MODELING AND ANALYSIS OF INFORMATION SYSTEMS, VOL. 31, NO. 1, 2024
journal homepage: www.mais-journal.ru

THEORY OF DATA

Minimal Coverage of Generalized Typed Inclusion Dependencies in Databases

S. V. Zykin[1]

DOI: 10.18255/1818-1015-2024-1-78-89

[1]Sobolev institute of mathematics SB RAS, Novosibirsk, Russia

MSC2020: 68P15
Research article
Full text in Russian

Received February 12, 2024
Revised February 26, 2024
Accepted February 28, 2024

The paper discusses the theory and algorithms necessary to construct a minimal covering of generalized typed inclusion dependencies. Traditionally, the construction of minimal covering is used for all types of dependencies in order to obtain a non-redundant and consistent database design. Generalized inclusion dependencies correspond to referential integrity constraints, when several main and several external relations are involved in one constraint, which corresponds to an ultragraph edge. In previous work, based on a study of the properties of dependencies, a system of axioms was presented with proof of consistency and completeness. In this work, the closures were studied for generalized typed inclusion dependencies. An algorithm for constructing closures has been developed and its correctness has been proven. The results obtained are further used to develop an algorithm for constructing a minimal covering. At the end of the article, examples are presented that demonstrate the operation of the algorithms.

Keywords: database; generalized inclusion dependencies; minimal covering

INFORMATION ABOUT THE AUTHORS

Zykin, Sergey V. (corresponding author)	ORCID iD: 0000-0002-0576-2149. E-mail: szykin@mail.ru Leading researcher, Dr. Sc.

Funding: State task IM SB RAS, project No. FWNF-2022-0016.

For citation: S. V. Zykin, “Minimal coverage of generalized typed inclusion dependencies in databases”, *Modeling and Analysis of Information Systems*, vol. 31, no. 1, pp. 78–89, 2024. DOI: 10.18255/1818-1015-2024-1-78-89.

МОДЕЛИРОВАНИЕ И АНАЛИЗ ИНФОРМАЦИОННЫХ СИСТЕМ, ТОМ 31, № 1, 2024
сайт журнала: www.mais-journal.ru

THEORY OF DATA

Минимальное покрытие обобщенных типизированных зависимостей включения в базах данных

С. В. Зыкин[1]
DOI: 10.18255/1818-1015-2024-1-78-89

[1]Институт математики им. С.Л. Соболева СО РАН, Новосибирск, Россия

УДК 004.652.4
Научная статья
Полный текст на русском языке

Получена 12 февраля 2024 г.
После доработки 26 февраля 2024 г.
Принята к публикации 28 февраля 2024 г.

В статье рассматривается теория и алгоритмы, необходимые для построения минимального покрытия обобщенных типизированных зависимостей включения. Традиционно аппарат построения минимальных покрытий используется для всех видов зависимостей с целью получения не избыточного и непротиворечивого проекта базы данных. Обобщенные зависимости включения соответствуют ссылочным ограничениям целостности, когда в одном ограничении участвуют несколько главных и несколько внешних отношений, что соответствует ребру ультраграфа. В предыдущей работе на основе исследования свойств зависимостей представлена система аксиом с доказательством непротиворечивости и полноты. В данной работе проведены исследования замыканий для обобщенных типизированных зависимостей включения. Разработан алгоритм построения замыканий, доказана его корректность. Полученные результаты далее используются для разработки алгоритма построения минимального покрытия. В конце статьи представлены примеры, которые демонстрируют работу алгоритмов.

Ключевые слова: база данных; обобщенные зависимости включения; минимальное покрытие

ИНФОРМАЦИЯ ОБ АВТОРАХ

Зыкин, Сергей Владимирович (автор для корреспонденции)	ORCID iD: 0000-0002-0576-2149. E-mail: szykin@mail.ru Вед. науч. сотр., доктор техн. наук

Финансирование: Госзадание ИМ СО РАН, проект № FWNF-2022-0016.

Для цитирования: S. V. Zykin, “Minimal coverage of generalized typed inclusion dependencies in databases”, *Modeling and Analysis of Information Systems*, vol. 31, no. 1, pp. 78–89, 2024. DOI: 10.18255/1818-1015-2024-1-78-89.

Введение

Теоретической основой ссылочных ограничений целостности в базах данных являются зависимости включения. Они позволяют получить согласованное и не избыточное множество связей на схеме базы данных (БД) за счет построения минимальных покрытий.

Развитию теории зависимостей включения в настоящее время уделяется внимание с точки зрения взаимодействия с другими видами зависимостей в БД. С другой стороны, вопросы исследования новых видов ссылочных ограничений, которые предлагает практика, не потеряли своей актуальности.

Введем следующие обозначения: пусть $U = \{A_1, A_2, \ldots, A_n\}$ — множество атрибутов, R_i — отношения (таблицы) в БД, которые определены на множестве U, $[R_i]$ — схема отношения R_i (атрибуты отношения R_i), $[R_i] \subseteq U, 1 \leq i \leq k$, $R = (R_1, R_2, \ldots, R_k)$ — БД, $S = \{[R_1], [R_2], \ldots, [R_k]\}$ — схема БД [1, 2], для которой не заданы ссылочные ограничения целостности. Пусть $R_i[X]$ — проекция отношения R_i по атрибутам X и $t_i[X]$ кортеж проекции $R_i[X]$.

