УДК 004.272

Б.А. Кулик, В.Г. Курбанов, А.Я. Фридман

ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ДАННЫХ И ЗНАНИЙ МЕТОДАМИ АЛГЕБРЫ КОРТЕЖЕЙ

Кулик Б.А., Курбанов В.Г., Фридман А.Я. **Параллельная обработка данных и знаний методами алгебры кортежей.**

Аннотация. Алгебра кортежей – математическая система для формализации многоместных отношений. С ее помощью можно моделировать в одних и тех же структурах как данные (графы, многоместные отношения), так и знания (семантические сети, модели рассуждений, формулы исчисления высказываний и предикатов, продукционные системы, онтологии и т.д.). В то же время сами эти структуры имеют матрицеподобную форму, а все алгоритмы их обработки легко распараллеливаются.

Ключевые слова: параллельные вычисления, алгебра кортежей, структуры данных, базы знаний.

Kulik B.A., Kurbanov V.G., Fridman A.Ya. **Parallel Processing of Data and Knowledge by Means of N-tuple Algebra.**

Abstract. *N*-tuple algebra is a mathematical system to formalize *n*-ary relations. This algebra provides for modeling both data (graphs, *n*-ary relations, etc.) and knowledge (semantic networks, reasoning models, formulas of propositional and predicate calculi, production systems, ontologies and so on) by the same structures. These structures look like matrices and can be easily processed by parallel algorithms.

Keywords: parallel computation, *n*-tuple algebra, data structure, knowledge base.

1. Введение. Распараллеливание операций наиболее эффективно осуществляется в тех случаях, когда структуры данных представимы в форме матриц или таблиц. Однако во многих задачах и алгоритмах обработки данных и знаний используются структуры, для которых матричная форма представления не предусматривается, к ним относятся и различные структуры искусственного интеллекта (предикаты, правила, логические формулы и т.д.) Для выполнения параллельных вычислений в этих структурах необходимо предварительно составить граф программы; только потом выбираются его независимые ветви, допускающие параллельную обработку. В частности, в современных системах искусственного интеллекта параллелизм достигается в основном за счет построения графа пар резольвирующихся дизъюнктов Ковальского [1]) и выбора в нем тех связей, которые допустимо обрабатывать одновременно [2]. Этот метод очень сложен в использовании (необходимо учитывать много ограничений и условий) и обладает малой эффективностью.

Поэтому целесообразно найти такую обобщенную форму представления структур данных и знаний, которая позволяла бы относительно легко реализовать вычислительный процесс в виде большого числа независимых операций. По мнению авторов, подобные задачи можно решать, если выразить многие типы данных и знаний в структурах алгебры кортежей [3, 4], одним из свойств которых является матрицеподобная форма представления и возможность эффективного распараллеливания операций.

2. Краткие сведения об алгебре кортежей. *Алгебра кортежей* (АК) математически определена как алгебраическая система, в которой должны быть заданы носитель, совокупность операций, совокупность отношений, а также основные свойства операций и отношений. В некоторых случаях свойства однозначно определяются с помощью доказательства изоморфизма данной системы и какой-либо известной алгебраической системы. В частности, доказано, что *АК изоморфна алгебре множеств и относится к классу булевых алгебр* [5].

Носителем в АК служит произвольное множество многоместных отношений, для представления которых предложено четыре типа структур (о них далее), называемых *АК-объектами*. Каждый АК-объект погружен в определенное пространство *атрибутов*, множество значений атрибута называется *доменом*. Имена АК-объектов содержат идентификатор и заключенную в прямые скобки последовательность имен атрибутов, входящих в *схему отношения*, в которой задан этот АК-объект. Например, запись $R[XYZ]$ говорит, что АК-объект R определен в пространстве атрибутов X, Y и Z.

«Произвольное множество многоместных отношений», определенное выше как носитель АК, подразумевает, что для этих отношений нет никаких ограничений на состав атрибутов, в которых они заданы – пространства атрибутов разных отношений могут совпадать, а могут существенно отличаться, и эта «произвольность» не нарушает свойств АК как алгебраической системы.

