



**Параллельное и распределенное программирование, грид-технологии, программирование на графических процессорах**

<https://doi.org/10.25559/SITITO.019.202304.893-903>  
УДК 004.0272.26

## **Аксиоматический метод доказательства соответствия формализованных в различных моделях данных запросов**

**В. И. Мунерман\*, Д. В. Мунерман**

Оригинальная статья

ФГБОУ ВО «Смоленский государственный университет», г. Смоленск,  
Российская Федерация

Адрес: 214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4

\* vimoona@gmail.com

### **Аннотация**

Аксиоматический метод в программировании используется давно и с целью более формального и точного описания процессов. Основное направление аксиоматизации в программировании было направлено на доказательство частичной правильности (верификацию) программ. Вместе с тем, ряд работ был ориентирован на обработку больших потоков данных – файлов. В них рассматриваются алгоритмы как унарных, так и бинарных операций и на основе аксиом доказывается их частичная правильность. Кроме того, аксиоматический подход позволил решить ряд задач в области разработки программного обеспечения для различных системных и прикладных областей. В статье рассматривается аксиоматический метод, который позволяет формализовать объекты обработки данных и операции преобразования объектов посредством специального описания исходных объектов, и объекта – результата преобразования. В статье рассматривается возможность преобразования на основе аксиоматики для массовой обработки данных алгебраических выражений запросов, формализованных в одной модели данных, в частности в многомерно-матричной в другую – реляционную модель данных. Необходимость такого преобразования состоит в том, что выражение в многомерно-матричной модели значительно проще, чем в реляционной. Его проще преобразовывать с целью оптимизации запроса, и оно позволяет решить проблему распараллеливания запроса как на уровне задачи, так и на уровне отдельных операций. Для этой цели даны теоретико-множественные алгебраические определения двух основных операций обработки данных, соответствующих слиянию строго и нестрого упорядоченных файлов. На основе этих определений приведены аксиоматические описания этих операций. Приведены определения операций сложения и умножения многомерных матриц. Сконструированы соответствующие им операции-запросы в реляционной модели данных. Это соответствие доказано на основе аксиоматического метода. Показано, что каждый запрос на обработку данных есть теорема, и сделан вывод об эквивалентности моделей как аксиоматических теорий. Приведен пример, который показывает, как устанавливается такое соответствие.

**Ключевые слова:** аксиоматический метод, многомерно-матричная модель, реляционная модель, соответствие моделей данных

**Конфликт интересов:** авторы заявляют об отсутствии конфликта интересов.

**Для цитирования:** Мунерман В. И., Мунерман Д. В. Аксиоматический метод доказательства соответствия формализованных в различных моделях данных запросов // Современные информационные технологии и ИТ-образование. 2023. Т. 19, № 4. С. 893-903. <https://doi.org/10.25559/SITITO.019.202304.893-903>

© Мунерман В. И., Мунерман Д. В., 2023



Контент доступен под лицензией Creative Commons Attribution 4.0 License.  
The content is available under Creative Commons Attribution 4.0 License.



**Parallel and Distributed Programming, Grid Technologies, GPU Programming**

## **Axiomatic Method for Proof of Conformity of Queruies Formalized in Diferent Data Models**

**V. I. Munerman<sup>\*</sup>, D. V. Munerman**

Original article

Smolensk State University, Smolensk, Russian Federation

Address: 4 Przhivalsky St., Smolensk 214000, Russian Federation

<sup>\*</sup> vimoon@gmail.com

### **Abstract**

The axiomatic method in programming has been used for a long time and for the purpose of a more formal and accurate description of processes. The main direction of axiomatization in programming was aimed at proving the partial correctness (verification) of programs. At the same time, a number of works were focused on processing large data streams – files. They consider algorithms for both unary and binary operations and, based on axioms, prove their partial correctness. In addition, the axiomatic approach made it possible to solve a number of problems in the field of software development for various system and application areas. The article discusses an axiomatic method that allows you to formalize data processing objects and object transformation operations through a special description of the source objects and the object – the result of the transformation. The article discusses the possibility of converting, on the basis of axiomatic for mass data processing, algebraic query expressions formalized in one data model, in particular in a multidimensional matrix, into another – a relational data model. The need for such a transformation is that the expression in the multidimensional matrix model is much simpler than in the relational one. It is easier to transform to optimize a query, and it solves the problem of query parallelization both at the task level and at the level of individual operations. For this purpose, set-theoretic algebraic definitions of two basic data processing operations corresponding to the merging of strictly and loosely ordered files are given. Based on these definitions, axiomatic descriptions of these operations are given. Definitions of the operations of addition and multiplication of multidimensional matrices are given. The corresponding query operations in the relational data model were constructed. This correspondence is proven based on the axiomatic method. It is shown that each request for data processing is a theorem, and a conclusion is made about the equivalence of models as axiomatic theories. An example is given that shows how such a correspondence is established.

**Keywords:** axiomatic method, multidimensional matrix model, relational model, data model accordance

**Conflict of interests:** The authors declare no conflict of interest.

**For citation:** Munerman V.I., Munerman D.V. Axiomatic Method for Proof of Conformity of Queruies Formalized in Diferent Data Models. *Modern Information Technologies and IT-Education*. 2023;19(4):893-903. <https://doi.org/10.25559/SITITO.019.202304.893-903>



## Введение

Аксиоматический метод в программировании используется давно и с целью более формального и точного описания процессов [1-4]. Основное направление аксиоматизации в программировании было направлено на доказательство частичной правильности (верификацию) программ [5-10]. Вместе с тем, ряд работ [11-13] был ориентирован на обработку больших потоков данных – файлов. В них рассматриваются алгоритмы как унарных, так и бинарных операций и на основе аксиом доказывается их частичная правильность. Кроме того, аксиоматический подход позволил решить ряд задач в области разработки программного обеспечения для различных системных и прикладных областей [14-18]. Аксиоматический метод неразрывно связан с алгебраическим методом.

Алгебраический метод позволяет конструктивно описывать структуры объектов и алгоритмы операций. Структура объектов определяется либо строго, с использованием математической терминологии, либо по аналогии с известными структурами, например, в файловой модели файл определяется на языке теории множеств, а в реляционном отношении подобный объект имеет представляется в виде таблицы. Алгоритмы также определяются либо посредством описания на естественном языке, либо с использованием инструментов формализации, присущих теоретическому и практическому программированию. То есть алгебраическому методу присущ принцип: «после применения операции в соответствии с известным алгоритмом к объекту (объектам) с известной структурой получается объект-результат с заданной структурой».

Аксиоматический метод используется для формализации операции преобразования объектов путем описания объекта-результата преобразования исходных объектов. То есть он предоставляет описание объектов и операций их преобразования в соответствии с принципом: «что должно быть результатом операций над объектами-операндами, независимо от того, какова реальность этих объектов и как эта операция выполняется» [19].

Следуя этим принципам, на основе алгебры преобразователей предикатов [20] были построены аксиомы, которые позволили проводить формальные доказательства частичной правильности программ.

В статье рассматривается возможность преобразования на основе аксиоматики для массовой обработки данных [19], [21, 22] алгебраических выражений запросов, формализованных в одной модели данных, в частности в многомерно-матричной<sup>1</sup> [23] в другую – реляционную модель данных. Необходимость такого преобразования состоит в том, что выражение в многомерно-матричной модели значительно проще, чем в реляционной. Его проще преобразовывать с целью оптимизации запроса,

и оно позволяет решить проблему распараллеливания запроса, как на уровне задачи, так и на уровне отдельных операций.