Впервые определение зависимости включения было представлено в работе [3].

Определение 1. Пусть R_i и R_j — отношения БД (возможно совпадающие), для которых заданы два множества атрибутов $V \subseteq [R_i]$ и $W \subseteq [R_j]$, с одинаковой мощностью. Тогда соотношение $R_i[V] \subseteq R_j[W]$ называется зависимостью включения.

Замечание. Если имеет место равенство $V = W$, то зависимость в определении 1 является типизированной (typed), в противном случае — нетипизированной. Для типизированных зависимостей атрибут $A_i \in V$ соответствует атрибуту $A_i \in W$ независимо от места расположения во множествах V и W. Для нетипизированных зависимостей первый атрибут по порядку множества V соответствует первому атрибуту множества W, второй – второму и т. д. В этом заключается принципиальное различие этих двух видов зависимостей. Следует заметить, что в языке SQL (Structured Query Language) атрибут идентифицируется своим именем, а не местом расположения в записи, что соответствует типизированным зависимостям.

Проблема построения проекта БД исследуется достаточно давно. Общий подход заключается в использовании зависимостей данных, которые подвергаются проверке на избыточность, противоречивость и полноту. После аксиоматизации появляется возможность использования зависимостей для построения минимальных покрытий, которые затем используются при формировании схемы БД. Основной проблемой при этом становится выводимость. Препятствием является взаимодействие зависимостей включения с функциональными зависимостями.

Неоднократно исследователями осуществлялись попытки решить проблему выводимости совместно для функциональных зависимостей и зависимостей включения. Однако, отсутствие полной системы аксиом не позволяет решить данную проблему [3—5], за исключением некоторых частных случаев, где полная аксиоматизация [6, 7] была получена. В работе [8] сформулированы условия существования нормальной формы зависимостей включения (IDNF), которая основана на нормальной форме Бойса-Кодда (BKNF), дополненная условием ацикличности.

В работе [9] рассматривается задача совместной аксиоматизации функциональных зависимостей, зависимостей включения и частного случая многозначных зависимостей, у которых компоненты базиса названы независимыми атомами. В результате исследований было получено, что конечные и неограниченные задачи выводимости для комбинированного класса атомов, унарных функциональных зависимостей и унарных зависимостей включения аксиоматизируемы и разрешимы за полиномиальное время.

Большое количество исследований было посвящено разработке алгоритмов определения обычных и условных зависимостей включения в существующих БД [10—16]. Для этого используются различные источники информации, например, свойства схемы БД, трудоемкие алгоритмы анализа сохраненных данных в БД, обработка текущих запросов к БД. Такие подходы требуют дополнительной проверки результатов, поскольку полная автоматизация нереализуема из-за специфики бизнес-правил в приложениях. Польза таких исследований в том, что частично автоматизируется трудоемкий процесс поиска неизвестных зависимостей включения.

Практически значимые исследования посвящены неопределенным значениям в БД. Выводимость при наличии неопределенных значений исследуется в работах [17—20]. Взаимодействие функциональных зависимостей и нетипизированных зависимостей включения стало препятствием для получения полной и непротиворечивой системы аксиом (решение есть только для унарных зависимостей). Однако, удаление семантических несоответствий, прежде всего синонимии, на этапе проектирования БД позволяет использовать типизированные зависимости включения [20], которые не взаимодействуют с функциональными зависимостями.

В данной работе рассматривается проблема выводимости для обобщенных типизированных зависимостей включения, для которых в работе [21] получена полная и непротиворечивая система аксиом.

1. Предварительные сведения

В работе [21] рассмотрен практический пример схемы БД, в котором одной связью объединяются несколько отношений. Эта связь содержит несколько главных отношений, первичные ключи которых совпадают, и несколько внешних отношений, в которых первичные ключи главных отношений становятся внешними ключами. При этом внешние ключи могут принимать только те значения, которые уже есть хотя бы в одном главном отношении. В примере также показано, что практический смысл имеет ситуация, когда атрибуты внешних ключей принимают неопределенные значения (*Null*), если они не являются компонентами первичного ключа во внешнем отношении. Заметим, что с точки зрения теории ссылочная целостность не обязательно должна опираться на первичные ключи, это может быть любой другой набор атрибутов главного отношения, допускающий наличие неопределенных значений.

Чтобы учитывать неопределенные значения, в работе [20] было предложено условие соответствия, которое устанавливает частичный порядок для кортежей.

Определение 2. Кортеж $t_i[X]$ *соответствует* кортежу $t_j[X]$ для множества атрибутов X ($t_j[X] \preccurlyeq t_i[X]$):

- если $t_i[A_l] \neq Null$, тогда $t_j[A_l] = t_i[A_l]$ либо $t_j[A_l] = Null$;
- если $t_i[A_l] = Null$, тогда $t_j[A_l] = Null$,

для каждого атрибута $A_l \in X$.

Соответствие кортежей в определении 2 удовлетворяет условию транзитивности: если $t_j[X] \preccurlyeq t_i[X]$ и $t_i[X] \preccurlyeq t_m[X]$, тогда $t_j[X] \preccurlyeq t_m[X]$. На основе определения 2 в работе [20] вводится определение типизированной зависимости включения.

