

УДК 519.16:004.021

М.Ю. Перминова

Программный модуль получения явных выражений коэффициентов производящих функций, основанных на использовании композиции

Рассматривается программная реализация нового метода нахождения коэффициентов производящих функций. Описан разработанный автором программный модуль получения явных выражений коэффициентов композиции производящих функций в системе компьютерной алгебры Maxima, приводятся примеры его использования для решения конкретных задач.

Ключевые слова: программный модуль, система компьютерной алгебры, Maxima, производящие функции, коэффициенты композиции производящих функций.

doi: 10.21293/1818-0442-2017-20-1-65-69

Системы компьютерной алгебры (СКА), относящиеся к классу систем символьных вычислений, широко используются не только в инженерных и теоретических расчетах, математическом моделировании, но и в таких областях, как анализ звука и изображений [1–3]. Как правило, СКА востребованы математиками, исследователями и инженерами, активно применяющими математические методы. Одной из тенденций развития систем компьютерной алгебры является развитие преобразующих процедур и процедур управления преобразованиями [3], в том числе операций над производящими функциями. Производящие функции [4] являются эффективным инструментом решения математических задач комбинаторики, теории вероятности, математической физики и т.д. [5, 6]. В настоящее время появился новый метод нахождения явных выражений коэффициентов композиции производящих функций [7]. Коротко опишем данный метод. Для нахождения явных выражений коэффициентов производящих функций необходимо знать композиты производящих функций и операции над композитами. Композитой $A^\Delta(n, k)$ называется функция, являющаяся коэффициентом производящей функции $[A(x)^k] = \sum_{n \geq k} A^\Delta(n, k)x^n$. Существует множество

операций над композитами: сумма, произведение, сдвиг, обращение и др. Одной из таких операций является композиция.

Пусть даны две производящие функции $A(x)$ и $B(x)$. Тогда для вычисления композиции $F(x) = B(A(x))$ будет верно выражение

$$f(n) = \sum_{k=1}^n A^\Delta(n, k) \cdot b(k), \quad f(0) = b(0), \quad (1)$$

где $f(n)$ – явное выражение коэффициентов композиции производящих функций; $b(k)$ – коэффициенты производящей функции $B(x)$; $A^\Delta(n, k)$ – композита $A(x)$.

Обзор СКА показал, что получение явных выражений коэффициентов производящих функций

реализовано для простейших случаев [8]. В общем случае эта задача не решена. В данной статье предлагается вариант реализации программного модуля, основанного на автоматизации описанного выше метода.

Постановка задачи

Пусть дана производящая функция $F(x)$ на языке системы компьютерной алгебры.

Необходимо найти явное выражение коэффициентов $f(n)$ с использованием метода [7], алгоритма последовательной генерации ограниченных разбиений натурального числа [9], алгоритма декомпозиции полиномов, основанного на генерации разбиений [10, 11], и формулы композиции производящих функций (1).

Композиты производящих функций и операции над ними известны и находятся в базе знаний в виде некоторых обобщенных функций.

Для решения данной задачи в СКА разработан программный модуль, основанный на перечисленных выше методе и алгоритмах.

Структура программного модуля

Рассмотрим структуру разработанного программного модуля (рис. 1).

На вход блока синтаксического анализа поступает исходная производящая функция $F(x)$, соответствующая стандартному описанию функции на исходном языке. Если синтаксический анализ прошел успешно, то $F(x)$ поступает на вход блока семантического анализа. Иначе генерируется ошибка. Если семантический анализ $F(x)$ прошел успешно, то на вход блока анализа на наличие композиции полиномов поступает $F(x)$. Иначе генерируется ошибка.

Блок поиска композиции полиномов базируется на разработанных ранее алгоритме последовательной генерации ограниченных разбиений натурального числа [9] и алгоритме декомпозиции полиномов, основанном на генерации разбиений [10, 11]. Если композиция полиномов найдена, то список с полиномами $[A(x), B(x)]$ передается на вход блока получения явного вида выражений коэффициентов $f(n)$. В случае неудачи $F(x)$ поступает на вход блока

поиска композиции других классов производящих функций.

Блок поиска композиции других классов производящих функций основан на представлении $F(x)$ синтаксическим деревом в виде списка, составлении списка подвыражений $F(x)$ и их анализе. Если подобная композиция найдена, то на вход блока полу-

чения $f(n)$ поступает список производящих функций $[A(x), B(x)]$. Иначе генерируется ошибка.