## Основные понятия

Здесь рассматриваются определения, аксиомы и методы, обеспечивающие преобразование алгебраических выражений запроса в массовой обработке данных из одной модели данных в другую.

Определения операций массовой обработки данных даны в терминах теоретико-множественной (файловой) модели данных [19], которая служит связующим звеном между алгеброй многомерных матриц и реляционной алгеброй.

В этой модели файл определяется следующим образом. Пусть даны множество однотипных записей  $X$  и множество ключей  $K$ , которые идентифицируют эти записи. Файлом  $X_K$  называется фактор-множество множества однотипных записей  $X$  по отношению эквивалентности, порожденному множеством  $K$ . Составляющие файл классы эквивалентности обозначаются  $X_{K^*}$ , или  $X_{K_{(1)}^*}, X_{K_{(2)}^*}, \dots$ . Полезно

рассматривать экземпляры множества ключей, которым в реальном множестве  $X$  не соответствует ни одной записи. В этом случае, при определении операций над файлами, целесообразно считать, что таким экземплярам множества ключей соответствует универсальная неопределенная запись  $\Theta$ . Класс эквивалентности, соответствующий экземпляру множества ключей  $K^*$  и состоящий из единственной записи  $\Theta$ , будет обозначаться  $\Theta_{K^*}$ .

Если каждый класс эквивалентности файла  $X_K$  содержит единственную запись, то файл  $X_K$  называется строго упорядоченным, если же в каждом классе эквивалентности может быть более одной записи – нестрого упорядоченным.

Далее рассматриваются две бинарные операции: слияние строго упорядоченных файлов, слияние нестрого упорядоченных файлов, и вспомогательная унарная операция – сжатие, необходимость которой обусловлено тем, что в реляционной алгебре нет общей операции слияния строго упорядоченных файлов и ее приходится реализовать как композицию двух операций – объединения и сжатия.

**Слияние строго упорядоченных файлов ( $ms$ ).** Пусть даны два файла  $X_K$  и  $Y_K$  строго упорядоченные по одному и тому же множеству ключей  $K$ . В результате слияния этих строго упорядоченных файлов образуется файл  $Z_K$ , классы эквивалентности задаются соотношением  $Z_{K^*} = f(X_{K^*}, Y_{K^*})$ . Функция  $f(X_{K^*}, Y_{K^*})$ , определенная на классах эквивалентности исходных файлов, задает характер операции.

В сложных вычислительных задачах построение функции  $f$  может быть следующим:

$$f(X_{K^*}, Y_{K^*}) = \begin{cases} g_1(Y_{K^*}), & \text{если } X_{K^*} = \Theta_{K^*} \\ g_2(X_{K^*}), & \text{если } Y_{K^*} = \Theta_{K^*} \\ g_3(X_{K^*}, Y_{K^*}), & \text{если } X_{K^*} \neq \Theta_{K^*}, Y_{K^*} \neq \Theta_{K^*} \end{cases}$$

<sup>1</sup> Соколов Н. П. Введение в теорию многомерных матриц. Киев: Наукова думка, 1972. 176 с.



### Слияние нестрого упорядоченных файлов (*mns*).

Пусть  $X_L$  и  $Y_M$  – файлы, упорядоченные (возможно строго) по множествам ключей  $L$  и  $M$ , причем выполняется условие  $L \cap M \neq \emptyset$ , и пусть  $K$  – множество ключей, связанное с множествами  $L$  и  $M$  соотношениями:

1.  $K \subseteq L \cup M$ ,
2.  $L \cap K \neq \emptyset$  и  $M \cap K \neq \emptyset$ .

Это означает, что множество ключей  $K$  состоит из ключей, входящих в множества  $L$  и  $M$ , причем в  $K$  содержится, по крайней мере, по одному ключу из каждого множества. Тогда, по крайней мере, один файл  $X_{K \cap L}$  или  $Y_{K \cap M}$  нестрого упорядочен по множеству ключей. Если  $M \not\subseteq L$  и  $L \not\subseteq M$ , то файлы  $X_{K \cap L}$  и  $Y_{K \cap M}$  нестрого упорядочены. Слияние файлов производится по множеству ключей  $K$ . Пусть  $K^*$  – фиксированный экземпляр множества ключей  $K$ , а  $(K \cap L)^*$  и  $(K \cap M)^*$  – такие фиксированные экземпляры множеств ключей  $(K \cap L)$  и  $(K \cap M)$ , что значения одноименных ключей в них совпадают. Тогда можно задать вычисление класса эквивалентности файла  $Z_K$  по следующему правилу:

$$Z_{K^*} = \begin{cases} \Theta_{K^*}, & \text{если } X_{(K \cap L)^*} = \Theta_{(K \cap L)^*}, \text{ или } Y_{(K \cap M)^*} = \Theta_{(K \cap M)^*}, \\ f(X_{(K \cap L)^*}, Y_{(K \cap M)^*}), & \text{в противном случае.} \end{cases}$$

Функция  $f(X_{(K \cap L)^*}, Y_{(K \cap M)^*})$  определена на классах эквивалентности  $X_{(K \cap L)^*}$  и  $Y_{(K \cap M)^*}$ , а ее значение – класс эквивалентности  $Z_{K^*}$ , состоящий из элементов, каждый из которых вычисляется из пары элементов, принадлежащей декартову произведению  $X_{(K \cap L)^*} \times Y_{(K \cap M)^*}$ .

Если функция  $f(X_{(K \cap L)^*}, Y_{(K \cap M)^*})$  реализует групповую операцию, то операция слияния нестрого упорядоченных файлов включает в себя и операцию сжатия файла  $Z_K$ , в результате которой получается файл  $Z_{K'}$  (множество ключей  $K'$  есть подмножество множества ключей  $K$ , то есть  $K' \subseteq K$ ).

Сжатие (*quant*). Это вспомогательная операция, которая позволяет при необходимости установить соответствие с действиями типа свертки и группировки в алгебре многомерных матриц и реляционной алгебре. Пусть даны файлы  $X_K$ , нестрого упорядоченный по множеству ключей  $K$ , и  $Y_K$ , строго упорядоченный по множеству ключей  $K$ . Классы эквивалентности этих файлов связаны соотношением  $Y_{K^*} = f(X_{K^*})$ , где  $f$  – функция, реализующая групповую операцию (операцию квантификации). Тогда считается, что файл  $Y_K$  получен из файла  $X_K$  в результате применения операции сжатия. В операциях сжатия каждому непустому классу эквивалентности  $X_{K^*}$  исходного файла, нестрого упорядоченного по множеству ключей  $K$ , ставится в соответствие класс  $Y_{K^*}$  выходного файла  $Y_K$ , строго упорядоченного по тому же множеству ключей. При этом единственная запись, принадлежащая  $Y_{K^*}$ , вычисляется как значение функции  $f$ , определенной на всех записях класса эквивалентности  $X_{K^*}$ . В качестве такой функции могут быть использованы, например,

такие агрегатные функции как вычисление суммы или среднего значения, поиск наименьшего или наибольшего значения и тому подобные.

Эти операция имеют следующие аксиоматические описания [19, 21, 22].

