, где $Z=R-XY$.

В данной работе мы всегда рассматривать $МЗЗ$ как частный случай $СЗ$.

Определение 2.14. Пусть дана *PC* $a=\langle R, C\rangle$ и пусть дано $X\subseteq R$. Замыканием $(X)_a^+$ множества атрибутов X на *PC* a является наибольшее множество атрибутов $Y\subseteq R$ такое, что $C\models X\to Y$.

Операция замыкания обладает следующими свойствами:

(C1) $X\subseteq(X)_a^+$.

(C2) Если $X\subseteq Y$, то $(X)_a^+\subseteq(Y)_a^+$ (монотонность).

(C3) $((X)_a^+)_a^+=(X)_a^+$ (идемпотентность).

Определение 2.15. Пусть даны *PC* $a=\langle R, C\rangle$ и $X\subseteq R$. Табло T_X для множества атрибутов X называется табло со следующими двумя строками:

— строкой $u_d=\langle a_1, a_2, \ldots, a_n\rangle$, и

— строкой u_X:

$$u_X(A_i)=\begin{cases} a_i, & \text{если} \quad A_i\in X \\ b_i, & \text{иначе}\end{cases}$$

$1\leq i\leq n$.

Верны следующие теоремы [1, 14]:

Теорема 2.1. $C\models X\to Y\Leftrightarrow Y\subseteq(X)_a^+$.

Теорема 2.2. *$(X)_a^+$ состоит из тех атрибутов $A\in R$ для которых A-столбец в* $\mathrm{chase}_a(T_X)$ *содержит только характерные переменные.*

3. Трансляция реляционных схем

В данном пункте описывается оператор трансляции *PC* и доказываются некоторые свойства, связанные с построением замыканий множества атрибутов относительно совокупности *ФЗ* и *СЗ*.

Пусть даны *PC* $a=\langle R, C\rangle$ и $M\subseteq R$. Для каждого объекта E (E может быть множеством атрибутов, *ФЗ*, *СЗ* или самой *PC*), рассмотрим новый объект $E\setminus M$, получаемый из E вычеркиванием всех вхождений символов, указанных в M, а именно:

— для каждой *ФЗ* $x=X\to Y\in C$: $x\setminus M=X-M\to Y-M$

— для каждого *СЗ* $x={}^*[X_1, X_2, \ldots, X_p]\in C$:

$$x\setminus M = {}^*[X_1-M, X_2-M, \ldots, X_p-M]$$

— $C\setminus M=\{x\setminus M|x\in C\}$

— $a\setminus M=\langle R\setminus M, C\setminus M\rangle$

— для каждого множества атрибутов X, мы конечно, имеем:

$$X\setminus M = X-M,$$

где обозначение $X-Y$ выражает обычную операцию вычитания множеств X и Y.

Определение 3.1. *PC* $a'=\langle R', C'\rangle$ получается трансляцией *PC* $a=\langle R, C\rangle$ на M, если $a'=a\setminus M$, $M\subseteq R$.

Теорема 3.1. *Дана PC $a=\langle R, C\rangle$. Для любых двух не пересекающихся под-*

множеств M и X множества атрибутов R справедливо:

$$(3.1) \qquad (MX)_a^+ = M(X)^+_{a \setminus M}.$$

Доказательство. Ради удобства и простоты в изложении доказательства данной теоремы, сначала введем некоторые сведения и обозначения.

Столбец A в табло T называется однородным если он состоит только из одинаковых переменных.

Так как Φ-правило может применимо ко какому-то столбцу A табло T только в том случае, когда A содержит неодинаковые переменные, а C-правило, если оно применимо, лишь копирует переменные в столбце с одного места в другое новое место того же столбца, отсюда следует, что однородность столбцов табло является инвариантом при трансформации табло, т. е. все однородные столбцы остаются однородными при применении Φ- и C-правил к табло.

Так как всякое табло рассматривается как отношение (см. определение 2.11), то в нем не должны быть две одинаковые строки. Две строки u и v табло T могут отличаться только по переменным, соответствующим неоднородным столбцам. Отсюда видно, что количество строк любого табло не изменяется при удалении из него нескольких или всех однородных столбцов.

Будем называть однородные столбцы, содержащие характерные переменные характерными.

Пусть T — табло на множестве атрибутов R и M — множество атрибутов, соответствующих однородным столбцам табло T, будем обозначать $T{:}M$ табло на $R-M$, получаемое из T удалением всех однородных столбцов с именами, указанными в M. Если $R \cap M = \emptyset$, то через $T.M$ будем обозначать табло на RM, получаемое добавлением к T характерных столбцов с именами, указанными в M.

Приходим к доказательству теоремы 3.1.

Доказательство будет проводиться по процессу вычисления следов соответствующих табло для MX в $PC\,a$ и для X в $PC\,a \setminus M$. Именно мы будем доказывать, что если любая порождающая последовательность для $\mathrm{chase}_a(T_{MX})$ является $T_0 = T_{MX}, T_1, T_2, \ldots, T_k = T^*_{MX} = \mathrm{chase}_a(T_{MX})$, то существует соответствующая порождающая последовательность для $\mathrm{chase}_{a \setminus M}(T_X)$

$$T'_0 = T_X, T'_1, T'_2, \ldots, T'_k = T^*_X = \mathrm{chase}_{a'}(T_X),$$

где $a' = a \setminus M$, такая что каждое T'_i, $1 \leqq i \leqq k$ получается из T_i удалением всех однородных столбцов с именами, указанными в M, т. е. $T'_i = T_i{:}\ M$, $1 \leqq i \leqq k$. Отсюда, из существования и единственности следов следует правильность равенства (3.1). Будем применять индукцию по числу k.