По схеме на рис. 1 видно, что два блока взаимодействуют с базой композит и операций, в которой находятся производящие функции с их композитами и операции над ними.

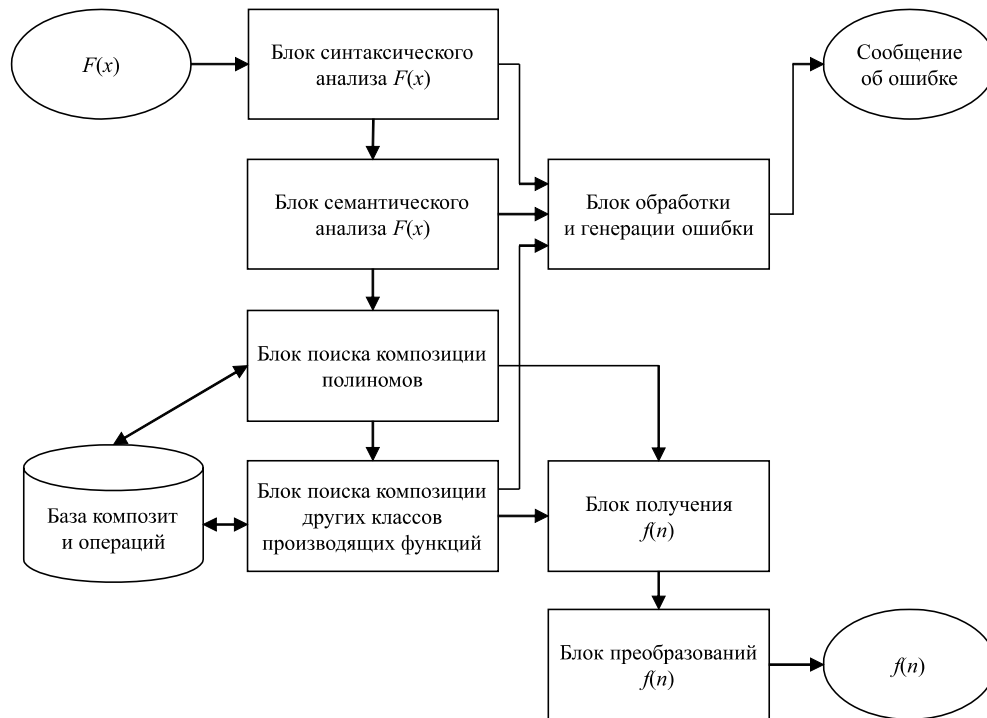


Рис. 1. Структура модуля получения явных выражений коэффициентов композиции производящих функций

Программная реализация

Для программной реализации модуля выбрана СКА *Maxima*. *Maxima* – это система компьютерной алгебры, аналогичная *Mathematica*, *Mathlab* и другим подобным системам. Несмотря на сложность программных продуктов, предназначенных для символьных вычислений, *Maxima* выгодно отличается от аналогичных программ тем, что является бесплатной и распространяется с открытым исходным кодом. Кроме того, *Maxima* переносима почти на все основные операционные системы и платформы, даже на карманные компьютеры [12].

Программный модуль получения явных выражений коэффициентов композиции производящих функций состоит из двух частей: *Decomposition* и *Lib*. Для корректной работы модуля обе его части необходимо подключить отдельными командами в *Maxima*.

Lib представляет собой базу композит и операций, где находятся производящие функции с их композитами и операции над ними. Производящие функции в модуле разделены по классам: полиномы; рациональные производящие функции; производящие функции логарифма; радикалы; тригонометрические производящие функции; гиперболические производящие функции; производящие функции на основе e^x .

Формат записей в базе:

```
/* Comment */
```

```
Composita_name(p1,p2,...,pn) := formula;
```

где *Comment* – комментарий (производящая функция, композита которой будет записана), *Composita_name(p₁,p₂,...,p_n)* – название композиты (в названиях изменяется только часть «name») с указанием параметров p_1, p_2, \dots, p_n и далее формула композиты.

Примеры записей в базе *Lib*:

```
/* Композита f(x)=x+2*x^2+x^3 */
```

```
Composita_x2x2x3(n,k):=binomial(2*k,n-k);
```

```
/* Композита f(x)=ln(1+x) */
```

```
Composita_ln(n,k):=stirling1(n,k)*k!/n!;
```

```
/* Композита f(x)=sinh(x) */
```

```
Composita_sinh(n,k):=
```

```
1/2^k*sum((-1)^i*binomial(k,i)*(k-2*i)^n/n!,i,0,k);
```

Записи в базе упорядочены в соответствии с монографией [7]. Всего в базе насчитывается более 100 композит производящих функций различных классов.