### Аксиома слияния при строгой упорядоченности агрегатов данных (A1)

$$^{so}(\mathcal{R}(A_1, \dots, A_k)R_1(A_1, \dots, A_k, B_{k+1}, \dots, B_m)) \oplus ^{so}(\mathcal{R}(A_1, \dots, A_k)R_2(A_1, \dots, A_k, C_{k+1}, \dots, C_n)) = \{(a_1, \dots, a_k, f(b_{k+1}, \dots, b_m, c_{k+1}, \dots, c_n))\}$$

Эта аксиома дает описание агрегата данных, который должен быть получен из агрегатов данных  $R_1(A_1, \dots, A_k, B_{k+1}, \dots, B_m)$  и  $R_2(A_1, \dots, A_k, C_{k+1}, \dots, C_n)$  после применения операции  $f(b_{k+1}, \dots, b_m, c_{k+1}, \dots, c_n)$  к их элементам с одинаковыми значениями  $a_1, \dots, a_k$ . Новый агрегат данных формируется в результате применения операции  $(\oplus)$  к этим агрегатам данных. В файловой модели данных она соответствует операции слияния строго упорядоченных файлов, в многомерно-матричной – матричное сложение, в реляционной – любая из теоретико-множественных операций над отношениями, между которыми установлено соответствие один-к-одному, при необходимости с дополнительной группировкой.

### Аксиома слияния при нестрогой упорядоченности агрегатов данных (A2)

$$^{nso}(\mathcal{R}(A_1, \dots, A_k, B_{k+1}, \dots, B_m)R_1(A_1, \dots, A_k, B_{k+1}, \dots, B_m, C_{m+1}, \dots, C_p)) \otimes ^{nso}(\mathcal{R}(B_1, \dots, B_k, D_{k+1}, \dots, D_n)R_2(B_1, \dots, B_k, D_{k+1}, \dots, D_n, E_{n+1}, \dots, E_q)) =$$

$$\{(a_{k+1}, \dots, a_m, d_{k+1}, \dots, d_n), f_{(b_1, \dots, b_k)}(C'_{m+1}, \dots, C'_p, E'_{n+1}, \dots, E'_q)\} | C'_{m+1} \subseteq C_{m+1}, \dots, C'_p \subseteq C_p, E'_{n+1} \subseteq E_{n+1}, \dots, E'_q \subseteq E_q\}$$

Эта аксиома дает описание агрегата данных, который должен быть получен из агрегатов данных  $R_1(A_1, \dots, A_k, B_{k+1}, \dots, B_m, C_{m+1}, \dots, C_p)$  и  $R_2(A_1, \dots, A_k, D_{k+1}, \dots, D_n, E_{n+1}, \dots, E_q)$  после применения групповой операции  $f_{b_1, \dots, b_k}(C'_{m+1}, \dots, C'_p, E'_{n+1}, \dots, E'_q)$  к группам их элементов по всем возможным значениям кортежей  $(b_1, \dots, b_k)$ . Новый агрегат данных формируется в результате применения операции  $(\otimes)$  к этим агрегатам данных. В файловой модели данных она соответствует операции слияния нестрого упорядоченных файлов, в многомерно-матричной ( $\lambda$ ,  $\mu$ )-свернутому произведению, в реляционной – операции соединения (`SELECT... FROM R1 JOIN R2 ON...`) отношений, между которыми установлены отношения один-ко-многим или многие-ко-многим.

Возможен случай, когда один из агрегатов данных строго упорядочен. Тогда эта аксиома (при строгой упорядоченности агрегата данных  $R_1$ ) имеет вид

$$^{so}(\mathcal{R}(A_1, \dots, A_k)R_1(A_1, \dots, A_k, C_{m+1}, \dots, C_p)) \otimes$$

$$^{nso}(\mathcal{R}(A_1, \dots, A_k)R_2(A_1, \dots, A_k, D_{k+1}, \dots, D_n, E_{n+1}, \dots, E_q)) =$$

$$\{(a_{k+1}, \dots, a_m, f_{(d_1, \dots, d_n)}(C'_{m+1}, \dots, C'_p, E'_{n+1}, \dots, E'_q)) | C'_{m+1} \subseteq C_{m+1}, \dots, C'_p \subseteq C_p, E'_{n+1} \subseteq E_{n+1}, \dots, E'_q \subseteq E_q\}$$

### Операции в многомерно-матричной и реляционной моделях

Здесь рассматриваются операции, соответствующие операциям файловой модели, которые определены в предыдущем разделе.

Пусть  $i_1, \dots, i_p$  – совокупность индексов, принимающих





значения от 1 до  $n_\alpha (\alpha = 1, \dots, p)$  соответственно. Тогда  $p$ -мерная матрица – это совокупность  $A = \{a_{i_1 \dots i_p}\}$  элементов некоторого типа, над которым определены аддитивная и мультипликативная операции. Стандартное обозначение многомерной матрицы  $A = \|a_{i_1 \dots i_p}\|$ . Далее определяются необходимые операции над многомерными матрицами.

**Сложение.** Суммой двух  $p$ -мерных матриц  $A = \|a_{i_1 \dots i_p}\|$  и  $B = \|b_{i_1 \dots i_p}\|$  с одинаковыми наборами индексов  $i_1, \dots, i_p$  называется  $p$ -мерная матрица  $C = \|c_{i_1 \dots i_p}\|$  с тем же набором индексов, элементы которой вычисляются по формуле  $c_{i_1 \dots i_p} = a_{i_1 \dots i_p} + b_{i_1 \dots i_p}$ .

**Умножение.** Пусть матрицы  $A = \|a_{i_1 \dots i_p}\|$  и  $B = \|b_{i_1 \dots i_q}\|$ ,  $p$  и  $q$ -мерные соответственно. Совокупности индексов этих матриц  $i_1, \dots, i_p$  и  $i_1, \dots, i_q$  разбиваются на четыре группы, содержащие соответственно  $\kappa$ ,  $\lambda$ ,  $\mu$  и  $\nu$  индексов ( $\kappa, \lambda, \mu, \nu \geq 0$ ). Причем  $\kappa + \lambda + \mu = p$ , а  $\lambda + \mu + \nu = q$ . Для полученных групп индексов используются обозначения:  $l = (l_1, \dots, l_\kappa)$ ,  $s = (s_1, \dots, s_\lambda)$ ,  $c = (c_1, \dots, c_\mu)$  и  $m = (m_1, \dots, m_\nu)$ . Тогда матрицы  $A$  и  $B$  можно представить в виде  $A = \|a_{lsc}\|$  и  $B = \|b_{scm}\|$ . Индексы групп  $s$  и  $c$  в матрицах  $A$  и  $B$  полностью совпадают. Так же как в операции свертки, индексы разбиения  $c$  называются кэлиевыми. Индексы разбиения  $s$  называются скоттовыми, а индексы разбиения  $m$ , так же, как и индексы разбиения  $l$ , – свободными.

Матрица  $C = \|c_{lsm}\|$ , элементы которой вычисляются по формуле  $c_{lsm} = \sum_{(c)} a_{lsc} \times b_{scm}$ , называется  $(\lambda, \mu)$ -свернутым произведением и обозначается  ${}^{\lambda, \mu}(A \times B)$ .

В реляционной алгебре нет операции, которая соответствует операции слияния строго упорядоченных файлов. Операция UNION не может точно реализовать подобную операцию, поскольку если в двух отношениях-операндах встретятся два полностью совпадающих кортежа, то в отношение-результат будет помещен только один из этих кортежей, а не результат их совместной обработки. Поэтому в большинстве современных систем управления базами данных имеется операция UNION ALL. Эта операция нарушает теоретико-множественный принцип – «отношение есть множество», а ее результат – «мультимножество», которое может содержать несколько полностью совпадающих элементов. Используя эту операцию можно сконструировать аналог операции слияния строго упорядоченных файлов, которая будет названа *слияние «строго упорядоченных» таблиц*. Термин «строго упорядоченных» взят в кавычки, поскольку в реляционной теории баз данных сортировка не рассматривается как самостоятельная операция. Заметим, что в технологии систем управления базами данных операция сортировки явно используется в

алгоритмах, реализующих основные операции. Термин «таблица» в дальнейшем будет использоваться как синоним термина «отношение», как принято в современных базах данных.