При $k=0$ имеем $T_0 = T_{MX}$ и $T'_0 = T_X$. Относительно $PC\,a$ и $a \setminus M$, по определению 2.15 T_{MX} и T_X отличаются только по $|M|$ столбцов с именами, указанными в M. Так как эти столбцы являются характерными, то мы имеем $T'_0 = T_0{:}M$.

Предположим $T'_k = T_k{:}M$, будем доказывать, что $T'_{k+1} = T_{k+1}{:}M$. По правилу построения порождающей последовательности мы имеем $T_{k+1} = T_k(x)$ для некоторого $x \in C$. Рассмотрим два случая:

Случай 1: x является *ФЗ* из C. Пусть $x=Y\to Z$. Допустим в T_k существуют две строки u и v такие, что $u(Y)=v(Y)$, а $u(Z)\neq v(Z)$. По индуктивному предположению мы должны иметь $u'=u(R-M)\in T_k'$, $v'=v(R-M)\in T_k'$. Из $u(Y)=v(Y)$ и $Y-M\subseteq Y$ следует $u(Y-M)=v(Y-M)$. Рассмотрим множество $E=\{A\in Z|u(A)\neq v(A)\}$. Так как T_k и T_k' отличаются только характерными столбцами, то мы имеем $E\cap M=\emptyset$. Из $E\subseteq Z$ и $E\cap M=\emptyset$ следует $E\subseteq Z-M$. Так как $Y\subseteq R$ то $Y-M\subseteq R-M$, поэтому $u'(Y-M)=u(R-M)(Y-M)=$ $=v(R-M)(Y-M)=v'(Y-M)$. С другой стороны $u'(E)$ и $v'(E)$ отличаются по всем компонетам точно как и $u(E)$и $v(E)$. Это означает, что *ФЗ* $x'=$ $=x\setminus M=Y-M\to Z-M$ применима k T_k' точно так же как и *ФЗ* $x=Y\to Z$ применима k T_k. Следовательно $T_{k+1}'\supseteq T_{k+1}$: M. Докажем и обратное заключение. Пусть при переходе от T_k' k T_{k+1}' была применима *ФЗ* $x'=Y-M\to Z-M$ из $C'=C\setminus M$, где $x=Y\to Z$ является *ФЗ* из C. Это означает, что существуют покрайней мере две строки u' и v' из T_k', для которых $u'(Y-M)=v'(Y-M)$ и $u'(Z-M)\neq v'(Z-M)$. Пусть $E=\{A\in Z-M|u'(A)\neq v'(A)\}$. По индуктивному предположению $T_k'=T_k$: M. Из этого следует существования таких строк u и v из T_k, что $u(R-M)=u'$ и $v(R-M)=v'$. Заметим, что $u(M)=v(M)$, так как все столбцы с именами в M суть характерны. Из $Y\subseteq R$ следует $Y-M\subseteq$ $\subseteq R-M$, и поэтому $u(Y-M)=u(R-M)(Y-M)=u'(Y-M)=v'(Y-M)=$ $=v(R-M)(Y-M)=v(Y-M)$. Из $u(M)=v(M)$ и $u(Y-M)=v(Y-M)$ следует $u(Y)=v(Y)$. Так как $E\subseteq Z-M$, то $E\subseteq Z$. Отсюда следует, что *ФЗ* $Y\to Z$ из C применима к T_k точно как и *ФЗ* $Y-M\to Z-M$ из $C\setminus M$ применима к T_k', а это означает $T_{k+1}'\subseteq T_{k+1}$: M. Таким образом, для любой *ФЗ* $x\in C$ мы имеем:

$$T_k'(x\setminus M) = T_{k+1}' = T_{k+1}\colon M = T_k(x)\colon M.$$

Первый случай доказан.

Случай 2: x является *СЗ* из C. Обозначим $x'=x\setminus M$.

Пусть

$$x = {}^*[X_1, X_2, \ldots, X_p] \quad \text{и} \quad T_{k+1} = T_k(x) = m_x(T_k),$$

$$T_{k+1}' = T_k'(x') = m_{x'}(T_k').$$

Мы должны доказать равенство:

$$T_{k+1}' = T_{k+1}\colon M.$$

Пусть $u\in T_{k+1}$: M, это означает $\exists v\in T_{k+1}$, что $v(R-M)=u$. Т. к. $v\in T_{k+1}=$ $=m_x(T_k)$, то $\exists v_i\in T_k[X_i]$, что $v_i=v(X_i)$, $1\leqq i\leqq p$. Докажем существование $u_i\in T_k'[X_i-M]$, что $u(X_i-M)=u_i$, $1\leqq i\leqq p$, отсюда следует $u\in m_{x'}(T_k')=T_{k+1}'$ и следовательно T_{k+1}: $M\subseteq T_{k+1}'$.

Так как $X_i-M\subseteq X_i$, то из $v_i=v(X_i)$ следует

$$v_i(X_i-M) = v(X_i)(X_i-M) = v(X_i-M), \quad 1 \leqq i \leqq p.$$

Обозначив $u_i=v_i(X_i-M)$, $1\leqq i\leqq p$, имеем $u_i=v_i(X_i-M)=v(X_i-M)$. Так как $u=v(R-M)$, $X_i-M\subseteq R-M$, то $u(X_i-M)=v(R-M)(X_i-M)=v(X_i-M)=u_i$. Что и требовалось доказать.

И так мы доказали T_{k+1}: $M\subseteq T_{k+1}'$. Теперь докажем обратное заключение.

Пусть $u\in T_{k+1}'$, докажем, что $u\in T_{k+1}$: M. Другими словами, нам нужно доказать существование $v\in T_{k+1}$ такое, что $v(R-M)=u$.

Построим строку v на R такую:

$$v(A_j) = \begin{cases} a_j, & \text{если } A_j \in M \\ u(A_j), & \text{иначе} \end{cases}$$

$1 \leqq j \leqq n$. Очевидно $v(R-M)=u$.