Определение 3. Типизированная зависимость включения $\sigma = R_j[X] \subsetneq R_i[X]$ между главным отношением $R_i[X]$ и внешним отношением $R_j[X]$ по атрибутам X выполнена, если для любого кортежа $t_j[X] \in R_j[X]$ имеется соответствующий кортеж $t_i[X]$ в отношении $R_i[X]$.

Из определения 3 следует, что кортеж $t_j[X] \in R_j[X]$ может иметь несколько соответствующих кортежей в отношении $R_i[X]$. Когда происходит замена неопределенных значений в кортеже $t_j[X]$, то в $R_i[X]$ должен присутствовать, по крайней мере, один соответствующий кортеж, что ограничивает варианты замены.

В работе [21] введено обобщение типизированных зависимостей включения: $W[X] \subsetneq V$, где W — множество внешних отношений зависимости, V — множество главных отношений зависимости.

Оба множества отношений зависимости определены на наборе атрибутов X, поэтому достаточно указать X только слева.

Определение 4. Обобщенная типизированная зависимость включения $\sigma = W[X] \subsetneq V$ между главными отношениями V и внешними отношениями W по атрибутам X выполнена, если для каждого отношения $R_i \in W$ и каждого кортежа $t_i \in R_i$ существует отношение $R_j \in V$ и существует соответствующий кортеж $t_j \in R_j$, то есть выполнено условие: $t_i[X] \preccurlyeq t_j[X]$.

Приведенное условие в определении 4 соответствует ссылочной целостности в БД.

На основании исследования свойств зависимостей в [21] была получена следующая система аксиом.

Аксиома 1 (рефлексия). Если $W \subseteq V$, тогда $W[X] \subsetneq V$.

Аксиома 2 (проекция). Если $W[X] \subsetneq V$ и $Y \subseteq X$, тогда $W[Y] \subsetneq V$.

Аксиома 3 (объединение). Если $W[X] \subsetneq V$ и $S[X] \subsetneq V$, тогда $\{W \cup S\}[X] \subsetneq V$.

Аксиома 4 (транзитивность). Если $W[X] \subsetneq S$ и $S[X] \subsetneq V$, тогда $W[X] \subsetneq V$.

Здесь V, W, S — множества отношений БД, X, Y — множества атрибутов.

Для представленной системы аксиом были доказаны непротиворечивость (все выводимые за счет аксиом зависимости выполнимы) и полнота (все выполнимые зависимости выводимы).

С использованием аксиом 1–4 получены следующие правила, которые далее используются при построении замыканий.

Правило 1 (декомпозиция). Если $W[X] \subsetneq V$ и $S \subseteq W$, тогда $S[X] \subsetneq V$.

Действительно, применив сначала аксиому 1, а затем аксиому 4 получим результат.

Правило 2 (пополнение). Если $W[X] \subsetneq V$ и $V \subseteq S$, тогда $W[X] \subsetneq S$.

Заметим, правило 2 позволяет дополнять в правую часть зависимости произвольное количество отношений. В [21] это правило использовано при доказательстве полноты системы аксиом 1–4 (теорема 2).

В работе [21] доказаны следующие две теоремы.

Теорема 1 (надежность). *Система аксиом 1–4 надежна.*

Теорема 2 (полнота). *Система аксиом 1–4 полна.*

При построении замыканий важными являются свойства, которые не выполнены для обобщенных зависимостей включения.

Утверждение 1. Объединение двух (и более) правых частей зависимостей, даже если они имеют непустое пересечение, но не совпадают друг с другом, не является эквивалентным преобразованием:

$$W[X] \subsetneq S \text{ и } W[X] \subsetneq V \text{ не эквивалентно } W[X] \subsetneq \{V \cup V\}.$$

Действительно, пусть $S \not\subseteq V$ и $V \not\subseteq S$. Допустим, что $R_i \in S$ и $R_i \notin V$. Тогда в зависимости $W[X] \subsetneq \{V \cup V\}$ отсутствует необходимость принадлежности кортежей W одному из отношений V, что противоречит зависимости $W[X] \subsetneq V$. Аналогично доказывается следующее утверждение.

Утверждение 2. Из зависимости $W[X] \subsetneq S$ не следует $W[X] \subsetneq V$, если $V \subset S$.

Из этого можно сделать вывод, что правая часть обобщенной зависимости включения не допускает удаления отношений.

2. Замыкание множества зависимостей включения

Для поиска избыточных зависимостей традиционно используется построение замыканий [1, 20]. Для обнаружения зависимостей $\sigma = W[X] \subsetneq V$, которые являются логическим следствием множества обобщенных зависимостей включения Σ, используется выводимость $\Sigma \vdash \sigma$ за счет системы аксиом 1–4. Поскольку система аксиом является полной и непротиворечивой, то гарантируется выполнение выводимой зависимости σ, если БД удовлетворяет множеству зависимостей Σ. В этом случае зависимость σ можно удалить.

Рассмотрим определение и структуру замыкания множества типизированных обобщенных зависимостей включения.