**Слияние «строго упорядоченных» таблиц.** Пусть  $A(K_1, \dots, K_p, W)$  и  $B(K_1, \dots, K_p, W)$  – таблицы, схемы которых совпадают.  $K_1, \dots, K_p$  – составной ключ, по которому эти таблицы находятся в третьей нормальной форме.  $W$  – неключевые атрибуты, которые могут быть однотипными кортежами, над которыми определена бинарная операция  $\omega$ . Также определена групповая (агрегатная) операция  $\Omega$  над группами значений  $W_A \omega W_B = \Omega(W_A \omega W_B)$ . Тогда запрос 

```
SELECT UnionAll.K1, ..., UnionAll.Kp, SUM(UnionAll.W)
FROM (SELECT * FROM A UNION ALL SELECT * FROM B) AS
UnionAll
GROUP BY UnionAll.K1, ..., UnionAll.Kp
```

 реализует операцию слияние «строго упорядоченных» таблиц.

**Inner Join.** Здесь рассматривается наиболее распространенный и самый вычислительно сложный вариант этой операции. Пусть даны таблицы  $A(K_1, \dots, K_p, W)$  и  $B(K_1, \dots, K_q, W)$ , схемы которых не совпадают.  $K_1, \dots, K_p$  и  $K_1, \dots, K_q$  – составные ключи, по которым эти таблицы находятся в третьей нормальной форме.  $W$  – неключевые атрибуты, которые могут быть не обязательно однотипными кортежами, над которыми определена бинарная операция  $\omega$ . Также определена групповая (агрегатная) операция  $\Omega$  над группами значений  $W_A \omega W_B = \Omega(W_A \omega W_B)$ . Кроме того, часть ключей обеих таблиц совпадает. Без ограничения общности можно считать, что это ключи  $K_1, \dots, K_s$  ( $s < p, s < q$ ). Тогда одному и тому же набору значений ключей  $K_1, \dots, K_s$  может соответствовать несколько строк таблицы, то есть они соответствуют нестрого упорядоченным файлам. Не нарушая общности можно предположить, что ключи, используемые в предикате ON имеют в обеих таблицах индексы  $1, \dots, s$ . Если часть остальных ключей таблицы  $A$  совпадают с ключами таблицы  $B$ , то в список полей оператора SELECT они включаются только один раз.

Следующий запрос демонстрирует общий вид этой операции.

```
SELECT A.Ks+1, ..., A.Kp, B.Ks+1, ..., B.Kq, \Omega(A.W \omega B.W)
FROM A INNER JOIN B ON A.K1, ..., A.Ks = B.K1, ..., B.Ks
GROUP BY A.Ks+1, ..., A.Kp, B.Ks+1, ..., B.Kq
```

## Соответствие операций в многомерно-матричной и реляционной моделях

Для доказательства соответствия операций необходимо представить формализованное в модели алгебраическое выражение в виде, который формализован в виде схемы аксиомы теории массовой обработки данных. Если полученные формулы совпадают с точностью до обозначений термов и констант, то эти операции соответствуют.

**Сложение многомерных матриц и слияние «строго упорядоченных» таблиц.** Операцию сложения многомерных матриц можно представить в виде формулы  $\|a_{i_1 \dots i_p}\| + \|b_{i_1 \dots i_p}\| = \|c_{i_1 \dots i_p}\|$ . (1)



Поскольку в каждой из матриц каждому набору значений индексов соответствует единственный элемент, и число элементов во всех матрицах одинаково, то при подстановке знаков формулы (1) в аксиому A1 получается формула

$$^{so}(\mathcal{H}(i_1, \dots, i_p)A(i_1, \dots, i_p, a)) \oplus ^{so}(\mathcal{H}(i_1, \dots, i_p)B(i_1, \dots, i_p, b)) = \{(i_1^*, \dots, i_p^*, a_{i_1, \dots, i_p}^* + b_{i_1, \dots, i_p}^*)\} \quad (2)$$

Здесь  $a$  и  $b$  – элементы матриц  $A$  и  $B$ ,  $i_1^*, \dots, i_p^*$  – фиксированный набор значений индексов

При слиянии «строго упорядоченных» таблиц считается, что они логически содержат строки со всеми возможными значениями ключей, то есть, часть строк может быть универсальной неопределенной строкой (записью)  $\Theta$ . Тогда приведенному SQL-запросу соответствует формула

$$^{so}(\mathcal{H}(K_1, \dots, K_p)A(K_1, \dots, K_p, w_a)) \oplus ^{so}(\mathcal{H}(K_1, \dots, K_p)B(K_1, \dots, K_p, w_b)) = \{(K_1^*, \dots, K_p^*, w_a + w_b)\} \quad (3)$$

Формулы (2) и (3) совпадают с точностью до обозначений. Следовательно, операция сложения многомерных матриц и слияния «строго упорядоченных» таблиц соответствуют друг другу.

**Умножение многомерных матриц и операция Join.**

В операции  $(\lambda, \mu)$ -свернутого произведения индексы матрицы  $A$  представляют собой последовательность вида  $l_1, \dots, l_k, s_1, \dots, s_\lambda, c_1, \dots, c_\mu$ , матрицы  $B$  –

$s_1, \dots, s_\lambda, c_1, \dots, c_\mu, m_1, \dots, m_\nu$ , а матрицы-результата  $C$  –  $l_1, \dots, l_k, s_1, \dots, s_\lambda, m_1, \dots, m_\nu$ . Тогда эту операцию можно представить формулой

$$\lambda, \mu \left( \left\| a_{l_1, \dots, l_k, s_1, \dots, s_\lambda, c_1, \dots, c_\mu} \right\| \times \left\| b_{s_1, \dots, s_\lambda, c_1, \dots, c_\mu, m_1, \dots, m_\nu} \right\| \right) = \left\| c_{l_1, \dots, l_k, s_1, \dots, s_\lambda, m_1, \dots, m_\nu} \right\| = \sum_{(c_1, \dots, c_\mu)} a_{l_1, \dots, l_k, s_1, \dots, s_\lambda, c_1, \dots, c_\mu} \times b_{s_1, \dots, s_\lambda, c_1, \dots, c_\mu, m_1, \dots, m_\nu} \quad (4)$$

После подстановки из формулы 4 в аксиому A2, аналогичной той, что была проделана в предыдущем разделе, она примет вид:

$$^{nso}(\mathcal{H}(l_1, \dots, l_k, s_1, \dots, s_\lambda, c_1, \dots, c_\mu)A(l_1, \dots, l_k, s_1, \dots, s_\lambda, c_1, \dots, c_\mu, a)) \otimes ^{nso}(\mathcal{H}(s_1, \dots, s_\lambda, c_1, \dots, c_\mu, m_1, \dots, m_\nu)B(s_1, \dots, s_\lambda, c_1, \dots, c_\mu, m_1, \dots, m_\nu, b)) = \{(l_1^*, \dots, l_k^*, s_1^*, \dots, s_\lambda^*, c_1^*, \dots, c_\mu^*, m_1^*, \dots, m_\nu^*), \sum_{(c_1, \dots, c_\mu)} a_{l_1, \dots, l_k, s_1, \dots, s_\lambda, c_1, \dots, c_\mu} \times b_{s_1, \dots, s_\lambda, c_1, \dots, c_\mu, m_1, \dots, m_\nu}\} \quad (5)$$

Здесь полиморфные символы операций  $\Sigma$  и  $\times$  означают групповую аддитивную и бинарную мультипликативную операции.