Если мы укажем такие строки $v_i \in T_k[X_i]$, для которых

$$v_i = v(X_i), \quad 1 \leqq i \leqq p,$$

то это означает, что

$$v \in m_x(T_k) = T_{k+1}.$$

Так как $u \in T'_{k+1} = m_{x'}(T'_k)$, для $x' = x \setminus M = {}^*[X_1-M, X_2-M, \ldots, X_p-M]$, то существуют такие $u_i \in T'_k[X_i-M]$, что $u_i = u(X_i-M)$, $1 \leqq i \leqq p$. Из $u_i \in T'_k[X_i-M]$ следует существование таких $u'_i \in T'_k$, что $u'_i(X_i-M)=u_i$, $1 \leqq i \leqq p$. По индуктивному предположению $T'_k = T_k : M$, из $u'_i \in T'_k$ следует существование таких $v'_i \in T_k$, что $v'_i(R-M)=u'_i$, $1 \leqq i \leqq p$.

Обозначим $v_i = v'_i(X_i)$. Имеем $X_i = (X_i-M) \cup (X_i \cap M)$, $X_i \cap M \subseteqq M$, $X_i \cap M \subseteqq X_i$, $X_i - M \subseteqq R-M$, $1 \leqq i \leqq p$. Так как все столбцы с именами в M являются характерными, то:

$$v_i(X_i \cap M) = v'_i(X_i \cap M) = v(X_i \cap M).$$

Тогда имеем

$$v_i(X_i-M) = v'_i(X_i)(X_i-M) = v'_i(X_i-M) = v'_i(R-M)(X_i-M) =$$
$$= u'_i(X_i-M) = u_i = u(X_i-M) = v(X_i-M),$$

$1 \leqq i \leqq p$. Что и требовалось доказать.

И так мы доказали T_{k+1}: $M \supseteqq T'_{k+1}$. Следовательно T_{k+1}: $M = T'_{k+1}$. Второй случай доказан, тем самым доказана теорема 3.1. □

Следствие 3.1. *Дана PC $a = \langle R, C \rangle$. Если $X, Y \subseteqq R$ и $X \subseteqq Y \subseteqq (X)^+_a$, то*

$$(X)^+_a = Y(\emptyset)^+_{a \setminus Y}.$$

Доказательство. По свойствам (C2) и (C3) пункта 2 для замыкания, из $X \subseteqq Y \subseteqq (X)^+_a$ следует $(X)^+_a \subseteqq (Y)^+_a \subseteqq ((X)^+_a)^+_a = (X)^+_a$. Следовательно $(X)^+_a = (Y)^+_a$. По теореме 3.1 имеем $(Y)^+_a = (Y\emptyset)^+_a = Y(\emptyset)^+_{a \setminus Y}$. Следовательно $(X)^+_a = Y(\emptyset)^+_{a \setminus Y}$. Что и требовалось доказать. □

Следствие 3.2. *Дана PC $a = \langle R, C \rangle$. Для всякого $X \subseteqq R$, имеем*

$$(X)^+_a = X(\emptyset)^+_{a \setminus X}.$$

Доказательство. Очевидно. □

Из следствий 3.1 и 3.2 мы видим, что вместо нахождения замыкания множества атрибутов X в PC a мы можем найти, сначала, замыкание пустого множества в PC $a \setminus X$, затем прибавить X к найденному замыканию чтобы получить окончательный результат. Чтобы найти $a \setminus X$, мы просто удалим из PC a все вхождения символов, указанных в X. Заметим что PC $a \setminus X$ прощее исходной по следующим характеристикам;

— $a\diagdown X$ можеть иметь меньшее количество символов по сравнению с исходной,

— $a\diagdown X$ может иметь меньшее количество зависимостей или их компонентов по сравнению с исходной. Действительно, если в $a\diagdown X$ возникаются *ФЗ* или *СЗ* следующих видов:

(1) $Y \to Z,\ Y \supseteq Z$

(2) $*[Y_1, Y_2, \ldots, Y_p]$, где некоторые Y_i пусты, $1 \leqq i \leqq p$, то мы можем удалить *ФЗ* вида (1) из $C\diagdown X$ и удалить все пустые компоненты из каждого *СЗ* вида (2), так как они не влияют на результата вычисления замыкания.

А если в $a\diagdown X$ возникаюся *ФЗ* вида:

(3) $\emptyset \to Y,\ Y \neq \emptyset$,

то мы просто внесем Y в результат и будем искать замыкания пустого множества в новой $PC\ a\diagdown X\diagdown Y = a\diagdown XY$. В этом случае $ФЗ\ \emptyset \to Y$ уже переходится к форме $\emptyset \to \emptyset$ и поэтому она должна быть удалена из $a\diagdown XY$.

Но это не все, в процессе вычисления замыкания пустого множества $PC\ a$, если мы находим некоторые первые элементы, например M, то внесем M в результат и переходим на вычисление замыкания пустого множества в новой *PC*, а именно в $a\diagdown M$.

Кроме этого, нужно заметить, что в упрощенной *PC* могут встречаться одинаковые *ФЗ* и *СЗ*. В этом случае, мы, конечно можем удалить повторяющие из них.

Пример 3.1. Дана $PC\ a = \langle R, C\rangle$, где

$R = 123456$

$C = \{13 \to 2,\ 2 \to 134,\ 213 \to 4,\ 145 \to 236,\ 36 \to 15,\ *[12345, 346],\ *[1245, 1346, 13]\}$.

Требуется вычислить $(X)_a^+$, где $X = 13$?

Мы имеем:

$R\diagdown X = 2456$

$C\diagdown X = \{\emptyset \to 2,\ 2 \to 4,\ 2 \to 4,\ 45 \to 26,\ 6 \to 5,\ *[245, 46],\ *[245, 46, \emptyset]\}$

$\Rightarrow \{\emptyset \to 2,\ 2 \to 4,\ 45 \to 26,\ 6 \to 5,\ *[245, 46],\ *[245, 46]\}$

$\Rightarrow \{\emptyset \to 2,\ 2 \to 4,\ 45 \to 26,\ 6 \to 5,\ *[245, 46]\}$.

