К. М. ТУЛЕНБАЕВ, ДАСТАН ІЛИЯС, НУРЖАН ЖУНУСОВ

(Университет им. Сулеймана Демиреля)

АЛГОРИТМ БУХБЕРГЕРА
ДЛЯ БИКОММУТАТИВНЫХ АЛГЕБР

Аннотация

В статье описывается алгоритм Бухбергера [3] построения редуцированного базиса Гребнера для биком-мутативных алгебр. Алгебру А называют бикоммутативной алгеброй, если выполняются два тождества: a*(b*c)= b*(a*c) и (a*b)*c= (a*c)*b. В частности, базис Гребнера может быть применен для решения системы полиномиальных уравнений от нескольких переменных. В статье [2] доказано, что для любой свободной конечнопорожденой бикоммутативной алгебры выполняется теорема Гильберта о базисе [1]. Ниже мы рассмотрим последовательность действий сразу с реализацией алгоритма Бухбергера и начнем с того, что договоримся, как мы будем представлять основные структуры. Представление и операции с полиномами В общем случае алгоритм Бухбергера может быть применен для полиномов от произвольного конечного числа переменных.

Ключевые слова: алгоритм Бухбергера, бикоммутативные алгебры, базиса Гребнера.

Кілт сөздер: Бухбергер алгоритмі бикоммутативті алгебра Гребнер базисі.

Keywords: Buchberger algorithm, Bicommutative algebra, Gröbner basis.

Разберем случай двух переменных. Отдельный полином удобно представлять в виде двумер-ного массива его коэффициентов **TPolynom**. Коэффициенты в массиве хранятся так, что индексы **i** и **j** массива – это степени переменных **x** и **y** в членах полинома. Такое представление полинома удобно тем, что умножение полинома на **x** или **y** в этом случае равносильно сдвигу массива вниз или вправо. С другой стороны, нам понадобится отдельный член полинома, который мы будем представлять записью **TMonom**, содержащей коэффициент и степени при переменных **x** и **y**. При таком представлении сложение полинома и отдельного члена (также как и вычитание) сводится к сложению коэффициентов.

Немного сложнее реализуется умножение полинома на отдельный член. Важно с самого начала определиться с порядком следования членов в полиноме. Дело в том, что далее нам понадобится сравнивать старшие члены двух полиномов, а это сравнение возможно только при заданном по-рядке следования. Кстати, от выбранного порядка следования зависит получаемый базис Гребнера. Выберем лексико-графический порядок для бикоммутативных алгебр, описанный в статье [2] – следования членов полинома (lex). Теперь мы можем написать процедуру, которая будет воз-вращать старший член полинома – она ищет в нашей матрице первый ненулевой коэффициент перебирая матрицу снизу вверх и в каждой строке справа налево. ***S-полином*** от двух полиномов **f** и **g** (обозначается как **S(f, g)**) специально "сконструирован≥ для сокращения старших членов двух полиномов. Не будем здесь приводить формулу для S-полинома, а просто покажем на примере, тем более что для объяснения формулы потребуется несколько страниц, а смысл очень простой. Пусть, например, есть два полинома:

$$\mathbf{f = x^3y^2 - x^2y^3 + x}$$
$$\mathbf{g = 3x^4y + y^2}$$

Умножим первый из них на $\mathbf{3x}$, а второй на $\mathbf{y}$, получим

$$\mathbf{f_1 = 3x^4y^2 - 3x^3y^3 + 3x^2}$$
$$\mathbf{g_1 = 3x^4y^2 + y^3}$$

Теперь вычтем из первого полинома второй и получим

$$\mathbf{f_1 - g_1 = -\,3x^3y^3 + 3x^2 - y^3}$$

Это и есть S-полином от полиномов **f** и **g**.

Настала пора определиться, что же такое базис Гребнера. Грубо говоря (и нам достаточно такого определения), это множество многочленов. Будем представлять его как простой массив полиномов list и их количество count в списке (**TGBasis**). Еще одно определение – отношение делимости двух членов полинома. Член **m** полинома делит член **k** полинома, когда степени всех переменных члена **m** меньше или равны соответствующих степеней члена **k** . Например, $\mathbf{2xy}$ делит $\mathbf{3x^2y}$ (получаем $\mathbf{3x/2}$), а $\mathbf{3x^2y}$ не делит $\mathbf{2xy}$.

Перейдем собственно к редукции. Редукция полинома по базису – это последовательное вычи-тание старших членов полинома, которые делятся на старшие члены полиномов, входящих в базис. Процедура **Reduce** выполняет редукцию полинома F по текущему базису. Алгоритм следующий: В начале процедуры результат **R** - полином, не содержащий ни одного члена. Для каждого полинома из базиса **F** проверяем, делит ли его старший член (**F.LT**) – старший член (**P.LT**) исходного полинома P. Если делит, полином **F** умножаем на частное от этого деления (**m**) и вычитаем из полинома **P**. Цикл в этом случае начинаем снова с первого полинома из базиса. Если не делит, берем следующий полином из базиса и выполняем деление для него. Когда мы прошли все полиномы базиса и деления не было, старший член исходного полинома **P** переносим в результат **R**. С новым полиномом **P** повторяем всю процедуру. Действия выполняются до тех пор, пока полином **P** содержит хотя бы один член. Если после редукции получается полином, не содержащий ни одного члена, говорят, что полином редуцируется к нулю по базису.