В *Lib* есть обобщенные и частные формулы композит производящих функций. Например, $ax+bx^2+cx^3$ – обобщенная формула, $x+2x^2+x^3$ – частная формула.

На основе базы композит и операций *Lib* построены два списка: *LIST_GF* – список производящих функций, *LIST_CO* – список композит произво-

дящих функций. Порядок элементов $LIST_GF_i$ строго соответствует порядку элементов $LIST_CO_i$. Поиск производящих функций осуществляется по списку $LIST_GF$. Производящие функции в этом списке упорядочены по количеству параметров p_n , необходимых для получения композиты производящей функции.

$Decomposition$ содержит в себе функции всех оставшихся блоков структуры (см. рис. 1). Эти функции вызываются главной функцией части $Decomposition - AllFunc(f)$, где f – исходная производящая функция в синтаксисе $Maxima$. Остановимся подробнее на программной реализации оставшихся блоков.

Синтаксический анализ $F(x)$ осуществляется стандартными функциями $Maxima$. Существуют следующие ограничения для $F(x)$:

- функция F должна зависеть только от одной переменной x ;
- если $F(x)$ – экспоненциальная производящая функция, то ее необходимо записывать как $e^{(expression)}$. Если $expression$ – это переменная, то скобки можно опустить;
- $F(x)$ должна представлять собой композицию двух производящих функций $A(x)$ и $B(x)$, и их композиты должны быть записаны в Lib ;
- все коэффициенты $F(x)$ должны быть записаны в числовом виде.

Семантический анализ $F(x)$ представляет собой разложение $F(x)$ в ряд Тейлора и проверку нулевого коэффициента. Это осуществляется стандартной функцией $Maxima - taylor(F(x),x,0,0)$.

В блоке поиска композиции полиномов главной функцией является $GetDecompositionPoly(F(x))$, она вызывает следующие функции:

1) $SearchDegrees(F(x))$ – формирует список, содержащий все возможные сочетания степеней m и s полиномов $A(x)$ и $B(x)$ соответственно, при которых возможно представление $F(x)$ в виде композиции $B(A(x))$. Для каждого сочетания степеней вызывается функция $GetT(m,s)$;

2) $GetT(m,s)$ – формирует список номеров уравнений T , которые необходимо решить для получения коэффициентов полиномов композиции. Для каждого уравнения T_j вызываются функции:

- $First(m,s,T_j)$ – находит путь от корня (m,s,T_j) к самому левому листу дерева разбиений;
- $NexT(m,s,T_j)$ – находит следующий путь в поддереве дерева разбиений;
- $GetPartition(L)$ – формирует разбиение P , где L – путь в дереве разбиений;
- $GetMonom(P)$ – формирует моном по разбиению P ;
- $GetEquation(Poly)$ – формирует уравнение, решение которого представляет собой коэффициенты полиномов $A(x)$ и $B(x)$.

Более подробно ознакомиться с алгоритмом работы функции $GetDecompositionPoly(F(x))$ можно в статьях [9–11].

Блок поиска композиции других классов производящих функций включает в себя следующие функции:

1) $ListSubExp(F(x))$ – формирует список подвыражений D для $F(x)$ на основе синтаксического дерева, представленного списком;

2) $ComposYavn(D)$ – по списку D проверяет, является ли $B(x)$ в композиции $F(x)=B(A(x))$ тригонометрической, гиперболической, логарифмической функцией или функцией квадратного корня;

3) $GetListSubexps(F(x))$ – возвращает список с производящей функцией, полученной после замены повторяющихся в ней подвыражений, и самим подвыражением. Вызывает функции:

- $RepeatSubExp(D)$ – возвращает список R повторяющихся подвыражений $F(x)$;
- $Replace(F(x),R)$ – заменяет повторяющиеся подвыражения в производящей функции $F(x)$.

Для связи с базой предназначена функция $GetNumbsInBase(F(x))$. Она формирует список с номерами позиций производящих функций из базы, с которыми сходна $F(x)$. Как видно из схемы (см. рис. 1), данная функция содержится в блоках поиска композиции различных классов производящих функций.

