ПРОГРАММИРОВАНИЕ, 2021, № 1, с. 35–38

КОМПЬЮТЕРНАЯ АЛГЕБРА

УДК 004.421.6

МЕТОД ЦВЕТНЫХ ГРАФОВ ДЛЯ УПРОЩЕНИЯ ВЫРАЖЕНИЙ С ИНДЕКСАМИ

© 2021 г. Г. Б. Шпиз[a],*, А. П. Крюков[a],**

[a] *Московский государственный университет имени М.В. Ломоносова*
119991 Москва, ГСП-1, Ленинские горы, д. 1, Россия

**E-mail: shpiz@theory.sinp.msu.ru*

***E-mail: kryukov@theory.sinp.msu.ru*

Поступила в редакцию 06.08.2020 г.
После доработки 28.08.2020 г.
Принята к публикации 14.09.2020 г.

Компьютерная алгебра все шире применяется в научных и прикладных вычислениях. В качестве примера приведем тензорные вычисления или в более широком смысле этого слова – упрощение выражений с индексами. В настоящей статье развивается метод цветных графов для упрощения абстрактных выражений с индексами на случай, когда индексы могут быть отнесены к нескольким различным типам. Примерами таких индексов могут быть верхние и нижние индексы в тензорных выражениях. Предложенный подход позволяет значительно уменьшить число перебираемых вариантов при поиске канонической формы выражения, что резко ускоряет процесс вычислений.

DOI: 10.31857/S0132347421010118

1. ВВЕДЕНИЕ

Выражения с индексами являются одним из самых распространенных математических объектов, которые используются для вычислений в различных областях математики, физике, других областях науки, а также в прикладных областях, а в первую очередь в инженерных науках. Наиболее типичный пример таких вычислений – это тензорные вычисления. Мы будем рассматривать абстрактные выражения с индексами, обладающие определенными свойствами симметрии по отношению к их перестановкам, и линейными соотношениями между собой.

Представленная работа является логическим продолжением работы [1], в которой были даны основные определения для случая, когда индексы одного типа. В настоящей работе мы даем обобщение предложенного формализма на случай индексов нескольких типов. Примерами таких индексов могут быть верхние и нижние индексы у тензоров.

Основная задача, которую мы решаем – это приведение выражения с индексами к канонической форме. Для этого будут использованы цветные графы, связанные с выражениями, и их автоморфизмы. Само выражение с индексами является коммутативным произведением индексированных символов (элементарных индексированных выражений), причем каждый индекс имеет “цвет”, зависящий только от места индекса, но не от его имени.

Заметим, что как и прежде нас не интересует природа индексных выражений, а мы рассматриваем их как некоторые абстрактные объекты, имеющие определенный набор свойств по отношению к перестановкам индексов.

В работе мы будем придерживаться правила суммирования Эйнштейна, когда наличие пары одинаковых индексов означает сумму по всем значениям данной пары:

$$T_i^i \equiv \sum_{i=1}^{i=n} T_i^i,$$

где n – это размерность пространства.

Такие индексы называются индексами суммирования или немыми индексами.

В данной работе мы не будем останавливаться на примерах использования выражений с индексами и типах алгоритмов, используемых для их упрощения. Скажем только, что традиционно для упрощения выражений с индексами используются следующие четыре подхода и их вариации:

- двойной класс смежности [7];
- групповая алгебра [8];
- базис Гребнера [5];
- изоморфизм графов [1].

Желающие могут обратиться к работам А.В. Корольковой, Д.С. Кулябова, Л.А. Севастьянова [2] и Г. Шпиза и А. Крюкова [1], в которых данные вопросы рассмотрены более подробно. Там же можно найти большой список литературы, в котором описаны традиционно применяемые подходы и алгоритмы.

Из работ, которые вышли в последнее время или не были упомянуты в работе [1], в первую очередь хочется указать на подробный обзор М. Маскалума [3] по использованию компьютерной алгебры в задачах теории гравитации, которые являются одним из наиболее частых примеров тензорных вычислений. Упомянем работу, которая использует алгебру графов, но для ограниченного случая полиномов из тензоров Римана, – это работа [4]. С точки зрения компьютерной алгебры, выражения с индексами могут быть разного типа. В частности, задача приведения многопетлевых интегралов в квантовой теории поля [6] может рассматриваться как такой пример.

Структура статьи следующая. В разделе 2 дадим постановку задачи. В разделе 3 дадим строгое определение выражений с абстрактными индексами и их свойств. В этом же разделе дадим строгое определение оператора приведения к каноническому виду. В заключение перечислим полученные результаты, а также направление дальнейшего развития предложенных методов.

2. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Прежде чем перейти к основной теме работы дадим несколько базовых определений объектов, с которыми мы будем работать в дальнейшем.

Определение 1. *Назовем* ***цветным символом*** *или просто* ***символом*** *выражение вида* $t(i_1,\dots,i_n,c_1,\dots,c_n)$, *где* $t\in T$ *– множество типов символов,* $i\in I$ *– множество индексов,* $c\in C$ *– множество цветов индексов* i.