Определение 5. Замыканием $V^+[X]$ множества отношений V по атрибутам X на зависимостях Σ будем называть множества отношений $W_1, W_2, \ldots, W_k$ таких, что зависимости $W_i[X] \subsetneq V$, $i = \overline{1,k}$, выводимы из Σ с использованием аксиом 1–4, другими словами $\Sigma \vdash W_i[X] \subsetneq V$, $i = \overline{1,k}$.

Оперирование множеством множеств отношений в замыкании затруднит определение выводимых зависимостей, поскольку остается открытым вопрос о выводимости за счет объединения подмножеств множеств отношений, уже находящихся в замыкании. Следующая лемма дает ответ на этот вопрос и существенно упрощает процесс построения замыкания.

Лемма 1. *Множества отношений $W_1, W_2, \ldots, W_k$, выводимые по атрибутам X из одного и того же множества отношений V, в замыкании $V^+[X]$ объединяются.*

Доказательство. Допустим, множества W_1 и W_2 принадлежат замыканию $V^+[X]$ и $W = \{W_1 \cup W_2\}$. По аксиоме объединения 3 выводима зависимость $W[X] \subsetneq V$. По аксиоме транзитивности 4 из V выводимо любое множество отношений S, для которого имеет место зависимость $S[X] \subsetneq \{W_1 \cup W_2\}$. По правилу 2 (пополнение) из множества $\{W_1 \cup W_2\}$ выводимо все, что выводимо из множеств W_1 и W_2 по отдельности. Следовательно, множества W_1 и W_2 в замыкании $V^+[X]$ можно заменить одним множеством W. Аналогичным способом к W присоединяются остальные множества $W_i[X]$, $i = \overline{3,k}$. □

Следствием леммы 1 является возможность представления замыкания $V^+[X]$ в виде одного множества отношений БД, что существенно упрощает его формирование и дальнейшее использование. Для проверки выводимости обобщенных зависимостей включения будем использовать аппарат построения замыканий [1, 20], адаптировав его для обобщенных типизированных зависимостей включения.

Построим функцию, реализующую алгоритм формирования замыкания. В качестве текущего замыкания и результата функции будем использовать список отношений $V^*[X]$. Входными параметрами функции являются: Σ^* — текущее множество зависимостей, на котором строится замыкание; V — исходное множество (список) отношений; X — множество атрибутов, по которому строится замыкание.

function CLOS_M(Σ^*,V,X)
CLOS_M $\leftarrow \emptyset$
$V^*[X] \leftarrow \emptyset$
for each R_i **from** V
    if $X \subseteq [R_i]$ **then** $V^*[X] \leftarrow V^*[X] \cup R_i$
end for
if $V^*[X] = \emptyset$ **then exit function**
insert $\leftarrow$ TRUE
while *insert*

insert $\leftarrow$ FALSE
for each $W_l[Y] \subsetneq V_l$ **from** Σ^*
if $V_l \subseteq V^*[X]$ **and** $W_l \nsubseteq V^*[X]$ **and** $X \subseteq Y$ **then**
$V^*[X] \leftarrow V^*[X] \cup W_l$
insert $\leftarrow$ TRUE
end if
end for
end while
CLOS_M $\leftarrow V^*[X]$
end function

Получим оценку вычислительной сложности алгоритма, реализованного функцией *CLOS_M*. Основной цикл *while* по своей организации является бесконечным. Однако, для выполнения очередной итерации необходимо дополнение к замыканию не менее одного множества отношений W_l. Следовательно, верхней границей количества итераций в алгоритме является произведение nk, где n — мощность множества зависимостей Σ^*, k — количество отношений в БД (при добавлении отношений к замыканию на каждой итерации).

Теорема 3 (замыкание)**.** *Функция* CLOS_M *корректно формирует замыкание* $V^+[X]$.

Доказательство. Рассмотрим произвольное множество отношений V и произвольное множество атрибутов X. В соответствии с леммой 1 замыкание представляет собой одно множество отношений. Компонентами этого множества являются отдельные отношения R_j. В теореме надо показать: 1) если $R_j \in V^*[X]$, тогда $R_j \in V^+[X]$, и 2) если $R_j \in V^+[X]$, тогда $R_j \in V^*[X]$. Другими словами $V^+[X]$ равно $V^*[X]$.

1. Покажем, что каждое отношение R_j, принадлежащее $V^*[X]$, также принадлежит $V^+[X]$. Замыкание $V^*[X]$ формируется в трех операторах:

a. $V^*[X] \leftarrow \emptyset$ ($V^*[X]$ равно $\emptyset$). Начальное множество $V^*[X]$ не содержит ни одного отношения, что может стать результатом построения замыкания, если в X есть атрибуты, которые отсутствуют в каждом отношении множества V. По аксиомам **A1-A4** в этом случае также ничего не выводимо: $V^+[X]$ равно $\emptyset$.

b. Далее в цикле к замыканию $V^*[X]$ присоединяются отношения R_i такие, что $R_i \in V$ и $X \subseteq [R_i]$. Замыкание $V^+[X]$ также будет содержать R_i, поскольку зависимость $R_i[X] \subsetneq V$ выводима по аксиоме рефлексии 1.