Случается, что для производящей функции по разным формулам в базе можно найти несколько композит. Например, для полинома $2x + 4x^2 + 3x^3$ будет две композиты:

$$\sum_{j=0}^k \binom{j}{n-k-j} 2^{k-j} 4^{-n+k+2j} 3^{n-k-j} \quad \text{– по формуле}$$

для производящей функции вида $cx^3 + bx^2 + ax$;

$$j = \left\lfloor \frac{3k-n}{2} \right\rfloor \binom{j}{n-3k+2j} 3^{k-j} \binom{k}{j} 2^{-n+3k-j} 4^{n-3k+2j} \quad \text{–}$$

по формуле для производящей функции вида $ax + bx^2 + px^m$.

Поэтому для выбора одной формулы коэффициентов вызывается функция $ChooseForm(V)$, где V – список из формул коэффициентов. Она возвращает самую простую формулу с наименьшим количеством переменных.

Блок получения явного выражения коэффициентов композиции производящих функций $f(n)$ содержит функцию $Superposition(A(x),B(x))$. В основе работы данной функции лежит формула композиции производящих функций, о которой говорилось ранее. Формулу коэффициентов $b(k)$ получаем из композиты производящей функции $B(x)$ (степень $k=1$).

На данный момент преобразования $f(n)$ выполняются стандартными функциями $Maxima$.

Пример работы модуля

Опишем подключение и работу с программным модулем получения закрытого вида выражений коэффициентов композиции производящих функций в системе компьютерной алгебры $Maxima$ (рис. 2).

Сначала объявляется переменная `file_search_maxima`. Она указывает списки каталогов для поиска файлов функциями `load`, `demo` и некоторыми другими функциями `Maxima`. В нашем случае к каталогам, которые заданы в `Maxima` по умолчанию, добавляется один пользовательский каталог, путь которого указан в квадратных скобках. Далее

```
(%i1) file_search_maxima: append (file_search_maxima,
["C:/work/mary/maxima/###.mac"]);

(%i2) load(Lib);
(%o2) C:/work/mary/maxima/Lib.mac

(%i3) load(Decomposition);
(%o3) C:/work/mary/maxima/Decomposition.mac

(%i4) AllFunc(sin(x)+sin(x)^2);
```

$$\sum_{k=1}^n \frac{\binom{1}{k-1} ((-1)^{n-k+1}) \sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} (2i-k)^n \binom{k}{i} (-1)^{\frac{n+k}{2}-i}}{2^k n!}$$

```
(%o4)
```

Рис. 2. Получение закрытого вида выражений коэффициентов композиции производящих функций в `Maxima`

Для тестирования программного модуля был разработан набор тестовых примеров. Они основаны на следующих производящих функциях: тригонометрические – $\sin(x)$, $\tan(x)$; гиперболическая – $\sinh(x)$; полиномы степеней от 2 до 6; экспоненциальные – $e^x - 1$ и $x \cdot e^x$; рациональные – $x/\sqrt{1-x^2}$ и $-x/(x-1)$. Примеры производящих функций из тестового набора: $\sin(x) + \sin(x)^2$, $-4x^6 - 24x^5 - 47x^4 - 28x^3 + 5x^2 + 2x$, $\tan(\sinh(x))$, $(x + 3x^2 + x^3) / \sqrt{1 - (x + 3x^2 + x^3)^2}$. Результаты численных экспериментов на наборе тестовых примеров показали работоспособность программного модуля.

Заключение

Разработанный программный модуль позволяет в автоматизированном режиме решать задачи комбинаторики, алгебры, теории чисел, статистики, математической физики и пр. Это обеспечивает ускорение и уменьшение числа ошибок. Программный модуль расширяет возможности `Maxima` по нахождению явных выражений коэффициентов производящих функций. Полученное с использованием программного модуля явное выражение коэффициентов производящей функции позволяет расширить базу энциклопедии целочисленных последовательностей OEIS [13]. Данными этой энциклопедии пользуются как любители, так и специалисты в математике, комбинаторике, теории чисел, теории игр, физике, хи-

функцией `load` загружаются обе части программного модуля: `Lib` и `Decomposition`. Затем вызывается главная функция `AllFunc(F(x))`, где $F(x)$ – производящая функция в синтаксисе `Maxima`. Результатом работы `AllFunc(F(x))` является закрытый вид выражений коэффициентов $f(n)$.