Посмотрим выполнение алгоритма на примере. Пусть базис состоит из двух полиномов:

$$\mathbf{y^2 - 1}$$
$$\mathbf{xy - 1}$$

Выполним редукцию полинома $x^2y + xy^2 + y^2$ по этому базису. Берем старший член первого полинома y^2. Он не делит старший член исходного полинома, поэтому переходим ко второму полиному. Старший член второго полинома xy делит старший член исходного. Выполняем умножение на частное от этого деления и вычитание:

$$x^2y + xy^2 + y^2 - x(xy - 1) = xy^2 + x + y^2$$

Цикл начинаем сначала (с первого полинома). Теперь его старший член делит старший член xy^2. Выполняем еще одну операцию:

$$xy^2 + x + y^2 - x(y^2 - 1) = 2x + y^2$$

Теперь старший член $2x$ не делится ни одним старшим членом из базиса. Переносим старший член $2x$ в результат, а с остатком y^2 повторяем деление. Он делится на старший член первого полинома из базиса. Выполняем операцию:

$$y^2 - (y^2 - 1) = 1$$

Старший член 1 снова не делится ни одним старшим членом базиса. Переносим ее в результат и, так как остатка нет, завершаем выполнение. Итак, в результате редукции получили полином

$$2x + 1$$

Реализация данного алгоритма позволяет решать задачу о вхождении заданного элемента в заданный идеал свободной конечнопорожденной бикоммутативной алгебры.

ЛИТЕРАТУРА

1 Зарисский О., Самюэль Р. Коммутативная алгебра. – М.: ИЛ, 1963.

2 Dzhumadil’daev A. S., Tulenbaev K. M. Hilbert basis theorem for nonassociative algebras // Com. Algebra. 2013. – № 4 (To be published).

3 Бухбергер Б. и др. Компьютерная алгебра: символьные и алгебраические вычисления. – М.: Мир, 1986.

REFERENCES

1 Zarisskij O., Samjujel' R. Kommutativnaja algebra. – M.: IL, 1963.

2 Dzhumadil’daev A. S., Tulenbaev K. M. Hilbert basis theorem for nonassociative algebras // Com. Algebra. 2013. – № 4 (To be published).

3 Buhberger B. i dr. Komp'juternaja algebra: simvol'nye i algebraicheskie vychislenija. – M.: Mir, 1986.

Резюме

Түленбаев К. М., Дастан Іляс, Жүнісов Н.

(Сүлеймен Демирел атындағы университет)

БИКОММУТАТИВТІК АЛГЕБРА ҮШІН БУХБЕРГЕР АЛГОРИТМІ

Биокоммутативтік алгебра үшін Гребнердің ықшамдалған базисін құрудағы Бухбергер алгоритмі сипатталған. А алгербасын егер екі a*(b*c) = b*(a*c) және (a*b)*c = (a*c)*b ұқсастығы орындалса бикоммутативтік алгебра деп атайды. Негізінен Гребнер базисі бірнеше айнымалының көптеген теңдеулері жүйесінің үшін қолданылуы мүмкін. Мақала кез келген бикоммутативті алгебраның еркін соңғы нәтижесі Гильберттің базис туралы теоремасымен орындалатыны дәлелденген. Төменде біз бірден Бухбергер алгоритмінің жүзеге асырылу әрекетінің реттілігін қарастырамыз, оның негізгі құрылымды қалай түсінетінімізден бастаймыз. Жалпы алғанда Бухбергер алгоритмі полиномдар үшін айнымалылардың еркін соңғы саны үшін қолданылуы мүмкін.

Кілт сөздер: Бухбергер алгоритмі бикоммутативті алгебра Гребнер базисі.

Summary

K. M. Tulenbayev, Dastan Ilias, Nurzhan Zhunusov

(Suleyman Demirel University)

BUCHBERGER ALGORITHM FOR BICOMMUTATIVE ALGEBRAS

This article describes the Buchberger algorithm [3] A reduced Gröbner basis for bicommutative algebra. Bicommutative algebra A is called an algebra if the following two identities: a * (b * c) = b * (a * c) and (a * b) * c = = (a * c) * b. In particular, the Gröbner basis can be applied to solve a system of polynomial equations in several variables. In [2] it is proved that for any free finite bicommutative algebra theorem holds Hilbert basis [1]. Below we describe the sequence of actions immediately with the implementation of Buchberger's algorithm and start with the fact that we agree will represent the main structure. Presentation and operations with polynomials in general Buchberger algorithm can be applied to polynomials of arbitrary finite number of variables.

Keywords: Buchberger algorithm, Bicommutative algebra, Gröbner basis.

Поступила 25.04.2013г.