Произведение символов t *будем называть выражением с индексами.*

Символы t, могут обладать различными свойствами симметрии по отношению к перестановкам индексов, которые являются подгруппой группы перестановок, а также линейными соотношениями, которые включают три или более символа. Индексы в символьном выражении, встречающиеся дважды (возможно с разными цветами), считаются индексами суммирования. Индексы суммирования и только они будут именоваться натуральными числами.

Замечание. Если оба индекса суммирования принадлежат одному символу, то преобразования такого символа полностью факторизуются от остального выражения и нами рассматриваться не будут как тривиальные.

Будем считать, что на множестве символов определен лексикографический порядок, причем:

- любой тип символа меньше индекса;
- любой свободный индекс меньше любого индекса суммирования;
- индексы суммирования и только они являются натуральными числами;
- цвета соответствуют номерам индексов, а не их именам. Таким образом при переименовании индексов их цвета не меняются.

Таким образом, мы определяем лексикографический порядок на символах $t(i_1,\dots,i_n,c_1,\dots,c_n)$ следующим образом.

$t(i_1,\dots,i_n,c_1,\dots,c_n) < t'(i'_1,\dots,i'_n,c'_1,\dots,c'_n) \Leftrightarrow (t, i_1, \dots, i_n, c_1,\dots,c_n) < (t',i'_1,\dots,i'_n,c'_1,\dots,c'_n)$ в лексикографическом смысле. Мы также считаем что на множестве мономов (произведение символов) определен естественный лексикографический порядок, а множество всех мономов становится линейно упорядоченным.

Важную роль в дальнейших построениях будет играть понятие сигнатуры символа.

Определение 2. ***Сигнатурой*** *k-го индекса i_k символа $v = t(i_1,\dots i_n,c_1,\dots,c_n)$ будем называть пару (c_k,i_k) и обозначать эту сигнатуру σ_k.*

Раскраска индексов может использоваться для различения пространств с которыми эти индексы связаны. Например, верхние индексы могут быть связаны с некоторым (арифметическим) векторным пространством, а нижние с его сопряженным ($t^{kl}_{\mu\nu}$). Индексы в различных типах символов, также могут быть связаны с различными пространствами. Так, индексы в символах частных производных связаны с пространством линейных дифференциальных операторов с постоянными коэффициентами, а индексы в символах функций с пространством аргументов. Например, если символ соответствует дифференциальному оператору вида $p(i_1,\dots)d(j_1,\dots)$, то цвет может определять принадлежность индекса коэффициенту или дифференцированию.

Определение 3. ***Сигнатурой символа*** *$t(i_1, \dots, i_n, c_1,\dots,c_n)$ будем называть последовательность $\sigma(t) = (t,\sigma_1,\dots,\sigma_n)$, где $(\sigma_1,\dots,\sigma_n)$ – упорядоченная по возрастанию последовательность сигнатур индексов $i_1,\dots,i_n$.*

Будем считать, что любое множество символов с попарно различными сигнатурами линейно независимо, а символы с одинаковыми сигнатурами и только они могут быть связаны линейными соотношениями. Множество всех таких соотношений будем обозначать R и будем предполагать, что любое переименование индексов переводит линейное соотношение из R в R, то есть множество соотношений инвариантно относительно переименований индексов.

Определение 4. *Сверточным выражением будем называть такое произведение символов, в котором ни один индекс не встречается более 2-х раз. Индексы, встречающиеся в сверточном выражении один раз, будем называть* ***свободными****, а встречающиеся 2 раза —* ***индексами суммирования****.*

Сверточные выражения будем называть **слабо эквивалентными**, если они получаются друг из друга перестановкой сомножителей и переименованием индексов суммирования. Класс слабой эквивалентности сверточного выражения m будем называть **мономом** и обозначать $[m]$.

3. СТРУКТУРНЫЙ ГРАФ МОНОМА

Пусть задано сверточное выражение $m = v_1 \dots v_n$, где v_i — символ.

Рассмотрим граф, вершины которого совпадают с входящими в m символами. Вершины считаются соединенными ребром тогда и только тогда, когда они содержат хотя бы один общий индекс.

Цветом вершины будем называть ее сигнатуру, а цветом ребра, соединяющего вершины v_i и v_j причем $i \leq j$ — последовательность $(i, \sigma(v_i), j, \sigma(v_j), c)$, где c — последовательность цветов общих индексов (суммирования) для v_i и v_j, упорядоченная по возрастанию.

Определение 5. *Описанный выше раскрашенный граф называется* ***графом*** *сверточного выражения m и обозначается $G(m)$.*

Заметим, что раскраска графа $G(m)$ зависит от порядка сомножителей и от переименования индексов суммирования.

Определение 6. *Упорядоченную по возрастанию последовательность цветов всех ребер графа $G(m)$ назовем цветом графа.*

Очевидно, что граф однозначно определяется своим цветом и лексикографический порядок цветов графов определяет линейный порядок на множестве графов вида $G(u)$ при $u \in [m]$.