Основные операции АК включают *операции алгебры множеств* (пересечение, объединение, дополнение) и *операции с атрибутами* (переименование, перестановка, элиминация, добавление фиктивного атрибута). Комбинации перечисленных операций позволяют ввести *производные операции* с отношениями: соединение, композиция, транзитивное замыкание и т.д. Для сравнения АК-объектов используются два *основных отношения – включения* и *равенства*. По своим аналитическим возможностям АК сопоставима с исчислением предикатов, причем АК-объекты моделируют области истинности предикатов и логических формул.

АК-объекты обеспечивают сжатое отображение многоместных отношений. При необходимости можно с помощью стандартных алгоритмов преобразовать АК-объекты в обычные многоместные отношения, состоящие из множеств кортежей элементов (в АК эти кортежи называются *элементарными кортежами*).

Однотипные АК-объекты – это структуры, заданные в одном пространстве атрибутов. В АК можно выполнять все теоретико-множественные операции не только с однотипными отношениями, но и с отношениями, имеющими разные схемы.

АК-объекты (C-кортеж, C-система, D-кортеж, D-система) формируются в виде матриц из подмножеств доменов атрибутов, называемых *компонентами*. В их число входят две *фиктивные компоненты*:

– *полная компонента* (обозначается “$*$”) есть множество, равное домену соответствующего (по месту ее расположения в кортеже) атрибута; например, если задан АК-объект $R[XYZ] = [A * C]$, то символ “$*$” здесь обозначает множество, равное домену атрибута Y.

– *пустое множество* – $\varnothing$.

Перейдем к описанию основных структур АК – C-систем и D-систем.

C-система записывается в виде матрицы, состоящей из компонент-множеств и ограниченной прямыми скобками. Например, $R[XYZ] = \begin{bmatrix} A_1 & A_2 & A_3 \\ B_1 & B_2 & B_3 \end{bmatrix}$ есть C-система, ее можно преобразовать в обычное отношение (т.е. во множество кортежей) следующим образом: $R[XYZ] = (A_1 \times A_2 \times A_3) \cup (B_1 \times B_2 \times B_3)$. При этом A_1, $B_1 \subseteq X$; A_2, $B_2 \subseteq Y$; A_3, $B_3 \subseteq Z$.

С помощью C-систем удобно представлять дизъюнктивные нормальные формы одноместных предикатов. C-система, состоящая из одной строки, называется *C-кортежем*. В логике C-кортежу соответствует отдельный конъюнкт.

D-системы в АК моделируют конъюнктивные нормальные формы одноместных предикатов. D-система обозначается матрицей компонент-множеств, ограниченной перевернутыми прямыми скобками.

Используя D-системы, легко вычислить дополнения C-систем. Так, D-система $\overline{T}[XYZ] = \left]\begin{matrix} \overline{A} & \varnothing & \overline{C} \\ \overline{D} & \overline{E} & \varnothing \end{matrix}\right[$ есть дополнение C-системы $T[XYZ] = \begin{bmatrix} A & * & C \\ D & E & * \end{bmatrix}$.

Аналогично C-системе, D-система из одной строки называется *D-кортежем*, ему в логике соответствует отдельный дизъюнкт.

Правила выполнения операций объединения и пересечения для C- и D-структур имеют свою специфику, они подробно описаны в [3,4]. Отметим, что в АК для выполнения всех теоретико-множественных операций и проверок отношений (равенства, включения и т.д.) нет необходимости преобразовывать АК-объекты во множество элементарных кортежей – все операции выполняются с матричными формами.

Для выполнения операций с АК-объектами, имеющими разные схемы отношений, вводятся *операции с атрибутами*, в частности, *добавление фиктивного атрибута* ($+Atr$) и *элиминация атрибута* ($-Atr$). Операция $+Atr$ соответствует *правилу обобщения* в исчислении предикатов, поэтому семантика отношений при ее выполнении не нарушается. Операция производится добавлением имени нового атрибута в схему отношения АК-объекта и нового столбца с фиктивными компонентами – в матричное представление. Например, пусть $R_k[XZ] = \begin{bmatrix} A_1 & A_3 \\ B_1 & B_3 \end{bmatrix}$.

Если это отношение моделирует предикат $R_k(x, z)$, то добавление фиктивного атрибута Y в $R_k[XZ]$ соответствует формуле $\forall y(R_k(x, z))$. Для АК-объекта $R_k[XZ]$ добавление фиктивного атрибута Y выполняется так: $+Y(R_k[XZ]) = \begin{bmatrix} A_1 & * & A_3 \\ B_1 & * & B_3 \end{bmatrix}$.