мии, биологии, информатике. В дальнейшем можно дополнить существующую базу композит; разработать алгоритмы добавления и удаления записей в базе; улучшить работу алгоритмов модуля; разработать аналогичные модули для получения явных выражений коэффициентов производящих функций, например, представленных суммой или произведением производящих функций; адаптировать разработанные модули для различных систем компьютерной алгебры: `Mathematica`, `Maple`, `Sage` и др.

Литература

1. Joel S. Cohen. Computer algebra and symbolic computation. Mathematical Methods [Электронный ресурс]. – Режим доступа: [http://www.ukma.edu.ua/~yubod/teach/compalgebra/%5BJoel_S._Cohen%5D_Computer_algebra_and_symbolic_comp\(BookFi.org\).pdf](http://www.ukma.edu.ua/~yubod/teach/compalgebra/%5BJoel_S._Cohen%5D_Computer_algebra_and_symbolic_comp(BookFi.org).pdf), свободный (дата обращения: 16.03.2017).
2. Афонский А.А., Дьяконов В.П. Электронные измерения в нанотехнологиях и микроэлектронике [Электронный ресурс]. – Режим доступа: <https://www.litres.ru/vladimir-dyakov-2/elektronnye-izmereniya-v-nanotehnologiyah-i-mikroelektronike-6599233/>, свободный (дата обращения: 19.03.2017).
3. Клименко В.П., Ляхов А.Л., Гвоздик Д.Н. Современные особенности развития систем компьютерной алгебры [Электронный ресурс]. – Режим доступа: <http://cyberleninka.ru/article/n/sovremennye-osobennosti-razvitiya-sistem-kompyuternoy-algebry>, свободный (дата обращения: 19.03.2017).
4. Ландо С.К. Лекции о производящих функциях / С.К. Ландо. – 3-е изд., испр. – М.: МЦНМО, 2007. – 144 с.
5. Stanley R.P. Enumerative Combinatorics. – Cambridge University Press, 2001. – Vol. 2. – 600 p.

6. Wilf H.S. Generatingfunctionology / H.S. Wilf. – Academic Press, 1994. – 231 p.

7. Кручинин В.В. Степени производящих функций и их применение / В.В. Кручинин, Д.В. Кручинин. – Томск: ТУСУР, 2013. – 236 с.

8. Перминова М.Ю. Обзор систем компьютерной алгебры для работы с производящими функциями: матер. Всерос. науч.-техн. конф. студентов, аспирантов и молодых ученых «Научная сессия ТУСУР–2013». – Томск: В-Спектр, 2013. – Ч. 2. – С. 323–325.

9. Перминова М.Ю. Алгоритмы рекурсивной генерации ограниченных разбиений натурального числа / М.Ю. Перминова, В.В. Кручинин // Доклады ТУСУРа. – 2014. – № 4(34). – С. 89–94.

10. Перминова М.Ю. Алгоритм декомпозиции полиномов, основанный на разбиениях / М.Ю. Перминова, В.В. Кручинин, Д.В. Кручинин // Доклады ТУСУРа. – 2015. – № 4(38). – С. 102–107.

11. Перминова М.Ю. Анализ алгоритма декомпозиции полиномов, основанного на разбиениях [Электронный ресурс] / М.Ю. Перминова // Кибернетика и программирование. – 2015. – № 6. – С. 21–34. – Режим доступа: http://e-notabene.ru/kp/article_17169.html, свободный (дата обращения: 03.02.2017).

12. Решение прикладных задач с помощью Maxima [Электронный ресурс]. – Режим доступа: http://lug.mephist.ru/wiki/Решение_прикладных_задач_с_помощью_Maxima, свободный (дата обращения: 19.01.2017).

13. Sloane J.A. The On-Line Encyclopedia of Integer Sequences [Электронный ресурс]. – Режим доступа: <https://oeis.org/>, свободный (дата обращения: 16.03.2017).

Перминова Мария Юрьевна

Аспирант каф. промышленной электроники (ПрЭ) ТУСУРа

Тел.: +7 (382-2) 70-15-53

Эл. почта: pmy@2i.tusur.ru

Perminova M.Yu.

Software module to obtain explicit expressions for generating functions coefficients based on use of composition

The article investigates the software implementation of a new method to find the coefficients of generating functions, describes the developed software module used to obtain explicit expressions for composition of generating functions coefficients in the computer algebra system Maxima and its structure, and gives examples of solving specific problems.

Keywords: software module, computer algebra system, Maxima, generating functions, coefficients of the generating functions composition.