Пусть схемы таблиц  $A(A_1, \dots, A_u)$  и  $B(B_1, \dots, B_v)$  могут быть представлены в виде  $A(L_1, \dots, L_k, S_1, \dots, S_\lambda, C_1, \dots, C_\mu, W)$  и  $B(S_1, \dots, S_\lambda, C_1, \dots, C_\mu, M_1, \dots, M_\nu, W)$ . Здесь атрибуты (столбцы таблицы)  $S_1, \dots, S_\lambda$  и  $C_1, \dots, C_\mu$  – общие для таблиц  $A$  и  $B$ , и атрибуты  $L_1, \dots, L_k, S_1, \dots, S_\lambda, M_1, \dots, M_\nu$  – ключи-кандидаты таблицы-результата запроса, представленного формулой

SELECT A.  $L_1, \dots, A.L_k, A.S_1, \dots, A.S_\lambda, B.M_1, \dots, B.M_\nu, \Omega(A.W \otimes B.W)$   
FROM A INNER JOIN B ON A. $C_1, \dots, A.C_\mu = B.C_1, \dots, B.C_\mu$   
GROUP BY A.  $L_1, \dots, A.L_k, A.S_1, \dots, A.S_\lambda, B.M_1, \dots, B.M_\nu$  (6)

После подстановки из формулы запроса 6 в аксиому A2 знаков запроса из формулы 6, получается формула

$$^{nso}(\mathcal{H}(L_1, \dots, L_k, S_1, \dots, S_\lambda, C_1, \dots, C_\mu)A(L_1, \dots, L_k, S_1, \dots, S_\lambda, C_1, \dots, C_\mu, W)) \otimes ^{nso}(\mathcal{H}(S_1, \dots, S_\lambda, C_1, \dots, C_\mu, M_1, \dots, M_\nu)B(S_1, \dots, S_\lambda, C_1, \dots, C_\mu, M_1, \dots, M_\nu, W)) = \{(L_1^*, \dots, L_k^*, S_1^*, \dots, S_\lambda^*, C_1^*, \dots, C_\mu^*, M_1^*, \dots, M_\nu^*), \Omega(A.W \otimes B.W)\} \quad (7)$$

Очевидно, что, как и в случае аксиомы A1, формулы 5 и 7 эквивалентны.

## Соответствие выражений в алгебре многомерных матриц выражениям запросов в реляционной алгебре

Каждый запрос состоит из последовательности операций и задан формулой (алгебраическим выражением) в одной из моделей. Но в аксиоматических теориях, каждая формула, выводимая из аксиом, есть теорема. В работе<sup>2</sup> доказано следующее утверждение.

**Теорема об эквивалентности теорий (Н. Бурбаки)**

Пусть  $\mathcal{T}$  – теория,  $\mathcal{A}_1, \dots, \mathcal{A}_n$  – ее явные аксиомы,  $a_1, \dots, a_h$  – ее константы,  $T_1, \dots, T_h$  – ее термы. Если  $(T_1|a_1) \dots (T_h|a_h) \mathcal{A}_i$ , ( $i=1, \dots, n$ ) являются теоремами теории  $\mathcal{T}'$ , и знаки теории  $\mathcal{T}$  являются знаками теории  $\mathcal{T}'$ , и схемы теории  $\mathcal{T}$  являются схемами теории  $\mathcal{T}'$ , то если  $\mathcal{A}$  теорема теории  $\mathcal{T}$ , то  $(T_1|a_1) \dots (T_h|a_h) \mathcal{A}$  есть теорема теории  $\mathcal{T}'$ .

Здесь  $(T_1|a_1) \dots (T_h|a_h)$  – подстановки термов  $T_1, \dots, T_h$  вместо констант  $a_1, \dots, a_h$  в формулах  $\mathcal{A}_i$ , ( $i=1, \dots, n$ ) и  $\mathcal{A}$ . Из доказательства соответствия операций и этой теоремы следует, что многомерно-матричная и реляционная модели есть эквивалентные аксиоматические теории. В алгебраическом смысле они гомоморфные универсальные алгебраические системы. Интуитивно это означает, что если:

– аксиомы теории  $\mathcal{T}$  выражают свойства объектов  $a_1, \dots, a_h$ ,

– формула  $\mathcal{A}$  выражает свойство, вытекающее из этих аксиом,

– объекты  $T_1, \dots, T_h$  в теории  $\mathcal{T}$  обладают свойствами, выраженными аксиомами теории  $\mathcal{T}$ , тогда они имеют свойство, выражаемое формулой  $\mathcal{A}$ . То есть результаты теории  $\mathcal{T}$  применяются в теории  $\mathcal{T}'$ .

Главное для рассматриваемых в статье целей следствие из этой теоремы состоит в том, что алгебраическому выражению в одной модели может соответствовать выражение в другой модели. Далее приводится пример, иллюстрирующий установление таких соответствий.

Пусть дана реляционная база данных, схемы и содержание таблиц которых приведены в таблице 1.

Значение поля  $S$  для каждого значения поля  $K1$  вычисляется по формуле

$$S = AW \times \sum_{(K2)} BW \times (CW + \sum_{(K4)} DW \times EW). \quad \text{В листинге 1}$$

приведен запрос, реализующий вычисление таблицы Res из таблиц  $A, B, C, D, E$ .

```
1. SELECT A.K1, SUM(A.W*Q4.W) FROM A Inner Join
2.   (SELECT B.K1, B.K2, SUM(B.W*Q3.W) AS W FROM
   B Inner Join
3.   (SELECT UA.K2, UA.K3, SUM(UA.W) AS W FROM
4.   SELECT C.K2, C.K3, C.W FROM C Union All
   SELECT Q2.* FROM
5.   (SELECT Q1.K2, Q1.K3, Q1.W FROM
6.   (SELECT D.K2 AS K2, D.K3 AS K3,
   SUM(D.W*E.W) AS W
7.   FROM D Inner Join E ON D.K4=E.K4
8.   GROUP BY D.K2, D.K3)
```

<sup>2</sup> Бурбаки Н. Теория множеств. М.: Мир, 1965. 456 с.