c. Для текущей зависимости $W_l[X] \subsetneq V_l$ к замыканию $V^*[X]$ присоединяется новое множество W_l: $V^*[X]$ равно $V^*[X] \cup W_l$. При этом должны быть выполнены условия: $V_l \subseteq V^*[X]$, $W_l \nsubseteq V^*[X]$ и $X \subseteq Y$. Пусть $R_i \in W_l \backslash V^*[X]$ одно из отношений, которое будет дополнено к $V^*[X]$. Используя индукцию, покажем, что $R_i \in V^+[X]$. В качестве базиса используем вариант (b). Предположим, что все отношения в $V^*[X]$ до рассмотрения зависимости $W_l[X] \subsetneq V_l$ принадлежат $V^+[X]$. Поскольку зависимость $W_l[X] \subsetneq V_l$ выводима по аксиоме проекции 2, зависимость $R_i[X] \subsetneq V_l$ выводима по правилу 1 (декомпозиция) и зависимость $V_l[X] \subsetneq V$ выводима в силу леммы 1, то по аксиоме транзитивности 4 имеем $R_i \in V^+[X]$. Следовательно, $V^*[X] \subseteq V^+[X]$.

2. Допустим, что отношение R_i принадлежит замыканию $V^+[X]$. В этом случае существует последовательность вывода, состоящая из k строк, в которой последней строке соответствует зависимость $W_k[X] \subsetneq V$ и $R_i \in W_k$. Пусть k равно 1. Единственной аксиомой, которая может быть использована при изначально пустом множестве $V^+[X]$, является аксиома рефлексии 1, $V^+[X] = W_1$, $W_1 \subseteq V$ и отношения множества W_1 определены на атрибутах X. В алгоритме отношения множества W_1 будут присоединены к $V^*[X]$ в первом цикле *for*. Допустим, что выводимые зависимости $W_j[X] \subsetneq V$, $j = \overline{1, k-1}$, удовлетворяют условию $W_j \subseteq V^*[X]$. На шаге k множество W_k было присоединено

к замыканию $V^+[X]$. Присоединение W_k не могло быть сделано за счет аксиомы рефлексии 1, так как множество W_k уже содержалось бы в $V^*[X]$. Аналогичная ситуация с аксиомой объединения 3: если W_k получено объединением двух множеств, то оба эти множества уже содержатся в $V^*[X]$, поскольку цепочка их вывода не превышает $k-1$ шагов.

Использование аксиомы проекции 3 подразумевает наличие зависимости $W_k[Y] \subsetneq V_l$, для которой $X \subseteq Y$ и $V_l \subseteq V^+[X]$. Сформулированные условия согласуются с условиями оператора *IF* во втором операторе цикла *for*. Следовательно, по алгоритму множество W_k будет присоединено к $V^*[X]$.

Пусть на шаге k при построении $V^+[X]$ была использована аксиома транзитивности 4 для зависимостей $W_k[X] \subsetneq S$ и $S[X] \subsetneq V_l$. Множества отношений V_l и S принадлежат $V^+[X]$, следовательно, они принадлежат $V^*[X]$, поскольку цепочка их вывода не превышает $k-1$ шагов. Перечисленные условия не противоречат условиям оператора *IF* во втором операторе цикла *for*. Следовательно, по алгоритму множество W_k будет присоединено к $V^*[X]$, что доказывает соотношение $V^+[X] \subseteq V^*[X]$. □

3. Построение минимального покрытия множества зависимостей включения

В настоящее время отсутствуют СУБД, в которых была бы реализована поддержка обобщенных зависимостей включения. Однако, с учетом реализации обычных зависимостей включения (по одному отношению справа и слева в зависимости) можно предположить, что реализация ссылочной целостности, соответствующей обобщенным зависимостям включения, потребует дополнительной памяти для индексных файлов и дополнительного времени для обслуживания этих файлов. Следовательно, актуальной задачей является минимизация множества Σ, причем, избыточными могут быть сами зависимости и компоненты этих зависимостей.

Допустимые состояния БД останутся прежними после удаления выводимых зависимостей, поскольку их выполнимость будет гарантирована оставшимися зависимостями.

Определение 6. Два множества зависимостей Σ и Σ^* будем считать эквивалентными, если любая зависимость $W[X] \subsetneq V$, выводимая в Σ, также выводима в Σ^*, и любая зависимость $W^*[X] \subsetneq V^*$, выводимая в Σ^*, также выводима в Σ.

Не сложно заметить, что из определения 6 следует очевидное утверждение, которое избавляет от необходимости проверки выводимости любой зависимости.

Утверждение 3. Два множества зависимостей Σ и Σ^* являются эквивалентными, если любая зависимость $W[X] \subsetneq V$, принадлежащая Σ, выводима в Σ^*, и любая зависимость $W^*[X] \subsetneq V^*$, принадлежащая Σ^*, выводима в Σ^*.

Таким образом, для проверки эквивалентности двух множеств зависимостей не требуется проверять выводимость всех возможных зависимостей. Достаточно проверить выводимость собственных зависимостей каждого из множеств в другом множестве.