После обработки $ФЗ\ \emptyset \to 2$ мы имеем:

$C\diagdown X \Rightarrow \{\emptyset \to 4,\ 45 \to 6,\ 6 \to 5,\ *[45, 46]\}$.

После обработки $\emptyset \to 4$ мы получим:

$C\diagdown X \Rightarrow \{5 \to 6,\ 6 \to 5,\ *[5, 6]\}$.

Теперь нам нужно вычислить $(\emptyset)_{a'}^+$, где $a' = \langle R', C'\rangle$;

$R' = 56$; $C' = \{5 \to 6,\ 6 \to 5,\ *[5, 6]\}$.

Легко проверить, что $(\emptyset)_{a'}^+ = 56$.

Окончательно мы получим $(13)_a^+ = 123456$.

4. Замыкание пустых множеств

Как видно из следствия 3.2 задача определения замыкания множества атрибутов X в $PC\,a$ сводится к вычислению замыкания пустого множества в $PC\,a \setminus X$. В этом пункте исследуется природа этого замыкания.

Пусть дана $PC\,a=\langle R, C\rangle$, где $R=\{A_1, A_2, \ldots, A_n\}$ — множество атрибутов и C — совокупность $\Phi З$ и $СЗ$ на R. Напомним, что по определению 2.15 табло $T_\emptyset$ для пустого множества $\emptyset$ содержит только две строки: $u_a=\langle a_1, a_2, \ldots, a_n\rangle$ и $u_n=\langle b_1, b_2, \ldots, b_n\rangle$.

Главным инструментом используемым при поисках результатов данного пункта являются табло и их следы. Чтобы найти $(\emptyset)_a^+$ мы сначала вычислим $\text{chase}_a(T_\emptyset)$ путем применения Φ- и C-правил а затем вносим в $(\emptyset)_a^+$ те атрибуты $A\in R$, для которых A-столбец в $\text{chase}_a(T_\emptyset)$ является характерным.

Нам нужны некоторые определения и леммы.

Определение 4.1. Пусть дано табло T на множестве атрибутов

$$R = \{A_1, A_2, \ldots, A_n\}.$$

Для каждой строки $u\in T$ рассматривается так называемая симметричная u строка u' определяющаяся следующим образом:

$$u'(A_i) = \begin{cases} b_i, & \text{если} \quad u(A_i) = a_i \\ a_i, & \text{если} \quad u(A_i) = b_i. \end{cases}$$

Другим словами, u' получается из u изменением символов a на b и наоборот.

Определение 4.2. Табло T называется симметричным если для $\forall u\in T, u'\in T$.

Лемма 4.1. *Пусть дано симметричное табло T на множестве атрибутов $R=\{A_1, A_2, \ldots, A_n\}$ и пусть g набор подмножеств $Y_1, Y_2, \ldots, Y_p$ множества R таких, что $R=Y_1 Y_2 \ldots Y_p$. Тогда табло $T(g)=m_g(T)$ симметрично.*

Доказательство. Пусть $u\in T(g)$, докажем $u'\in T(g)$. По определению 2.5 из $u\in T(g)$ следует существование таких $u_i\in T$, что $u(Y_i)=u_i(Y_i)$, $1\leqq i\leqq p$. Так как T симметрично то для каждой u_i должно существовать ее симметричная строка $u_i'\in T$, $1\leqq i\leqq p$. По определению 4.1 мы имеем $u'(Y_i)=u_i'(Y_i)$, $1\leqq i\leqq p$, следовательно по определению 2.5 $u'\in T(g)$. □

Определение 4.3. Пусть $g=\{Y_1, Y_2, \ldots, Y_p\}$ — набор подмножеств конечного непустого множества $R=\{A_1, A_2, \ldots, A_n\}$, где $R=Y_1 Y_2 \ldots Y_p$. Для каждого подмножества $X\subseteq R$ построим множество $L_g(X)=L$ следующим рекурсивным методом:

1. $L:=X$
2. если $L\cap Y_i\neq\emptyset$ для некоторого i, $1\leqq i\leqq p$, то

$$L:=L\cup Y_i$$

3. L расширяется только по правилу 2.

Легко видеть, что отображение L_g обладает следующими свойствами:

Свойства отображения L_g. Для всяких g, Y_i и X удовлетворящих перечисленным в определении 4.3 условиям:

(L1) $L_g(X)$ существует и единственно

(L2) L_g является замкнутой функцией $2^R \to 2^R$, т. е. L_g обладает следующими свойствами:

(L2.1) $X \subseteq L_g(X)$

(L2.2) если $X \subseteq Y$, то $L_g(X) \subseteq L_g(Y)$

(L2.3) $L_g(L_g(X)) = L_g(X)$

(L3) $L_g(X)$ представляет собою оъединение некоторых Y_i, т. е. для каждого i, $1 \leqq i \leqq p$, выполняется только одно из следующих двух соотношений:

либо $Y_i \subseteq L_g(X)$,

либо $Y_i \cap L_g(X) = \emptyset$

(L4) $L_g(\emptyset) = \emptyset$.

Пусть дана $PC\ a = \langle R, C \rangle$. Мы рассмотрим C как оъединение двух не пересекающихся множеств F и J, где

F — множество $\Phi З$ на R,

J — множество $CЗ$ на R,

притом вместо $a = \langle R, C \rangle$ в нужном случае будем писать $a = \langle R, FJ \rangle$.

Для каждой $\Phi З\ f = X \to Y \in F$ будем обозначать $\mathrm{left}(f) = X$ и $\mathrm{right}(f) = Y$ и называть соответственно левой и правой частями $\Phi З\ f$.