Операция $+Atr$ часто используется для приведения разнотипных АК-объектов к одной схеме отношения, после чего можно, применяя стандартные алгоритмы АК, выполнять все необходимые операции и проверки. С учетом этого введены *обобщенные операции* ($\cap_G$, $\cup_G$), в которых операции $\cap$ и $\cup$ выполняются после приведения разнотипных АК-объектов к одной схеме отношения. Операции $\cap_G$ и $\cup_G$ семантически соответствуют логическим связкам конъюнкции и дизъюнкции. Доказано, что алгебра отношений с обобщенными операциями изоморфна обычной алгебре множеств. Тем самым в теории отношений было снято ограничение, что законы алгебры множеств выполняются только для отношений, определенных на одном и том же декартовом произведении.

Операция элиминации атрибута (например, X) при применении к C-структурам соответствует навешиванию квантора $\exists x$, а к D-структурам – навешиванию квантора $\forall x$ на соответствующие логические формулы. В АК эта операция выполняется как удаление атрибута из схемы отношения и соответствующего ему столбца из матричного представления АК-объекта. Например, для D-системы $R[XYZ] = \left]\begin{matrix} A & \varnothing & B \\ C & D & \varnothing \end{matrix}\right[$ при вычислении $-Y(R[XYZ])$ получим

$Q[XZ] = \left]\begin{matrix} A & B \\ C & \varnothing \end{matrix}\right[$. При этом, если D-системе $R[XYZ]$ соответствует логическая формула $F(x, y, z)$, то D-системе $Q[XZ]$ – логическая формула $\forall y F(x, y, z)$.

Расширенный вариант АК предусматривает применение некоторых совокупностей атрибутов в качестве отдельных (составных) атрибутов. Тогда компонентами АК-объектов становятся не обычные множества, а многоместные отношения, выраженные АК-объектами. Другим расширением является использование в структурах АК атрибутов, домены которых представлены структурами, отличающимися по свойствам от обычных множеств (например, нечеткими множествами). Для атрибутов, представляющих систему интервалов на числовой оси, в АК разработан метод квантования интервалов, после применения которого такие атрибуты можно обрабатывать всеми методами АК [5].

Значительная часть алгоритмов выполнения операций АК (пересечение, объединение, дополнение, проверка включения АК-объектов и т.д.) имеет полиномиальную сложность. Экспоненциальной сложностью характеризуется, в частности, алгоритмы преобразования C-систем в D-системы и обратно. Во многих процедурах обработки информации подобные преобразования не нужны. Если все же возникает необходимость применения таких алгоритмов, их трудоемкость удается снизить до полиномиальной для некоторых частных случаев структур АК-объектов [6].

В настоящее время исследована и обоснована возможность применения АК в следующих структурах данных и знаний [5]:

1) графы и сети;

2) модели исчисления высказываний и предикатов;

3) системы искусственного интеллекта (семантические сети, экспертные системы, фреймы, онтологии);

4) модели дедуктивных и абдуктивных рассуждений;

5) логико-вероятностные методы, включая вероятностную логику;

6) дискретные автоматы.

Продолжаются исследования, связанные с возможностью использования АК в других областях, в частности, в структурах с неопределенностями и в динамических системах.

3. Методы и средства распараллеливания операций в структурах АК. В основе операций АК лежат теоретико-множественные операции с компонентами (подмножествами доменов атрибутов). Например, для вычисления пересечения двух однотипных C-кортежей требуется найти пересечение пар компонент в соответствующих атрибутах, причем в результате получится пустое множество, если пересечение хотя бы одной пары компонент пусто. Пересечение C-кортежа R с однотипной C-системой Q вычисляется как результат пересечения R с каждым C-кортежем из Q. Для двух однотипных C-кортежей R и R_1 имеет место включение $R \subseteq R_1$, если все компоненты R включены в соответствующие компоненты R_1. Некоторые операции в АК не требуют операций с компонентами. Например, объединение двух однотипных C-систем получается просто приписыванием всех C-кортежей одной из этих C-систем другой C-системе.