Пусть задано исходное множество обобщенных типизированных зависимостей включения Σ. На его основе построим множество Σ^*. Для каждой зависимости $W[X] \subsetneq V \in \Sigma$ во множество Σ^* дополним зависимости $R_i[X] \subsetneq V$, $i = \overline{1,m}$, где $W = \cup_{i=1}^{m} R_i$. Из способа построения Σ^* можно сделать следующий вывод.

Утверждение 4. Два множества зависимостей Σ и Σ^* эквивалентны.

Действительно, выводимость зависимостей Σ^* из множества Σ следует по правилу 1 (декомпозиция), а по аксиоме 3 следует выводимость зависимостей Σ из множества Σ^*.

Для построения минимального множества зависимостей будем использовать множество Σ^*, поскольку несколько отношений в левой части зависимостей могут помешать друг другу при проверке выводимости. Например, зависимость $R_i[X] \subsetneq V$ выводима, зависимость $R_j[X] \subsetneq V$ не выводима, тогда зависимость $\{R_i, R_j\}[X] \subsetneq V$ не выводима, что препятствует удалению $R_i[X] \subsetneq V$.

Рассмотрим процедуру построения минимального покрытия Σ^*:

```
procedure MIN_CLOS(Σ*)
modification ← TRUE
while modification
    modification ← FALSE
    for each R_i[X] ⊊ V_i from Σ*
        if R_i ∈ CLOS_M( Σ* − {R_i[X] ⊊ V_i}, V_i, X) then
            Σ* ← Σ* − {R_i[X] ⊊ V_i}
            modification ← TRUE
        end if
    end for
    for each R_i[X] ⊊ V_i  bf from Σ*
        if |V_i| > 1 then
            for each R_j from V_i
                V_j ← V_i − R_j
                Σ⁻ ← Σ* − {R_i[X] ⊊ V_i} ∪ {R_i[X] ⊊ V_j}
                if R_i ∈ CLOS_M( Σ*, V_j, X) then
                    Σ* ← Σ⁻
                    modification ← TRUE
                end if
            end for
        end if
    end for
end while
end procedure
```

С учетом количества итераций в алгоритме *CLOS_M* максимальное количество итераций в алгоритме *MIN_CLOS* будет равно $n^2k + n^2k^2$, где n — мощность множества зависимостей Σ^*, k — количество отношений в БД.

Замечание. Можно предположить, что удаление атрибутов, на которых определены отношения в зависимостях, будет способствовать минимизации множества зависимостей. Однако, всякий раз мы будем иметь большее количество зависимостей, которые все выводимы из исходного множества зависимостей и, следовательно, будут удалены в очередном цикле *while* алгоритма *CLOS_M.*

Необходимость повторного поиска избыточных зависимостей в цикле *while* при анализе Σ^* обусловлена тем, что после удаления избыточных отношений в правых частях зависимостей, возможно, что некоторые из зависимостей окажутся избыточными. Для демонстрации этого свойства рассмотрим пример (аналогичный пример для функциональных зависимостей представил в сети Интернет А. Ю. Филиппович).

Пример 1. Пусть исходное множество зависимостей Σ^* задано для трех отношений: R_1, R_2 и R_3, каждое из которых определено на атрибутах X:

$$\Sigma^* = \{R_3[X] \subsetneq \{R_1, R_2\}, R_2[X] \subsetneq R_1, R_2[X] \subsetneq R_3\}. \quad (1)$$

Проверим на избыточность зависимость $R_3[X] \subsetneq \{R_1, R_2\}$: строим замыкание $\{R_1, R_2\}^+ = R_1, R_2$ на оставшемся множестве зависимостей. В результате имеем, что отношение R_3 не принадлежит замыканию, следовательно, зависимость $R_3[X] \subsetneq \{R_1, R_2\}$ не лишняя.

Проверка на избыточность зависимостей $R_2[X] \subsetneq R_1$ и $R_2[X] \subsetneq R_3$ показала, что эти зависимости не являются избыточными: отношение в левой части не принадлежит замыканию правой части на оставшемся множестве зависимостей. На следующем этапе сначала удалим отношение R_1 из правой части зависимости $R_3[X] \subsetneq \{R_1, R_2\}$. Полученное множество не эквивалентно исходному, поэтому возвращаем R_1 на место. После удаления R_2 получается следующее множество зависимостей:

$$\Sigma^- = \{R_3[X] \subsetneq R_1, R_2[X] \subsetneq R_1, R_2[X] \subsetneq R_3\}.$$

Не сложно показать, что множества Σ^* и Σ^- эквивалентны. Действительно, вторая и третья зависимости в них совпадают. Замыкание правой части зависимости $R_3[X] \subsetneq \{R_1, R_2\}$ на множестве Σ^- содержит R_3 по правилу 2 (пополнение), и замыкание правой части зависимости $R_3[X] \subsetneq R_1$ на множестве Σ^* содержит R_3 по алгоритму построения замыкания. Следовательно, множества Σ^* и Σ^- эквивалентны.

Анализируя множество Σ^- не трудно заметить, что зависимость $R_2[X] \subsetneq R_1$ является избыточной (транзитивной). Эта зависимость будут удалена при *повторном* выполнении цикла *while* в алгоритме *MIN_CLOS*. После удаления избыточной зависимости получим следующее множество:

$$\Sigma^- = \{R_3[X] \subsetneq R_1, R_2[X] \subsetneq R_3\}. \tag{2}$$

На последующих итерациях алгоритма удалить ничего не получится, следовательно, минимальное покрытие Σ^* равно $\{R_3[X] \subsetneq R_1, R_2[X] \subsetneq R_3\}$. Рассмотренный пример показал необходимость повторной проверки на избыточность множества зависимостей после удаления избыточных отношений в правых частях зависимостей.