Теорема 4.1. *Для всякой* $PC\ a = \langle R, FJ \rangle$ *справедливо:*

$$(4.1) \qquad (\emptyset)_a^+ \supseteq (\emptyset)_{\langle R, F\rangle}^+ \cup \Big(\bigcup_{\substack{f \in F \\ g \in J}} (\mathrm{right}(f) - L_g(\mathrm{left}(f))) \Big)$$

Доказательство. Так как $F \subseteq FJ = C$, то $(\emptyset)_{\langle R, F\rangle}^+ \subseteq (\emptyset)_a^+$. Будем доказывать, что для $\forall f \in F$, $\forall g \in J$

$$\mathrm{right}(f) - L_g(\mathrm{left}(f)) \subseteq (\emptyset)_a^+.$$

Пусть $f = X \to Z$ и $g = {}^*[Y_1, Y_2, \ldots, Y_p]$. Докажем $E = Z - L_g(X) \subseteq (\emptyset)_a^+$. Обозначим $L = L_g(X)$. Так как $\mathrm{chase}_a(T_\varnothing)$ не зависит от порядка применения Φ- и C-правил, то сначала мы применим C-правило g к T_0 чтобы получить T_1, затем применим Φ-правило $X \to Z$ к T_1 чтобы получить T_2. Другими словами, будем вычислять

$$T_1 = T_0(g) = m_g(T_0)$$

$$T_2 = T_1(X \to Z)$$

и будем доказывать, что в T_2 все A-столбцы, где $A \in E$ являются характерными. По замечанию, изложенному в подготовке к доказательству теоремы 3.1 характерные столбцы остаются характерными при применении Φ- и C-правил, мы будем заключать $E \subseteq (\emptyset)_a^+$.

Напомним, что $T_0 = T_\varnothing$ следовательно T_0 содержит только две строки $u_d = \langle a_1, a_2, \ldots, a_n \rangle$ и $u_n = \langle b_1, b_2, \ldots, b_n \rangle$.

Пусть $K = R - L$. Так как $L \cap K = \emptyset$ и операция соединение коммутативна и ассоциативна то мы можем представить T_1 как декартово произведение двух табло P и Q

$$T_1 = P * Q,$$

где $P = \underset{y_i \subseteq L}{*} T_0[y_i]$ и $Q = \bigcup_{y_i \cap L = \emptyset} T_0[y_j]$.

Заметим что P является табло на L а Q — табло на K, кроме того по лемме 4.1 P и Q являются симметричными табло. Следовательно для $\forall t \in P$ и $\forall u, v \in Q$ мы имеем $t \cdot u, t \cdot v \in T_1$ и если $u \neq v$ то $t \cdot u \neq t \cdot v$, где $t \cdot u$ понимается как конкатенация двух строк t и u.

Пусть $A_k \in E$ и $u \in T_1$ для которых $u(A_k) = b_k$. Так как $E = Z - L$ то $E \subseteq Z$ следовательно $A_k \in Z$. Из $E \cap L = \emptyset$ и $K = R - L$ следует $E \subseteq K$. Так как Q симметрично то из $u(K) \in Q$ следует существование $v \in Q$ такое, что $v(A_k) = a_k$.

Обозначим $t = u(L)$ и рассмотрим строку $w = t \cdot v$. Так как $X \subseteq L$ то имеем $u(X) = t(X) = w(X)$, $w(A_k) = a_k$. Это означает, что мы можем применить Φ-правило $X \to Z$ к строкам u и w в табло T_1 чтобы изменить значение $u(A_k) = b_k$ на a_k. Так как A_k является произвольным атрибутом в E то после применения Φ-правила мы получим табло T_2 в котором все A-столбцы, $A \in E$ являются характерными и поэтому в $\mathrm{chase}_a(T_0)$ они будут оставаться характерными. Следовательно $E \subseteq (\emptyset)_a^+$. □

Приведем пример в котором:

1. Знак сравнения в соотношении (4.1) является строгим,
2. $(\emptyset)^+_{\langle R, F \rangle}$ не является подмножеством множества

$$\bigcup_{\substack{f \in F \\ g \in J}} \left(\mathrm{right}(f) - L_g(\mathrm{left}(f))\right).$$

Пример 4.1. Дана $PC\ a = \langle R, FJ \rangle$, где

$R = 12345$,

$F = \{\emptyset \to 4,\ 4 \to 5,\ 3 \to 1,\ 1 \to 2\}$,

$J = \{{}^*[123, 45],\ {}^*[124, 3, 15]\}$.

Применив, например алгоритм нахождения замыкания для случая $\Phi 3$ [4] мы имеем $(\emptyset)^+_{\langle R, F \rangle} = 45$. Продолжем вычислять:

f	g	$L_g(\mathrm{left}(f))$	$\mathrm{right}(f) - L_g(\mathrm{left}(f))$
$\emptyset \to 4$	*[123, 45]	$\emptyset$	4
$\emptyset \to 4$	*[124, 3, 15]	$\emptyset$	4
$4 \to 5$	*[123, 45]	45	$\emptyset$
$4 \to 5$	*[124, 3, 15]	1245	$\emptyset$
$3 \to 1$	*[123, 45]	123	$\emptyset$
$3 \to 1$	*[124, 3, 15]	3	1
$1 \to 2$	*[123, 45]	123	$\emptyset$
$1 \to 2$	*[124, 3, 15]	1245	$\emptyset$

Следовательно

$$\bigcup_{\substack{f \in F \\ g \in J}} \left(\mathrm{right}(f) - L_g(\mathrm{left}(f))\right) = 14,$$

В результате получим

$$(\emptyset)^+_{\langle R,F\rangle} \cup \Big(\bigcup_{\substack{f\in F\\ g\in J}} (\mathrm{right}(f) - L_g(\mathrm{left}(f)))\Big) = 145.$$

С другой стороны, применив табло мы получим

$$(\emptyset)^+_a = 1245.$$

5. Вычисление замыкания

В этом пункте предлагается общая схема вычисления замыкания множества атрибутов в *РС* содержащей *ФЗ* и *СЗ* и детально исследуется алгоритм вычисления выражения

$$\bigcup_{\substack{f\in F\\ g\in J}} \big(\mathrm{right}(f) - L_g(\mathrm{left}(f))\big). \tag{5.1}$$

Постановка Задачи. Дана $PC\ a = \langle R, FJ\rangle$, где

R — множество атрибутов,
F — множество *ФЗ* на R,
J — множество *СЗ* на R.