Приведенные примеры достаточно полно характеризуют специфику выполнения операций в АК. Для обоснования наиболее подходящей архитектуры вычислительного комплекса, предназначенного для распараллеливания операций в АК, целесообразно за основу взять часто встречающуюся операцию пересечения двух C-систем. Пусть заданы C-системы:

$$R_1[XYZ] = \begin{bmatrix} \{a,b,d\} & \{f,h\} & \{b\} \\ \{b,c\} & * & \{a,c\} \end{bmatrix} \text{ и}$$

$$R_2[XYZ] = \begin{bmatrix} \{a,d\} & * & \{b,c\} \\ \{b,d\} & \{f,h\} & \{a,c\} \\ \{b,c\} & \{g\} & \{b\} \end{bmatrix}.$$

Их пересечение вычисляется как пересечение каждого C-кортежа первой C-системы с каждым C-кортежем второй:

$$[\{a, b, d\}\ \{f, h\}\ \{b\}] \cap [\{a, d\} * \{b, c\}] = [\{a, d\}\ \{f, h\}\ \{b\}];$$

$$[\{a, b, d\}\ \{f, h\}\ \{b\}] \cap [\{b, d\}\ \{f, h\}\ \{a, c\}] = \varnothing;$$

$$[\{a, b, d\}\ \{f, h\}\ \{b\}] \cap [\{b, c\}\ \{g\}\ \{b\}] = \varnothing;$$

$$[\{b, c\} * \{a, c\}] \cap [\{a, d\} * \{b, c\}] = \varnothing;$$

$$[\{b, c\} * \{a, c\}] \cap [\{b, d\}\ \{f, h\}\ \{a, c\}] = [\{b\}\ \{f, h\}\ \{a, c\}];$$

$$[\{b, c\} * \{a, c\}] \cap [\{b, c\}\ \{g\}\ \{b\}] = \varnothing.$$

Затем из непустых C-кортежей формируем C-систему:

$$R_1 \cap R_2 = \begin{bmatrix} \{a,d\} & \{f,h\} & \{b\} \\ \{b\} & \{f,h\} & \{a,c\} \end{bmatrix}.$$

На этом типичном примере нетрудно сформулировать основные закономерности распараллеливания операций в АК. Во-первых, можно одновременно выполнять операции с компонентами при вычислении пересечения пар кортежей. Во-вторых, параллельное выполнение допускают операции пересечения одного из кортежей первой C-системы со всеми *C*-кортежами второй. И, в-третьих, при определенной организации вычислительного процесса можно выполнить параллельно все вычисления с компонентами, необходимые для получения окончательного результата.

Выигрыш во времени расчета за счет распараллеливания вычислений в АК будет достаточно велик. Например, при вычислении пересечения двух *C*-систем в пространстве из *K* атрибутов по третьему из перечисленных вариантов распараллеливания, если первая *C*-система содержит *M* строк, а вторая – *N*, то время расчета уменьшится в $K \times M \times N$ раз. Той же сложностью и такой же возможностью распараллеливания обладает и операция проверки включения одного АК-объекта в другой, если первый выражен как *C*-система, а второй – как *D*-система. Здесь в качестве элементарной операции с компонентами используется операция проверки включения множеств.

Рассмотренную операцию пересечения *C*-систем можно считать типовой операцией, поскольку она соответствует следующим операциям в традиционных системах: 1) конъюнкция предикатов или формул, выраженных *C*-системами; 2) дизъюнкция предикатов или формул, представленных как *D*-системы; 3) операции соединения и композиции отношений, которые, в частности, используются при вычислении транзитивного замыкания отношения; 4) операция поиска ответа на вопрос в тех случаях, когда структура данных и структура вопроса выражены *C*-системами. Ясно, что для выполнения операций с компонентами АК-объектов достаточно использовать в качестве процессорного элемента (ПЭ) вычислительное устройство, способное обрабатывать булевы векторы. Для каждого атрибута целесообразно предусмотреть множество однородных ПЭ с общей памятью, в которой хранятся заданные и получаемые в процессе вычислений компоненты данного атрибута. Такая структура соответствует SMP архитектуре (symmetric multiprocessing) [7].

Работу со множеством атрибутов предпочтительно организовать с помощью модулей SMP в качестве элементов вычислительных систем с распределенной памятью MPP (massive parallel processing) [8]. Поскольку атрибуты могут существенно различаться по числу значений, в вычислительный комплекс следует включить разные SMP-кластеры, отличающиеся разрядностью процессорных элементов и соответствующих им модулей памяти. Таким образом, в общем случае для параллельной реализации операций со структурами АК подходит гибридная архитектура, сочетающая в себе особенности систем с MPP и SMP архитектурой. Схема такого комплекса показана на рисунке 1.