Дополнительно проверим условия ссылочной целостности полученного результата в соответствии с определением 4. Для исходного множества (1) выполнены следующие условия:

- если кортеж t принадлежит отношению R_1, то ограничения отсутствуют;
- если $t[X] \in R_2[X]$, то $t[X] \in R_1[X]$ и $t[X] \in R_3[X]$;
- если $t[X] \in R_3[X]$, то $t[X] \in R_1[X]$ или $t[X] \in R_2[X]$, в случае если $t[X] \in R_2[X]$, то $t[X] \in R_1[X]$; перепишем последнее условие:
- если $t[X] \in R_3[X]$, то $t[X] \in R_1[X]$, условие $t[X] \in R_3[X]$ может быть не выполнено.

Для результирующего множества (2) выполнены следующие условия:

- если кортеж $t \in R_1$, то ограничения отсутствуют;
- если $t[X] \in R_2[X]$, то $t[X] \in R_1[X]$ и $t[X] \in R_3[X]$;
- если $t[X] \in R_3[X]$, то $t[X] \in R_1[X]$.

Проверка показала, что ссылочные ограничения целостности остались прежними после выполненных преобразований.

После рассмотрения алгоритма *MIN_CLOS* закономерным является вопрос о корректности его работы. Это означает, что выводимая зависимость будет удалена в алгоритме и удаленная зависимость является выводимой.

Теорема 4 (Минимальное покрытие)**.** *Процедура* MIN_CLOS *корректно формирует минимальное покрытие множества обобщенных типизированных зависимостей включения.*

Доказательство. Достаточно просто обосновывается удаление выводимых зависимостей. В соответствии с теоремой 3 зависимость является выводимой на проверочном множестве зависимостей, если ее левая часть принадлежит замыканию правой части зависимости. В алгоритме *MIN_CLOS* циклически просматриваются все возможные варианты выводимости, и выход из бесконечного цикла *while* осуществляется только при отсутствии выводимых зависимостей и избыточных отношений в правых частях зависимостей. Следовательно, результирующее множество Σ^* не содержит выводимых зависимостей.

Также не сложно обосновать, что удаленные зависимости являются выводимыми. Зависимости в первом цикле *for* удаляются, если левая часть зависимости принадлежит замыканию правой части без учета самой зависимости. По теореме 3 такая зависимость является выводимой. Во втором цикле *for* удаляются отношения в правой части зависимости σ^* и проверяется выводимость вновь полученной зависимости σ^- на исходном множестве Σ^*. Если выводимость получена, то в соответствии с определением 6 новое множество зависимостей Σ^- эквивалентно Σ^* (выводимость σ^* на Σ^- гарантируется правилом 2 (пополнение)). Следовательно, исходное множество Σ^* заменяем на множество Σ^-, в котором количество отношений в правой части σ^- меньше. □

Замечание. Различный порядок удаления зависимостей и отношений в алгоритме *MIN_CLOS* может привести к различным минимальным покрытиям, однако, все они эквивалентны друг другу. Если дополнить критерий минимальности для выбора одного из всех покрытий (как это сделано в работе [2] для функциональных зависимостей), то соответствующий алгоритм будет NP-трудным и не будет гарантировать единственность решения.

Рассмотрим пример, в котором есть два минимальных покрытия исходного множества зависимостей.

Пример 2. Исходное множество зависимостей задано для трех отношений, определенных на атрибутах X:

$$\Sigma^* = \{R_3[X] \subsetneq \{R_1, R_2\}, R_2[X] \subsetneq R_1, R_1[X] \subsetneq R_2\}. \tag{3}$$

Различный порядок удаления отношений в первом правиле порождает два минимальных покрытия:

$$\Sigma_1^* = \{R_3[X] \subsetneq R_1, R_2[X] \subsetneq R_1, R_1[X] \subsetneq R_2\} \tag{4}$$

и

$$\Sigma_2^* = \{R_3[X] \subsetneq R_2, R_2[X] \subsetneq R_1, R_1[X] \subsetneq R_2\}. \tag{5}$$

Ссылочные ограничения целостности для множеств зависимостей (3)–(5) совпадают:

- если кортеж $t[X] \in R_1$, то $t[X] \in R_2[X]$;
- если $t[X] \in R_2[X]$, то $t[X] \in R_1[X]$;
- если $t[X] \in R_3[X]$, то $t[X] \in R_1[X]$ и $t[X] \in R_2[X]$.

Проверка показала, что ссылочные ограничения целостности являются эквивалентными для различных минимальных покрытий. Поскольку условие эквивалентности, кроме всего прочего, обладает свойством транзитивности, то все минимальные покрытия будут эквивалентны друг другу (все вместе они эквивалентны исходному множеству зависимостей). Следовательно, всем им соответствует одно и то же множество выполнимых (и выводимых) зависимостей и одни те же ограничения на допустимые состояния БД.