Т а б л и ц а 1. База данных примера  
Table 1. Example database

| A  |     | B  |    |   | C  |    |    | D  |    |    |    | E  |   | Res |        |
|----|-----|----|----|---|----|----|----|----|----|----|----|----|---|-----|--------|
| K1 | W   | K1 | K2 | W | K2 | K3 | W  | K2 | K3 | K4 | W  | K4 | W | K1  | S      |
| p1 | 100 | p1 | e1 | 6 | e1 | s1 | 10 | e1 | s1 | m1 | 10 | m1 | 3 | p1  | 93300  |
| p2 | 200 | p1 | e2 | 3 | e1 | s1 | 5  | e1 | s1 | m2 | 5  | m2 | 5 | p2  | 133000 |
|    |     | p2 | e1 | 2 | e1 | s2 | 7  | e1 | s2 | m4 | 7  | m3 | 2 |     |        |
|    |     | p2 | e2 | 7 | e2 | s1 | 5  | e2 | s1 | m2 | 5  | m4 | 7 |     |        |
|    |     |    |    |   | e2 | s2 | 1  | e2 | s2 | m3 | 1  |    |   |     |        |
|    |     |    |    |   | e2 | s3 | 2  | e2 | s3 | m1 | 2  |    |   |     |        |

Источник: составлено авторами.  
Source: Compiled by the authors.

```

9.          AS Q1)
10.         AS Q2)
11.        AS UA
12.       GROUP BY UA.K2, UA.K3)
13.      AS Q3
14.     ON B.K2=Q3.K2
15.    GROUP BY B.K1, B.K2)
16.   AS Q4
17.  ON A.K1=Q4.K1
18. GROUP BY A.K1

```

Листинг 1. Вычисление таблицы Res  
Listing 1. Res Table Calculation

Этот запрос реализуется следующим образом:

1. Сначала выполняется вложенный запрос Q1, который включается в подзапрос Q2.
2. Подзапрос Q2 – второй операнд в операции Union All, которая выполняется в подзапросе UA.
3. Затем выполняется подзапрос Q3, который включается в подзапрос Q4 в качестве второго операнда операции Inner Join.

4. Последним выполняется основной запрос, в котором подзапрос Q4 используется в качестве второго операнда операции Inner Join.

Подзапросы Q2 и UA реализуют операцию слияния «строго упорядоченных» таблиц.

Как видно из листинга этот запрос, решающий достаточно простую задачу, достаточно сложен как в написании, так и в преобразовании его с целью оптимизации, как в плане распараллеливания на уровне выражения, так и в плане определения наилучшего порядка выполнения операций. Упрощение может быть достигнуто при записи этого запроса в виде выражения в алгебре многомерных матриц.

Домены ключей K1, K2, K3, K4 – конечные множества. Поэтому можно задать нумерацию их элементов. Тогда этим ключам соответствуют индексы  $i_1, i_2, i_3, i_4$ . Таблицам соответствуют матрицы

$$A = \|a_{i_1}\| = \|100, 200\|, B = \|b_{i_2}\| = \begin{bmatrix} 6 & 3 \\ 2 & 7 \end{bmatrix}, C = \|c_{i_3}\| = \begin{bmatrix} 10 & 5 & 7 \\ 5 & 1 & 2 \end{bmatrix}, D = \|d_{i_4}\| = \begin{bmatrix} 10 & 5 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, E = \|e_{i_4}\| = \|5, 2, 7\|.$$

Таблице Res соответствует матрица R, аналогичная матрице A. Выражение, вычисляющее значение матрицы R вид  $R = {}^{0,1}(A \times {}^{0,1}(B \times (C + {}^{0,1}(D \times E))))$ . Очевидно, что это выражение гораздо проще выражения запроса, приведенного в листинге 1. Оно может быть легко преобразовано как вручную, так и, в более сложном случае, посредством специализированного программного обеспечения, которое несложно разработать. Кроме того, методы

распараллеливания сложения и умножения многомерных матриц [24, 25] позволяют построить аналогичные методы для соответствующих реляционных операций. Все это возможно в том случае, когда эти выражения соответствуют друг другу.

Для доказательства того, что эти алгебраические выражения соответствуют друг другу, достаточно подставить их термы в аксиомы A1 и A2. После подстановки будут получены следующие формулы:

$$\begin{aligned}
& {}^{so}(\mathcal{H}(i_1)A(i_1, w_1)) \otimes \\
& ({}^{nso}(\mathcal{H}(i_1))({}^{nso}(\mathcal{H}(i_2))({}^{so}(\mathcal{H}(i_2))B(i_2, w_2)) \otimes \\
& ({}^{nso}(\mathcal{H}(i_1))({}^{so}(\mathcal{H}(i_3))C(i_3, w_3)) \oplus \\
& ({}^{so}(\mathcal{H}(i_2, i_3))({}^{so}(\mathcal{H}(i_4))E(i_4, w_5)) \otimes {}^{nso}(\mathcal{H}(i_2, i_3))D(i_2, i_3, i_4, w_4)) \\
& (i_2, i_3, s_1 = \sum_{(i_4)} w_4 \times w_5))) (i_2, i_3, s_2 = w_3 + s_1)) (i_1, i_2, s_3 = \sum_{(i_3)} w_2 \times s_2)) (i_1, s_4 = \sum_{(i_2)} w_1 \times s_3) \\
& {}^{nso}(\mathcal{H}(K1)A(K1, w_1)) \otimes \\
& ({}^{nso}(\mathcal{H}(K1))({}^{nso}(\mathcal{H}(K2))({}^{so}(\mathcal{H}(K1, K2))B(K1, K2, w_2)) \otimes \\
& ({}^{nso}(\mathcal{H}(K2))({}^{so}(\mathcal{H}(K2, K3))C(K2, K3, w_3)) \oplus \\
& ({}^{so}(\mathcal{H}(K2, K3))({}^{so}(\mathcal{H}(K4))E(K4, w_5)) \otimes {}^{nso}(\mathcal{H}(K2, K3))D(K2, K3, K4, w_4)) \\
& (K2, K3, s_1 = \sum_{(K4)} w_4 \times w_5))) (K2, K3, s_2 = w_3 + s_1)) (K1, K2, s_3 = \sum_{(K3)} w_2 \times s_2)) (K1, s_4 = \sum_{(K2)} w_1 \times s_3)
\end{aligned}$$



Очевидно, что они совпадают с точностью до обозначений. Следовательно, выражение запроса в реляционной алгебре соответствует выражению в алгебре многомерных матриц.

## Заключение

Из сказанного в статье можно сделать следующие выводы.

1. Аксиоматический подход позволяет достаточно просто установить соответствие между многомерно-матричной и реляционной моделями данных при решении задач массовой обработки данных. Однако он может быть применен и для других типов обработки данных, для чего достаточно задать аксиомы, описывающие основные объекты и операции их обработки.
2. В рассматриваемом случае установленное соответствие позволяет заменить сложное выражение запроса в реляционной модели простым выражением в алгебре многомерных матриц. Это существенно облегчит процесс проектирования сложных запросов на обработку данных.
3. Алгебраическое многомерно-матричное выражение проще верифицировать, задавая соответствующие пред-

и постулаты. Кроме того, отсутствие в нем циклов позволяет достаточно просто разработать программный верификатор запросов.

4. Многомерно-матричное представление запроса позволяет легко распараллелить запрос на уровне задачи посредством преобразований выражения, выделяя в нем последовательности операций, которые могут выполняться одновременно.

5. Методы распараллеливания операций сложения и умножения многомерных матриц за счет распределения данных могут быть легко адаптированы для распределения данных при выполнении реляционных операций.

6. Возможны случаи, когда использование многомерных матриц может быть неэффективным, например, когда матрицы сильно разрежены. В этом случае, представление разреженных матриц в виде отношений позволит ускорить обработки при использовании соответствующих реляционных операций.

Таким образом, можно утверждать, что предложенный метод позволяет эффективно решать как теоретические, так и практические задачи массовой обработки данных.