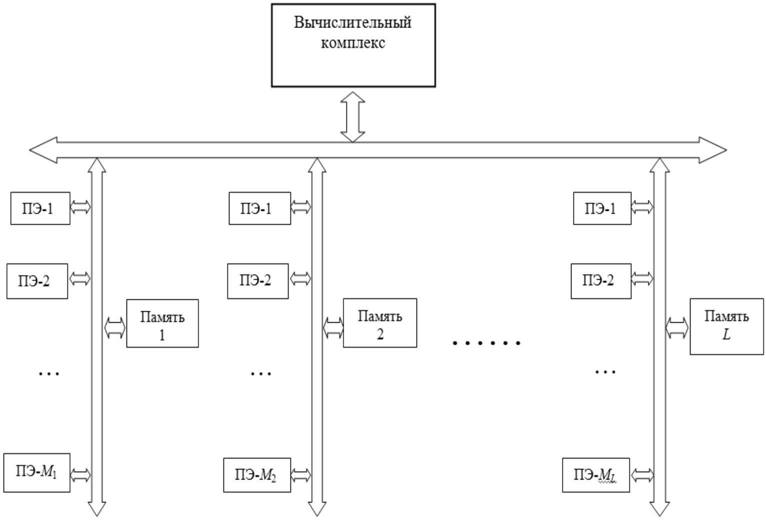

Рис. 1. Схема гибридной MPP-SMP архитектуры

Рассмотрим, как выполняется операция пересечения двух C-систем A и B, представленных в виде матриц $\|a_{iv}\|$ и $\|b_{jv}\|$ с размерностью $M \times K$ и $N \times K$ соответственно. Пусть из множества SMP-кластеров вычислительного устройства выбраны те, которые соответствуют атрибутам пространства, где заданы A и B. Совокупность всех компонент каждого v-го атрибута размешается в памяти соответствующего SMP-кластера (на рис. 1, например, Память 1, Память 2, …, Память K). Для вычисления пересечения векторов $\|a_{1v}\|$ и $\|b_{1v}\|$ в соответствующие процессорные элементы ПЭ-1 в каждом из SMP-кластеров загружаются компоненты этих векторов и выполняется их пересечение. Если все результаты непустые, то они записываются в соответствующие ячейки Памяти v. Если требуется выполнить одновременно пересечение вектора $\|a_{1v}\|$ со всеми векторами матрицы $\|b_{jv}\|$, то в соответствующих SMP-кластерах компоненты векторов одновременно передаются в процессорные элементы ПЭ-1, ПЭ-2, …, ПЭ-N и выполняются требуемые операции. Для параллельного вычисления пересечений всех векторов матриц $\|a_{iv}\|$ и $\|b_{jv}\|$ необходимо использовать дополнительные ресурсы вычислительного комплекса. Возможны два варианта: 1) использовать процессорные элементы ПЭ-N+1, ПЭ-N+2 и т.д. в используемых SMP-кластерах; 2) использовать другие SMP-кластеры.

4. Заключение. Алгебра кортежей позволяет представлять и обрабатывать данные и знания разнообразных форматов. В настоящей работе предложена гибридная архитектура вычислительного комплекса, обеспечивающая эффективное распараллеливание операций на структурах АК.

Литература

1. *Kowalski R. A*. Proof Procedure Using Connection Graphs // J. of the ACM. 1975. vol. 22(4). pp. 572–599.
2. *Loganantharaj R., Mueller R. A.* Parallel Theorem Proving with Connection Graph // 8th Int. Conf. on Autom. Deduc. LNCS 230. 1986. pp. 337–352.
3. *Кулик Б.А.* Вероятностная логика на основе алгебры кортежей // Известия РАН. Теория и системы управления. 2007. № 1. С. 118–127.
4. *Kulik B., Fridman A., Zuenko A.* Logical Inference and Defeasible Reasoning in N-tuple Algebra // In: “Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems”. IGI Global. 2013. pp. 102–128.
5. *Кулик Б.А., Зуенко А.А., Фридман А.Я.* Алгебраический подход к интеллектуальной обработке данных и знаний // СПб.: Изд-во Политехн. ун-та. 2010. 235 с.
6. *Кулик Б.А.* Новые классы КНФ с полиномиально распознаваемым свойством выполнимости // Автоматика и телемеханика. 1995. № 2. С. 111–124.
7. *Severance C., Dowd K.* High Performance Computing (RISK Architectures, Optimization & Benchmarks). 2nd Edition // O’Reily Media. 1998. 466 p.
8. *Batcher K. E.* Design of a Massively Parallel Processor // IEEE Transactions on Computers. 1980. vol. C–29 (9). pp. 836–840.