Определение 7. *Минимальный элемент этого множества называется* ***структурным графом*** *монома $[m]$ и обозначается $\mathbf{G}_0([m])$.*

Определение 8. ***Канонической формой*** *$\hat{G}$ цветного графа G называется граф минимального цвета среди всех графов, получающихся из G перенумерацией вершин.*

Таким образом структурный граф монома $[m]$ является канонической формой графа $G(m)$ при $m \in [m]$.

Сверточные выражения соответствующие структурному графу получаются друг из друга автоморфизмами структурного графа (естественно сохраняющими раскраску вершин и ребер), то есть перестановками сомножителей в исходном выражении, и перестановками имен индексов суммирования с одинаковыми сигнатурами. Таким образом группа “автоморфизмов монома” является прямым произведением группы автоморфизмов его структурного графа и группы перестановок имен индексов с одинаковой сигнатурой.

Нумерации, приводящие граф G к канонической форме $\hat{G}$ определены с точностью до автоморфизмов графа $\hat{G}$. Алгоритм вычисления перенумераций вершин цветного графа легко получается из соответствующего алгоритма для не цветных графов, приведенного в нашей предыдущей работе [1]. В нем меняется только определение сигнатур.

Каноническая форма выражения $m \in [m]$ является средним арифметическим всех выражений из $[m]$, граф которых имеет каноническую форму. То есть результатом усреднения любого выражения $v \in [m]$, граф которого имеет каноническую форму по группе автоморфизмов монома $[m]$. Суммирование естественно следует ввести с учетом линейных соотношений для символов, входящих в выражение и законов операций сложения и умножения. При суммировании каждый символ перед перемножением следует приводить к каноническому виду, а затем раскрывать скобки.

4. ЗАКЛЮЧЕНИЕ

Представленная работа является развитием идеи определения канонического представления выражения с индексами с использованием автоморфизмов цветного графа, строящегося по определенным правилам в случае когда индексы выражения могут иметь различный тип.

В работе дано определение цветного графа и способ его построения для произвольного выражения с цветными индексами. Разобран способ приведения выражения с индексами к каноническому виду, сводящийся к вычислению нумераций вершин соответствующего ему цветного графа, приводящих этот граф к каноническому виду. Предложенный способ позволяет последовательно учесть все виды соотношений между элементарными выражениями с абстрактными индексами, включая линейные соотношения общего вида. Причем объем перебора в рамках предложенного подхода существенно сокращается, так как на каждом этапе необходимо делать перебор не по всему групповому пространству форм индексного выражения, а только по группе автоморфизмов, связанных с цветным графом.

В настоящее время имеется экспериментальная программная реализация алгоритмов на языке Python, на которой отрабатываются предложенные методы. В дальнейшем предполагается сделать реализацию этих методов для наиболее популярных систем компьютерной алгебры таких как MAPLE [9] и REDUCE [10].

Авторы выражают благодарность В.А. Ильину за плодотворные обсуждения, С.А. Абрамову за стимулирование исследований и возможность выступить на семинаре по компьютерной алгебре [11].

Работа была выполнена в рамках Государственного задания № 115041410196.

СПИСОК ЛИТЕРАТУРЫ

1. *Shpiz G., Kryukov A.* Canonical Representation of Polynomial Expressions with Indices // Programming and Computer Software. 2019. V. 45. P. 81–87. https://doi.org/10.1134/S0361768819020105
2. *Korol'kova A.V., Kulyabov D.S., Sevast'yanov L.A.* Tensor Computations in Computer Algebra Systems // Programming and Computer Software. 2012. V. 39. P. 135–142.
3. *Malcolm A.H.* MacCallum. Computer algebra in gravity research // Living Reviews in Relativity. 2018. V. 21. P. 6. https://doi.org/10.1007/s41114-018-0015-6
4. *Li H., Li Zh., Li Y.* Riemann Tensor Polynomial Canonicalization by Graph Algebra Extension // Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation. July, 2017. P. 269–276. https://doi.org/10.1145/3087604.3087625
5. *Portugal R.* An algorithm to simplify tensor expressions // Comput. Phys. Commun. 1998. V. 115. P. 215–230.
6. *Steinwachs Ch.F.* Combinatorial aspects in the one-loop renormalization of higher derivative theories // ArXiv: 1909:00810, 2019. 24 p. https://arxiv.org/abs/1909.00810
7. *Rodionov A.Y., Taranov A.Y.* Combinatorial aspects of simplification of algebraic expressions // In Proc. Int. Conf. EUROCAL'89, Lecture Notes in Computer Science. 1989. V. 378. P. 192–201.
8. *Ilyin V.A., Kryukov A.P.* ATENSOR – REDUCE program for tensor simplification // Computer Physics Communications 1996. V. 96. P. 36–52.
9. Maplesoft. https://www.maplesoft.com/
10. *Hearn A.C. and Schöpf R.* REDUCE User's Manual, Free Version. https://reduce-algebra.sourceforge.io/manual/manual.html
11. *Абрамов С.А., Боголюбская А.А.* Семинар по компьютерной алгебре в 2016–2017 гг. // Программирование. 2018. № 2. С. 3–4; http://www.ccas.ru/sabramov/seminar/doku.php