## Список использованных источников

1. Ершов А. П. Вычислимость в произвольных областях и базисах // Проблемы информатики. 2019. № 3 (44). С. 64-92. EDN: YKUIFY
2. Hoare C. A. R. An axiomatic basis for computer programming // Communications of the ACM. 1969. Vol. 10, issue 12. P. 576-580. <https://doi.org/10.1145/363235.36325>
3. Hoare C. A. R. Towards a Theory of Parallel Programming // The Origin of Concurrent Programming ; ed. by P. B. Hansen. New York, NY: Springer, 1972. P. 231-244. [https://doi.org/10.1007/978-1-4757-3472-0\\_6](https://doi.org/10.1007/978-1-4757-3472-0_6)
4. Hoare C. A. R. Parallel programming: An axiomatic approach // Language Hierarchies and Interfaces. Lecture Notes in Computer Science ; ed. by F. L. Bauer [et al.]. Vol. 46. Berlin, Heidelberg: 1976. P. 11-42. [https://doi.org/10.1007/3-540-07994-7\\_47](https://doi.org/10.1007/3-540-07994-7_47)
5. Ануреев И. С., Марьясов И. В., Непомнящий В. А. Верификация С-программ на основе смешанной аксиоматической семантики // Моделирование и анализ информационных систем. 2010. Т. 17, № 3. С. 5-28. EDN: MVLICF
6. Непомнящий В. А., Рякин О. М. Верификация программ обработки данных с использованием автоматной модели файлов // Прикладная информатика. 1988. № 14. С. 180-190. EDN: YIPYZ
7. Непомнящий В. А., Рякин О. М. Прикладные методы верификации программ. М.: «Радио и связь», 1988. 256 с. EDN: RVKWLN
8. Крытый С. Л., Максимец А. Н. Верификация программ: состояние, проблемы, результаты. I // Кибернетика и системный анализ. 2013. Т. 49, № 6. С. 3-14. URL: <http://jnas.nbuv.gov.ua/article/UJRN-0000407572> (дата обращения: 16.07.2023).
9. Промский А. В. Операционная и аксиоматическая семантики языка Си // Четвертый сибирский конгресс по прикладной и индустриальной математике (ИНПРИМ-2000) : Тезисы докладов. Ч. II. Новосибирск: Изд-во Института математики СО РАН, 2000. С. 125.
10. Городняя Л. В. От дискретной математики к семантике языков программирования // Всероссийская научная конференция «Математические основы информатики и информационно-коммуникационных систем»: сб. тр. Всероссийская научная конференция. Тверь: ТвГУ, 2021. С. 141-155. <https://doi.org/10.26456/mfscsics-21-22>
11. Непомнящий В. А. Верификация программ обработки файлов на языке Паскаль // Программирование. 1981. № 2. С. 34-43. EDN: TGZYPZ
12. Непомнящий В. А. Проблемно-ориентированный подход к верификации программ обработки файлов // Программирование. 1989. Т. 15, № 5. С. 5-16. EDN: XWNQPY
13. Stapp L. Axiomatic approach to the system of files // Logics of Programs and Their Applications. Logic of Programs 1980. Lecture Notes in Computer Science ; ed. by A. Salwicki. Vol. 148. Berlin, Heidelberg: Springer, 1983. P. 270-294. [https://doi.org/10.1007/3-540-11981-7\\_20](https://doi.org/10.1007/3-540-11981-7_20)





14. Andrews G. R., Reitman R. P. An axiomatic approach to information flow in programs // ACM Transactions on Programming Languages and Systems (TOPLAS). 1980. Vol. 2, issue 1. P. 56-76. <https://doi.org/10.1145/357084.35708>
15. Chen Y. T., Tanik M. M. An axiomatic approach of software functionality measure // The Third International Workshop on Rapid System Prototyping. NC, USA, Research Triangle Park: IEEE Press, 1992. P. 181-187. <https://doi.org/10.1109/IWRSP.1992.243908>
16. Hintersteiner J. D., Nain A. S. Integrating Software into Systems: An Axiomatic Design Approach // Proceedings of the 3<sup>rd</sup> International Conference on Engineering Design and Automation. Vancouver, B. C. Canada, 1999. P. 1-4. URL: <https://www.axiomaticdesign.com/wp-content/uploads/eda99.pdf> (дата обращения: 16.07.2023).
17. Developing a mobile learning environment: An axiomatic approach / W. Rodriguez [et al.] // Journal of International Technology and Information Management. 2018. Vol. 23, no. 3. P. 3. <https://doi.org/10.58729/1941-6679.1075>
18. An Axiomatic Basis for Communication / M. Karsten [et al.] // ACM SIGCOMM Computer Communication Review. 2007. Vol. 37, no. 4. P. 217-228. <https://doi.org/10.1145/1282427.1282405>
19. Мунерман В. И. Массовая обработка данных. Алгебраические модели и методы: монография. М.: ИНФРА-М, 2023. 229 с. <https://doi.org/10.12737/1906037>
20. Дейкстра Э. В. Дисциплина программирования. М.: Мир, 1978. 275 с.
21. Мунерман В. И. Аксиоматический метод формализации массовой обработки данных в системах высокой доступности // Системы высокой доступности. 2017. Т. 13, № 2. С. 56-62. EDN: ZBKLTN
22. Munerman V., Munerman D. An Axiomatic Approach to the Data Models Formalization for Mass Data Processing // 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). St. Petersburg and Moscow, Russia: IEEE Press, 2020. P. 1996-2000. <https://doi.org/10.1109/EIConRus49466.2020.9039205>
23. Sokolov N. P. Functions of multidimensional matrices and their applications for the solutions of linear systems of partial differential equations // Ukrainian Mathematical Journal. 1970. Vol. 22, issue 6. P. 657-674. <https://doi.org/10.1007/BF01086271>
24. Захаров В. Н., Мунерман В. И. Параллельная реализация обработки интенсивно используемых данных на основе алгебры многомерных матриц // Аналитика и управление данными в областях с интенсивным использованием данных: XVII Международная конференция DAMDID/RCDL. Обнинск: МИФИ, 2015. С. 217-223.
25. Мунерман В. И. Архитектура программно-аппаратного комплекса для массовой обработки данных на базе многомерно-матричной модели // Системы высокой доступности. 2015. Т. 11, № 2. С. 13-18. EDN: UBGEKV

## References

1. Ershov A.P. Rcomputability in Arbitrary Domains and Bases. *Problems of Informatics*. 2019;(3):64-92. (In Russ., abstract in Eng.) EDN: YKUIFY
2. Hoare C.A.R. An axiomatic basis for computer programming. *Communications of the ACM*. 1969;10(12):576-580. <https://doi.org/10.1145/363235.36325>
3. Hoare C.A.R. Towards a Theory of Parallel Programming. In: Hansen P.B. (eds) The Origin of Concurrent Programming. New York, NY: Springer; 1972. p. 231-244. [https://doi.org/10.1007/978-1-4757-3472-0\\_6](https://doi.org/10.1007/978-1-4757-3472-0_6)
4. Hoare C.A.R. Parallel programming: An axiomatic approach. In: Bauer F.L., et al. Language Hierarchies and Interfaces. *Lecture Notes in Computer Science*. Vol. 46. Berlin, Heidelberg: Springer; 1976. p. 11-42. [https://doi.org/10.1007/3-540-07994-7\\_47](https://doi.org/10.1007/3-540-07994-7_47)
5. Anureev I.S., Maryasov I.V., Nepomniaschy V.A. C-programs verification based on mixed axiomatic semantics. *Automatic Control and Computer Sciences*. 2011;45:485-500. <https://doi.org/10.3103/S0146411611070029>
6. Nepomnyashchiy V.A., Ryakin O.M. Verifikaciya programm obrabotki dannyh s ispol'zovaniem avtomatnoj modeli fajlov [Verification of Data Processing Programs Using a File Automaton Model]. *Journal of Applied Informatics*. 1988;(14):180-190. (In Russ., abstract in Eng.) EDN: YIIPYZ
7. Nepomnyashchiy V.A., Ryakin O.M. Prikladnye metody verifikatsii program [Applied program verification methods]. M.: Radio i svyaz; 1988. 255 p. (In Russ.) EDN: RVKWLN
8. Kryvyj S.L., Maksimets A.N. Verifikatsiya programm: sostojanie, problemy, rezultaty. I [Program Verification: Current State, Challenges, and Outcomes. I]. *Cybernetics and Systems Analysis*. 2013;49(6):3-14. Available at: <http://jnas.nbuv.gov.ua/article/UJRN-0000407572> (accessed 16.07.2023). (In Russ., abstract in Eng.)
9. Promsky A.V. Operacionnaya i aksiomaticheskaya semantiki yazyka Si [Operational and Axiomatic Semantics of the C Language]. In: *Chetvertyj sibirskij kongress po prikladnoj i industrial'noj matematike (INPRIM-2000)* = The Fourth Siberian Congress on Applied and Industrial Mathematics (INPRIM-2000). Part II. Novosibirsk: Publ. House of the Institute of Mathematics, SB RAS; 2000. p. 125. (In Russ.)
10. Gorodnyaya L.V. From Discrete Mathematics to the Semantics of Programming Languages. In: All-Russian Scientific Conference on "Mathematical Foundations of Informatics and Information and Communication Systems":