References

1. Kowalski R. A. Proof Procedure Using Connection Graphs. J. of the ACM. 1975. vol. 22(4). pp. 572–599.
2. Loganantharaj R., Mueller R. A. Parallel Theorem Proving with Connection Graph. 8th Int. Conf. on Autom. Deduc. LNCS 230. 1986. pp. 337–352.
3. Kulik B. A. [Probabilistic logic based on the N-tuple Algebra]. *Izvestija RAN. Teorija i sistemy upravlenija – Journal of Computer and Systems Sciences International*. 2007. vol. 1. pp. 118–127. (In Russ.).
4. Kulik B., Fridman A., Zuenko A. Logical Inference and Defeasible Reasoning in N-tuple Algebra. In: “Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems”. IGI Global. 2013. pp. 102–128.
5. Kulik B. A., Zuenko A. A. Fridman A. Ya. *Algebraicheskij podhod k intellektual'noj obrabotke dannyh i znanij* [Algebraic Approach to Intelligent Processing of Data and Knowledge]. Saint-Petersburg. Polytechnic Univ. Publ. 2010. 235 p. (In Russ.).
6. Kulik B. A. [New classes of CNF with polynomially recognizable property of satisfiability]. *Avtomatika i telemehanika – Automation and Remote Control*. 1995. vol. 2. pp. 111–124. (In Russ.).
7. Severance C., Dowd K. High Performance Computing (RISK Architectures, Optimization & Benchmarks). 2nd Edition. O’Reily Media. 1998. 466 p.
8. Batcher K. E. Design of a Massively Parallel Processor. IEEE Transactions on Computers. 1980. vol. C–29 (9). pp. 836–840.

Поддержка исследований. Работа выполнена при финансовой поддержке РФФИ (проекты №№ 12-07-00302-а, 12-07-00550-а, 12-07-00689-а, 13-07-00318-а, 14-07-00256-а, 14-07-00257-а, 14-07-00205-а) и Президиума РАН (проект 4.3 Программы № 16).

Acknowledgements. This research is supported by RFBR (grants 12-07-00302-a, 12-07-00550-a, 12-07-00689-a, 13-07-00318-a, 14-07-00256-a, 14-07-00257-a, 14-07-00205-a), presidium RAS (grant 4.3 of the program №16).

Кулик Борис Александрович — д-р физ.-мат. наук, доцент, ведущий научный сотрудник лаборатории интеллектуальных электромеханических систем, Институт проблем машиноведения РАН. Область научных интересов: искусственный интеллект, методы логического анализа систем, логико-вероятностный анализ. Число научных публикаций — 90. ba-kulik@yandex.ru; 199178 Санкт-Петербург, В.О., Большой проспект, д.61; р.т. +7(812) 321 90 07.

Kulik Boris Alexandrovich — Ph.D., Dr. Sci., associate professor, leading researcher of Laboratory of Methods and Systems of Automation, Institute of Problems in Mechanical Engineering (IPME RAS). Scientific interests: artificial intelligence techniques of logical analysis systems, logic and probabilistic analysis. The number of publications — 90. ba-kulik@yandex.ru; 61, V.O., Bol'shoi av., 199178, Saint-Petersburg, Russian Federation; office phone +7(812) 321 90 07.

Курбанов Вугар Гариб оглы — к-т физ.-мат. наук, доцент, старший научный сотрудник, Институт проблем машиноведения РАН. Область научных интересов: общие вопросы математики, математическая кибернетика, математическое моделирование, оптимальные системы. Число научных публикаций — 60. vugar_borchali@yahoo.com; 196135, Санкт - Петербург, Большой пр., 61, В.О.; р.т. +7(812)3219007

Kurbanov Vugar Garib ogli — Ph.D., associate professor, senior researcher Institute of Problems in Mechanical Engineering (IPME RAS). Scientific interests: general questions of mathematical cybernetics, mathematical modeling, optimum systems. The number of publications — 60. vugar_borchali@yahoo.com; 61, V.O., Bol'shoi av., 199178, Saint-Petersburg, Russian Federation; office phone +7(812)3219007.