Для каждого заданного множества атрибутов $X \subseteqq R$ трбуется вычислить его замыкание $(X)^+_a$ относительно $PC\ a$.

На основе следствия 3.1 и теоремы 4.1 мы предлагаем следующую схему вычисления $(X)^+_a$.

Схема 5.1. Функция CLOSURE.
Вход: $PC\ a = \langle R, FJ\rangle$ и $X \subseteqq R$
Выход: $(X)^+_a$
CLOSURE (X, a)
ЭТАП 1. Преобразовать $PC\ a$ в $a \setminus X$ по формулам:
$R := R - X$;
$F := F \setminus X$;
удалить из F повторяющиеся *ФЗ* и *ФЗ* вида $Y \to \emptyset$;
$J := J \setminus X$;
удалить из каждой *СЗ* в J повторяющиеся компоненты и пустые компоненты;
удалить из J повторяющиеся *СЗ* и *СЗ* содержащие компонент равный R и *СЗ* вида $*[\emptyset]$;
ЭТАЛ 2. Вычислить $P := (\emptyset)^+_{\langle R,F\rangle}$ по алгоритму в [4];
ЭТАП 3. Вычислить

$$Q := \bigcup_{\substack{f\in F\\ g\in J}} \big(\mathrm{right}(f) - L_g(\mathrm{left}(f))\big).$$

ЭТАП 4. Преобразовать $PC\ a$ в $a \setminus (P \cup Q)$ точно как ЭТАП 1;

ЭТАП 5. Вычислить $E:=(\emptyset)_a^+$ по одному из алгоритмов указанных в [12];
ЭТАП 6. return $(X\cup P\cup Q\cup E)$;
СТОП.

Рассмотрим сложность по времени выполнения каждого этапа.
Пусть

$n=|R|$ = число атрибутов в R,
$k=|F|$ = число $\Phi 3$ в F,
c = число различных символов-атрибутов в F,
$d=|J|$ = число $C3$ в J,
b = среднее число компонентов в каждой $C3$ в J.

Заметим, что объединение всех компонентов в каждой $C3$ равно R, т. е. для представления каждой $C3$ требуется робно n символов. Длина представления всех $\Phi 3$ в F имеет порядок $\mathcal{O}(kc)$. Длина представления одной $C3$ в J составляет величину $\mathcal{O}(bn)$, а для всех $C3$ в $J-\mathcal{O}(dbn)$, следовательно длина входа l для функции CLOSURE составляет $\mathcal{O}(kc+dbn)$. Далее предлагаем, что время выполнения каждой операции над множествами атрибутов является постоянной величинной t.

ЭТАП 1. Операция $R-X$ требует t единиц времени.
Операция $F\diagdown X$ трбует $\mathcal{O}(k)$.
Операция $J\diagdown X-\mathcal{O}(db)$.

Для удаления повторяющихся объектов можно применить алгоритм сортировки со сложностью по времени $m\log_2 m$, где $m=kc$ для F и $m=dbn$ для J. Следовательно временная сложность ЭТАПа 1 не превышает $\mathcal{O}(l\cdot\log_2 l)$.

ЭТАП 2. В работе [4] описывается алгоритм вычисления $(M)^+_{\langle R,F\rangle}$ со сложностью пропорциональной длине входа, следовательно ЭТАП 2 выполняется за время $\mathcal{O}(kc)$.
ЭТАП 3. Будем рассматривать отдельно.
ЭТАП 4. $\mathcal{O}(l\log_2 l)$ как ЭТАП 1.
ЭТАП 5. По работе [12] существуют алгоритмы вычисления $(\emptyset)_a^+$ с полиномиальной временной сложностью.
ЭТАП 6. $4t$

Переходим к детальной реализации ЭТАПа 3 для функции CLOSURE.

Ради удобства в изложении мы будем рассматривать каждую $C3$ $g={}^*[Y_1, Y_2, \ldots, Y_p]$ как набор подмножеств множества R $g=\{Y_1, Y_2, \ldots, Y_p\}$ и поэтому будем писать $Y_i\in g$, $1\leqq i\leqq p$ в место длиной фразы «Y_i является компонентом в g...».

Сначала приводим некоторые леммы.

Лемма 5.1. *Для*

$$\forall g\in J,\quad \forall X\subseteq R,\quad \forall Y\in g:$$

$$Y\subseteq L_g(X)\Leftrightarrow \exists Y_0, Y_1, \ldots, Y_k\in g:$$

$$Y_0\cap X\neq\emptyset;\quad Y_i\cap Y_{i+1}\neq\emptyset,\quad 0\leqq i\leqq k-1;\quad Y_k=Y.$$

Доказательство. Непосредственно следует из определения 4.3. □

Для каждой *СЗ* g в J определим следующее отношение $\sim$ в R:
Для $\forall A, B \in R \;\; A \sim B \Leftrightarrow L_g(A) = L_g(B)$.

Очевидно отношение $\sim$ является отношением эквивалентности на R и поэтому R разделяется на не пересекающиеся подмножества $R_{g,1}, R_{g,2}, \ldots, R_{g,m}$.