Заключение

В работе рассмотрена проблема выводимости и построения минимального покрытия для обобщенных зависимостей включения, что позволяет избавиться от избыточных затрат по памяти и по времени при их практическом применении. Как отмечалось ранее, в настоящее время пока нет СУБД, реализующих ссылочные ограничения целостности, соответствующие обобщенным типизированным зависимостям включения. Однако, большинство СУБД имеют в своем составе аппарат триггеров, с использованием которого возможна реализация таких ограничений для конкретной БД.

References

[1] J. Ullman, *Principles of Database Systems.* Stanford University: Computer Science Press, 1980, 484 pp.

[2] D. Maier, *The Theory of Relational Databases.* Rockville: Computer Science Press, 1983, 637 pp.

[3] M. Casanova, R. Fagin, and C. Papadimitriou, "Inclusion dependencies and their interaction with functional dependencies", *Journal of Computer and System Sciences*, vol. 28, no. 1, pp. 29–59, 1984.

[4] A. K. Chandra and M. Y. Vardi, "The implication problem for functional and inclusion dependencies is undecidable", *SIAM Journal on Computing*, vol. 14, no. 3, pp. 671–677, 1985. DOI: 10.1137/0214049.

[5] R. Fagin and M. Y. Vardi, "Armstrong databases for functional and inclusion dependencies", *Information Processing Letters*, vol. 16, no. 1, pp. 13–19, 1983. DOI: 10.1016/0020-0190(83)90005-4.

[6] P. M. Kanellakis, R. Cosmadakis, and M. Y. Vardi, "Unary inclusion dependencies have polynomial time inference problems", in *Proceedings of the fifteenth annual ACM symposium on Theory of computing (STOC '83)*, 1983, pp. 264–277. DOI: 10.1145/800061.808756.

[7] S. S. Cosmadakis, P. C. Kanellakis, and M. Y. Vardi, "Polynomial-time implication problems for unary inclusion dependencies", *ACM*, vol. 37, no. 1, pp. 15–46, 1990. DOI: 10.1145/78935.78937.

[8] M. Levene and V. M. W., "Justification for inclusion dependency normal form", *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 2, pp. 281–291, 2000. DOI: 10.1109/69.842267.

[9] M. Hannula and S. Link, "On the interaction of functional and inclusion dependencies with independence atoms", in *Proceedings of the International Conference on Database Systems for Advanced Applications*, 2018, pp. 353–369. DOI: 10.1007/978-3-319-91458-9_21.

[10] J. Biskup and P. Dublish, "Objects in relational database schemes with functional, inclusion and exclusion dependencies", in *3rd Symposium on Mathematical Fundamentals of Database and Knowledge Base Systems*, 1991, pp. 276–290. DOI: 10.1007/3-540-54009-1_20.

[11] F. De Marchi, S. Lopes, and J.-M. Petit, "Efficient algorithms for mining inclusion dependencies", in *Advances in Database Technology — EDBT 2002*, C. S. Jensen *et al.*, Eds., Springer Berlin Heidelberg, 2002, pp. 464–476. DOI: 10.1007/3-540-45876-X_30.

[12] J. Bauckmann, Z. Abedjan, H. Müller, and F. Naumann, "Discovering conditional inclusion dependencies", in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 2094–2098. DOI: 10.1145/2396761.2398580.

[13] M. T. Gomez-Lopez, R. M. Gasca, and J. M. Perez-Alvarez, "Compliance validation and diagnosis of business data constraints in business processes", *Information Systems*, vol. 48, pp. 26–43, 2015.

[14] S. Ma, W. Fan, and L. Bravo, "Extending inclusion dependencies with conditions", *Theoretical Computer Science*, vol. 515, pp. 64–95, 2014. DOI: 10.1016/j.tcs.2013.11.002.

[15] F. Tschirschnitz, T. Papenbrock, and F. Naumann, "Detecting inclusion dependencies on very many tables", *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 1–29, 2017. DOI: 10.1145/3105959.

[16] Y. Kaminsky, E. Pena, and F. Naumann, "Discovering similarity inclusion dependencies", *Proceedings of the ACM on Management of Data*, vol. 1, no. 1, pp. 1–24, 2023. DOI: 10.1145/3588929.

[17] M. Levene and G. Loizou, "Null inclusion dependencies in relational databases", *Information and Computation*, vol. 136, no. 2, pp. 67–108, 1997. DOI: 10.1006/inco.1997.2631.

[18] M. Levene and G. Loizou, "The additivity problem for data dependencies in incomplete relational databases", in *Semantics in Databases*, vol. 1358, Springer, 1998, pp. 136–169. DOI: 10.1007/BFb0035008.

[19] H. Köhler and S. Link, "Inclusion dependencies reloaded", in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1361–1370.

[20] V. S. Zykin and S. V. Zykin, "Analysis of typed inclusion dependences with null values", *Automatic Control and Computer Sciences*, vol. 52, no. 7, pp. 638–646, 2018. DOI: 10.3103/S0146411618070258.

[21] S. V. Zykin, "Generalization of typed include dependencies with null values in databases", *Modeling and Analysis of Information Systems*, vol. 30, no. 3, pp. 192–201, 2023, in Russian.