- Proceedings. All-Russian Scientific Conference. Tver: TvSU; 2021. p. 141-155. (In Russ., abstract in Eng.) <https://doi.org/10.26456/mfscsics-21-22>.
11. Nepomnyashchiy V.A. *Verifikaciya programm obrabotki fajlov na yazyke Paskal'* [File processing program verification in Pascal]. *Programmirovaniye*. 1981;(2):34-43. (In Russ., abstract in Eng.) EDN: TGZYPZ
  12. Nepomnyashchiy V.A. *Problemno-orientirovannyj podhod k verifikacii programm obrabotki fajlov* [Problem-Oriented Approach to the Verification of File Processing Programs]. *Programmirovaniye*. 1989;15(5):5-16. (In Russ., abstract in Eng.) EDN: XWNQPZ
  13. Stapp L. Axiomatic approach to the system of files. In: Salwicki A. (eds) *Logics of Programs and Their Applications. Logic of Programs 1980. Lecture Notes in Computer Science*. Vol. 148. Berlin, Heidelberg: Springer; 1983. p. 270-294. [https://doi.org/10.1007/3-540-11981-7\\_20](https://doi.org/10.1007/3-540-11981-7_20)
  14. Andrews G.R., Reitman R.P. An axiomatic approach to information flow in programs. *ACM Transactions on Programming Languages and Systems (TOPLAS)*. 1980;2(1):56-76. <https://doi.org/10.1145/357084.35708>
  15. Chen Y.T., Tanik M.M. An axiomatic approach of software functionality measure. In: *The Third International Workshop on Rapid System Prototyping*. NC, USA, Research Triangle Park: IEEE Press; 1992. p. 181-187. <https://doi.org/10.1109/IWRSP.1992.243908>
  16. Hintersteiner J.D., Nain A.S. Integrating Software into Systems: An Axiomatic Design Approach. In: *Proceedings of the 3rd International Conference on Engineering Design and Automation*. Vancouver, B. C. Canada; 1999. p. 1-4. Available at: <https://www.axiomaticdesign.com/wp-content/uploads/eda99.pdf> (accessed 16.07.2023).
  17. Rodriguez W., et al. Developing a mobile learning environment: An axiomatic approach. *Journal of International Technology and Information Management*. 2018;23(3):3. <https://doi.org/10.58729/1941-6679.1075>
  18. Karsten M., et al. An Axiomatic Basis for Communication. *ACM SIGCOMM Computer Communication Review*. 2007;37(4):217-228. <https://doi.org/10.1145/1282427.1282405>
  19. Munerman V.I. *Mass Data Processing. Algebraic Models And Methods: Monographie*. M.: INFRA-M; 2023. 229 p. (In Russ., abstract in Eng.) <https://doi.org/10.12737/1906037>
  20. Dijkstra E.W. *A Discipline of Programming*. Englewood Cliffs: Prentice-Hall, 1997. 240 p.
  21. Munerman V.I. An axiomatic method of the formalization of mass data processing in highly available systems. *Highly Available Systems*. 2017;13(2):56-62. (In Russ., abstract in Eng.) EDN: ZBKLTN
  22. Munerman V., Munerman D. An Axiomatic Approach to the Data Models Formalization for Mass Data Processing. In: *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EConRus)*. St. Petersburg and Moscow, Russia: IEEE Press; 2020. p. 1996-2000. <https://doi.org/10.1109/EConRus49466.2020.9039205>
  23. Sokolov N.P. Functions of multidimensional matrices and their applications for the solutions of linear systems of partial differential equations. *Ukrainian Mathematical Journal*. 1970;22(6):657-674. <https://doi.org/10.1007/BF01086271>
  24. Zakharov V.N., Munerman V.I. *Parallel'naya realizaciya obrabotki intensivno ispol'zuemyh dannyh na osnove algebry mnogomernykh matric* [Parallel implementation of processing intensively used data based on the algebra of multidimensional matrices]. In: *Selected Papers of the XVII International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2015)*. Obninsk, Russia: MEPhi; 2015. p. 217-223. (In Russ., abstract in Eng.)
  25. Munerman V.I. Construction of Hardware-Software Complexes Architecture to Improve Massively Data Processing. *Highly Available Systems*. 2015;11(2):13-18. (In Russ., abstract in Eng.) EDN: UBGEVCV

Поступила 16.07.2023; одобрена после  
рецензирования 22.09.2023; принята к публикации  
20.11.2023.

Submitted 16.07.2023; approved after reviewing  
22.09.2023; accepted for publication 20.11.2023.

## Об авторах:

**Мунерман Виктор Иосифович**, доцент кафедры информатики физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), кандидат технических наук, доцент, **ORCID: <https://orcid.org/0000-0002-9628-4049>**, [vimoona@gmail.com](mailto:vimoona@gmail.com)

**Мунерман Даниил Викторович**, лаборант-стажер

## About the authors:

**Victor I. Munerman**, Associate Professor of the Chair of Applied Mathematics and Informatics, Faculty of Physics and Mathematics, Smolensk State University, Smolensk (4 Przhivalsky St., Smolensk 214000, Russian Federation), Cand. Sci. (Eng.), Associate Professor, **ORCID: <https://orcid.org/0000-0002-9628-4049>**, [vimoona@gmail.com](mailto:vimoona@gmail.com)

**Daniel V. Munerman**, Laboratory Assistant of the Chair of Applied Mathematics and Informatics, Faculty



кафедры прикладной математики и информатики физико-математического факультета, ФГБОУ ВО «Смоленский государственный университет» (214000, Российская Федерация, г. Смоленск, ул. Пржевальского, д. 4), **ORCID:** <https://orcid.org/0000-0002-5139-6645>, danvmoon@gmail.com

of Physics and Mathematics, Faculty of Physics and Mathematics, Smolensk State University, Smolensk (4 Przhhevsky St., Smolensk 214000, Russian Federation), **ORCID:** <https://orcid.org/0000-0002-5139-6645>, danvmoon@gmail.com

*Все авторы прочитали и одобрили окончательный вариант рукописи.*

*All authors have read and approved the final manuscript.*