Фридман Александр Яковлевич — д-р техн. наук, профессор, ведущий научный сотрудник лаборатории информационных технологий управления промышленно-природными системами, Институт информатики и математического моделирования Кольского научного центра РАН. Область научных интересов: моделирование комплексных технологий и их воздействия на окружающую среду, прикладные интеллектуализированные системы. Число научных публикаций — 230. fridman@iimm.kolasc.net.ru; 184209, Мурманская обл., г.Апатиты, ул.Ферсмана, 24а; р.т. +7(81555) 79782.

Fridman Alexander Jakovlevich — Ph.D., Dr. Sci., professor, leading researcher of Laboratory of Information Technologies for Nature-Industrial Systems Control, Institute for Informatics and Mathematical Modelling of Technological Processes of RAS. Scientific interests: modeling of complex technologies and their impact on the environment, application intellectualized system. The number of publications — 230. fridman@iimm.kolasc.net.ru; 24A, Fersman st., 184209, Apatity, Murmansk Region, Russian Federation; office phone +7(81555) 79782.

РЕФЕРАТ

Кулик Б.А., Курбанов В.Г., Фридман А.Я. **Параллельная обработка данных и знаний методами алгебры кортежей.**

При формализации и компьютерной реализации данных и знаний часто используются структуры, которые не представлены в матричной форме и поэтому для них трудно осуществить параллельную обработку данных. К таким трудным для ускорения вычислений структурам относятся различные формы искусственного интеллекта (предикаты, правила, логические формулы, семантические сети и т.д.) Для выполнения параллельных вычислений в этих структурах необходимо предварительно составить граф программы; только потом выбираются его независимые ветви, допускающие параллельную обработку. Этот метод очень сложен в использовании (необходимо учитывать много ограничений и условий) и обладает малой эффективностью.

Поэтому целесообразно найти такую обобщенную форму представления структур данных и знаний, которая позволяла бы относительно легко реализовать вычислительный процесс в виде большого числа независимых операций. По мнению авторов, подобные задачи можно решать, если выразить многие типы данных и знаний в структурах алгебры кортежей одним из свойств которых является матрицеподобная форма представления и возможность эффективного распараллеливания операций. С помощью алгебры кортежей отображаются практически все структуры искусственного интеллекта и в то же время структуры алгебры кортежей имеют матричную форму и тем самым позволяют легко распараллеливать алгоритмы обработки данных и знаний. Для этого предложена гибридная архитектура вычислительного комплекса, обеспечивающая эффективное распараллеливание операций на структурах алгебры кортежей.

SUMMARY

Kulik B.A., Kurbanov V.G., Fridman A.Ya. **Parallel Processing of Data and Knowledge by Means of N-tuple Algebra.**

The formalization and implementation of computer data and knowledge are often used structures that are not represented in matrix form and therefore difficult for them to implement parallel processing. Such difficult to speed up the calculations structures include various forms of artificial intelligence (predicates, rules, logical formulas, semantic network, etc.) to perform parallel computations in these structures must first create a graph of the program; Only then chooses its independent branches, allowing parallel processing. This technique is very complicated to use and low efficient due to the necessity of considering many conditions and restrictions.

Efficient paralleling of algorithms is achievable in cases when initial data are represented in a matrix form. Conversely, most knowledge processing and analyzing systems use structures not similar to matrices. This is why it looks reasonable to propose such a generalized structure form for data and knowledge representation that would allow for a comparatively easy transformation of a computational process into a large number of independent operations. In our opinion, N-tuple algebra (NTA) can solve this kind of problems by expressing many formats of data and knowledge as NTA structures. With NTA displayed virtually all structures of artificial intelligence, and at the same time, the structure of the N-tuple algebra have a matrix form, and thus make it easy to parallelize algorithms for processing data and knowledge. For this proposed hybrid architecture computing system that provides effective parallelization of operations on structures N-tuple algebra.