Лемма 5.2. Для $\forall X \subseteqq R$ и $\forall g \in J$:

$$L_g(X) = \bigcup_{R_{g,j} \cap X \neq \emptyset} R_{g,j}.$$

Доказательство. Заметим, что по свойству отображения L_g, $L_g(X)$ и каждый $R_{g,j}$ представляют собою объединения некоторых компонентов в g. Пусть $Y \in g$ и $Y \subseteqq L_g(X)$, по лемме 5.1 мы имеем $\exists Y_0, Y_1, \ldots, Y_k \in g$:

$$Y_0 \cap X \neq \emptyset; \quad Y_i \cap Y_{i+1} \neq \emptyset, \quad 0 \leqq i \leqq k-1; \quad Y_k = Y.$$

Возьмем любой элемент A из $Y_0 \cap X$. По лемме 5.1 $Y \subseteqq L_g(A) = R_{g,j}$ для некоторого j. Кроме этого, из $A \in L_g(A) \cap X$ следует $R_{g,j} \cap X \neq \emptyset$.

Обратно, пусть имеем $R_{g,j} \cap X \neq \emptyset$ для некоторого j, докажем $R_{g,j} \subseteqq L_g(X)$. Возьмем любой элемент A из $R_{g,j} \cap X$. По свойству разбиения по эквивалентности, из $A \in R_{g,j}$ следует $L_g(A) = R_{g,j}$. Пусть $Y \subseteqq R_{g,j} = L_g(A)$. По лемме 5.1 должно существовать $Y_0, Y_1, \ldots, Y_k \in g$ такие, что $Y_0 \cap \{A\} \neq \emptyset$; $Y_i \cap Y_{i+1} \neq \emptyset$, $0 \leqq i \leqq k-1$; $Y_k = Y$. Из $A \in X$ следует $Y_0 \cap X \neq \emptyset$. Опять по лемме 5.1 имеем $Y \subseteqq L_g(X)$. □

Теперь построим алгоритм реализации функции BASIS (g), которая выдает классы разбиения множества R по каждой *ФЗ* g из J.

АЛГОРИТМ 5.1. Функция BASIS

Вход: — Множество атрибутов R
— *СЗ* g над R

Выход: Разбиение множества R на классы по отношению эквивалентности $A \sim B \Leftrightarrow L_g(A) = L_g(B)$.

```
BASIS (R, g)
1. Инициализация
    |for каждый атрибут A в R do
    |  SET (A):=∅; COUNT (A):=0;
    |endfor A;
    |for каждый компонент Y в g, do
    |    |for каждый атрибут A в Y with Y⊈SET (A) do
    |    |  SET (A):=SET (A)∪Y; COUNT (A):=COUNT (A)+I
    |    |endfor A
    |endfor Y;
```

2. Вычисление
```
    P:=R; M:=∅;
  while P≠∅ do
    выбираем A в P;
    P:=P−A;
    T:=SET(A); BUF:=T−A; COUNT(A):=0;
      while BUF≠∅ do
        выбираем B в BUF;
        BUF:=BUF−B; P:=P−B;
          if COUNT(B)≧2 then
            ADD:=SET(B)−T; T:=T∪ADD;
            BUF:=BUF∪ADD; COUNT(B):=0;
          endif COUNT(B);
      endwhile BUF;
    M:=M∪{T};
  endwhile P;
3.  return (M);
end BASIS.
```

Теорема 5.1. (1) *АЛГОРИТМ 5.1 точно вычисляет разбиение множества R на классы по отношению эквивалентности:* $A\sim B\Leftrightarrow L_g(A)=L_g(B)$.

(2) *Сложность АЛГОРИТМа 5.1 по времени для ввода величины размерности l имеет порядок* $\mathcal{O}(l)$.

Доказательство. (1) Цикл for Y вычисляет для каждого атрибута $A\in R$ множество $\mathrm{SET}(A)=\bigcup\limits_{\substack{Y\in g\\ A\in Y}} Y$. Кроме этого он считает число компонентов в g, которые действительно участвуют в процессе расширения $\mathrm{SET}(A)$ и вносит результат в счетчик $\mathrm{COUNT}(A)$. При каждом проходе цикл while P вычисляет один класс эквивалентности T с представителем $A\in P$ по правилу, описанному в определении отображения L_g 4.3. В отличии от определения 4.3 АЛГОРИТМ 5.1 рассматривает не один компонент в g а целую группу компонентов $\mathrm{SET}(B)$, а именно T образуется и расширяется следующим образом:

(1) $T:=\mathrm{SET}(A)$

(2) если $T\cap\mathrm{SET}(B)\neq\emptyset$ для некоторого $B\in P$, то $T:=T\cup\mathrm{SET}(B)$

(3) T расширается только по правилу (2).

Для избежания повторного вычисления классов эквивалентности мы поддерживаем следующие принципы:

— каждая группа компонентов $\mathrm{SET}(A)$ может участвовать в вычислении не более одного раза. Этот принцип реализуется с помощью операции удаления из P каждого уже рассмотренного атрибута A $P:=P-A$ и состояния счетчика $\mathrm{COUNT}(A)$. $\mathrm{COUNT}(A)=0$ означает, что группа $\mathrm{SET}(A)$ уже рассмотрена. $\mathrm{COUNT}(A)=1$ означает, что атрибут A входит только в один компонент заданной $C3$ g. Если $B\in\mathrm{BUF}$, то из $\mathrm{BUF}\subseteqq T$ следует $B\in T$. Из $B\in T$ следует, что при расширении T, $\mathrm{SET}(B)$ уже есть в T если только $\mathrm{COUNT}(B)=1$. Таким образом T может расшириться в том случае, когда $B\in T$ и $\mathrm{COUNT}(B)\geqq 2$,

— только новые прибавленные к T атрибуты могут подаваться на рассмотрение. Этот принцип реализуется с помощью переменных BUF и ADD. ADD собирает новые прибавленные к T атрибуты и вносит в BUF.

Так как $P=R$ и R конечное множество, а операции $P:=P-A$, $\mathrm{BUF}:=\mathrm{BUF}-B$ и $P:=P-B$ уменшают P и BUF по крайней мере один элемент за каждый проход циклов while P и while BUF, кроме того в BUF прибавляются только не рассмотренные атрибуты то циклы while BUF и while P являются конечными. Следовательно АЛГОРИТМ 5.1 будет заканчиваться и давать в результате множество эквивалентных классов по отношению $\sim$ на R.

(2) В шаге инициализации цикл for A занимает время $\mathcal{O}(n)$, где $n=|R|$. Цикл for Y занимает время $\mathcal{O}(bn)$, где $b=|g|$.

В шаге вычисления каждый атрибут рассматривается один раз, операции над множествами занимают постояное количество времени поэтому шаг вычисление занимает время $\mathcal{O}(n)$.

Заметим, что длина входа АЛГОРИТМа 5.1 есть величина $\mathcal{O}(bn)$. □

Переходим к реализации ЭТАПа 3 СХЕМы 5.1. Построим функцию PART (R, F, J) которая вычисляет выражение (5.1).

АЛГОРИТМ 5.2. Функция PART

Вход: — Множество атрибутов R.
Множество *ФЗ* F.
Множество *СЗ* J.

Выход: Значение выражения (5.1)

```
PART (R, F, J)
Q:=∅;
for каждая СЗ g в J do
  M:=BASIS (R, g);
    for каждая ФЗ Y→Z в F do
      L:=∅;
        for каждый компонент T в M with T∩Y≠∅ do
          L:=L∪T;
        endfor T;
      Q:=Q∪(Z−L);
    endfor Y→Z;
endfor g;
  return (Q);
end PART.
```

Теорема 5.2 (1) *Алгоритм* PART *точно вычисляет значение выражения* (5.1).

(2) *Сложность алгоритма* PART *по времени имеет пордок* $\mathcal{O}(dkbn)$.

Доказательство. (1) Непосредственно следует из леммы 5.2.

(2) По теореме 5.1, для каждой $СЗ_g$ из J функция BASIS $(R. g)$ занимает время $\mathcal{O}(bn)$. Следовательно время вычисления d раз функции BASIS (R, g) для всех *СЗ* $g \in J$ составляет $\mathcal{O}(dbn)$. Так как каждый компонент T в M пред-

ставляет собою объединение некоторых компонентов в g, то мы имеем $|M| \leqq \leqq |g| = b$. Следовательно цикл for T занимает время $\mathcal{O}(bn)$. Цикл for $Y \to Z$, тогда занимает время $\mathcal{O}(kbn)$. Следовательно цикл for g занимает время $\mathcal{O}(dbn + dkbn)$ или $\mathcal{O}(dkbn)$. □

Список литературы

[1] A. V. AHO, C. BEERI and J. D. ULLMAN, The Theory of Joins in Relational Databases. *ACM TODS* **4:3**, September 1979, 297—314.

[2] W. W. ARMSTRONG, Dependency Structures of Data Base Relationships. 1974 *IFIP Cong., Geneva, Switzerland*, 580—583;

[3] C. BEERI, On the Membership Problem for Functional and Multivalued Dependencies in Relational Databases. *ACM TODS* **5:3**, September 1980, 241—259.

[4] C. BEERI and P. A. BERNSTEIN, Computational Problems Related to the Design of Normal Form Relational Schemas. *ACM TODS* **4:1**, March 1979, 30—59.

[5] C. BEERI, R. FAGIN and J. H. HOWARD, A Complete Axiomatization for Functional and Multivalued Dependencies. In Database Relations. *ACM SIGMOND Conf.* 1977. 47—61.

[6] E. F. CODD, A Relational Model of Data for Large Shared Data Banks. *CACM* **13:6**, June, 1970, 377—387.

[7] E. F. CODD, Further Normalization of the Database Relational Model. In Data Base Systems ed. Rustin, R. Englewood Cliffs, *NJ: Prentice-Hall*, 1972, 33—64.

[8] C. DELOBEL and R. G. CASEY, Decomposition of a Data Base and the Theory of Boolean switching Functions. *IBM J. of Research and Development*, **17:5**, *Armonk, NY*, September 1973, 374—386.

[9] J. DEMETROVICS, HO THUAN, NGUYEN XUAN HUY and LE VAN BAO, Translation of Relation Schemes, Balanced Relation Schemes and the Problem of Key Representation. *J. Inf. Process. Cybern., EIK* **23** (1987) 2/3, 81—97.

[10] J. DEMETROVICS, HO THUAN, NGUYEN XUAN HUY and LE VAN BAO, Balanced Relation Scheme and the Problem of Key Representation. *MTA SZTAKI, Közlemények* **32**/1985, 51—80.

[11] R. FAGIN, Multivalued Dependencies and a New Normal Form for Relational Databases. *ACM TODS* **2:3**, September 1977, 262—278.

[12] D. MAIER, The Theory of Relational Database. *Rockville, MD: Computer Science Press*, 1983, 637 p.

[13] D. MAIER, Minimum Covers in the Relational Database Model. *JACM* **27:4**, October 1980, 664—674.

[14] D. MAIER, A. O. MENDELZON and Y. SAGIV, Testing Implications of Data Dependencies. *ACM TODS* **4:4**, December 1979, 455—469.

[15] D. MAIER, Y. SAGIV and M. YANNAKAKIS, On the Complexity of Testing Implications of Functional and Join Dependencies. *JACM* **28:4**, October 1981, 680—695.

[16] J. RISSANEN, Independent Components of Relations. *ACM TODS* **2:4**, December 1977, 317—325.

[17] E. SCIORE, A Complete Axiomatization for Full Join Dependencies. *JACM*, **29:2**, April 1982, 373—393.

ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ
ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
И АВТОМАТИКИ АН ВНР
(БУДАПЕШТ)

(Поступило 20. VI. 1988 г.)