

Преобразование базиса Грёбнера нульмерного идеала к иному мономиальноальному упорядочению.

Федоров Глеб М33351

Октябрь 2020

1 Оглавление

1.1 Постановка проблемы

1.2 Дополнительная теория

1.3 Алгоритм FGLM

1.4 Примерная архитектура программы

1.5 Используемая литература

2 Постановка проблемы

Дан базис Грёбнера нульмерного идеала, построенный на мономиальном упорядочении m_1 . Привести данный базис к иному мономиальному упорядочению m_2 .

3 Дополнительная теория

3.1 Исключающий идеал

Определение: Пусть дан $I = \langle f_1, \dots, f_n \rangle \in F[x_1, \dots, x_n]$. Тогда l -м исключающим идеалом I_l называется идеал в $F[x_{l+1}, \dots, x_n]$, равный $I \cap F[x_{l+1}, \dots, x_n]$.

Теорема(об исключении): Пусть $I \subset F[x_1, x_2, \dots, x_s]$ - идеал и G - его базис Грёбнера по отношению к лех-упорядочению с $x_1 \rangle x_2 \rangle \dots \rangle x_n$. Тогда $\forall l : 0 \leq l \leq n$ множество

$$G_l = G \cap F[x_1, x_2, \dots, x_s]$$

является базисом Грёбнера l -го исключающего идеала I_l .

Доказательство: Зафиксируем l в интервале $(0, n)$. Так как $G_l \in I_l$ по построению, достаточно показать, что

$$\langle LT(I_l) \rangle = \langle LT(G_l) \rangle$$

Включение в одну сторону очевидно ($\langle LT(G_l) \rangle \subset \langle LT(I_l) \rangle$ по построению). Докажем, что $\langle LT(I_l) \rangle \subset \langle LT(G_l) \rangle$. Для этого достаточно показать что $LT(f)$, где $f \in I_l$, делится на некоторый $g \in G_l$. Заметим, что $f \in I$, то есть $LT(f)$ делится на $LT(g)$ для некоторого g (т.к. G является базисом Грёбнера идеала I). Так как $f \in I_l$, то $LT(g)$ содержит только переменные x_{l+1}, \dots, x_n . Так как используется лех-упорядочение с $x_1 \rangle x_2 \rangle \dots \rangle x_n$, то любой моном, содержащий хотя бы одну из переменных x_1, \dots, x_l , больше всех мономов из $F[x_{l+1}, \dots, x_n]$. Значит $g \in G_l$, что и требовалось доказать.

3.2 Соответствие идеала и многообразия

Определение: Пусть $I \in F[x_1, \dots, x_n]$ - некоторый идеал. Положим

$$V(I) = (a_1, \dots, a_n) \in F^n : f(a_1, \dots, a_n) = 0 \forall f \in I$$

Теорема: $V(I)$ является аффинным многообразием. В частности, если $I = \langle f_1, \dots, f_n \rangle$, то $V(I) = V(f_1, \dots, f_n)$

Доказательство: По теореме Гильберта о базисе идеал I конечно порождён, $I = \langle f_1, \dots, f_n \rangle$. Покажем, что $V(I) = V(f_1, \dots, f_n)$. Если $f(a_1, \dots, a_n) = 0$ для всех полиномов $f \in I$, то $f_i(a_1, \dots, a_n) = 0$ (так как $f_i \in I$). Следовательно, $V(I) \subseteq V(f_1, \dots, f_n)$. С другой стороны, пусть $(a_1, \dots, a_n) \in V(f_1, \dots, f_n)$, и пусть $f \in I$. Так как $I = \langle f_1, \dots, f_n \rangle$, то

$$f = \sum_{i=1}^s h_i f_i$$

для некоторых $h_i \in F[x_1, \dots, x_n]$. Но тогда

$$f(a_1, \dots, a_n) = \sum_{i=1}^s h_i(a_1, \dots, a_n) f_i(a_1, \dots, a_n) = \sum_{i=1}^s h_i(a_1, \dots, a_n) * 0 = 0$$

Следовательно, $V(f_1, \dots, f_n) \subseteq V(I)$, а значит эти два идеала равны.

3.3 Нульмерный идеал

Определение: Пусть A - это алгебра над полем K и J - двусторонний идеал в A . Рассматривая алгебру A как факторкольцо A/J , которое можно превратить в алгебру над K , если определить в ней умножение на элементы поля K следующим образом:

$$k(a + J) = ka + J, \forall k \in K, \forall a \in A$$

Построенная таким образом алгебра называется факторалгеброй алгебры A по идеалу J .

Теорема: Пусть поле F алгебраически замкнуто и $I \in F[x_1, x_2, \dots, x_n]$. Тогда следующие условия эквивалентны:

1. Алгебра $A = F[x_1, x_2, \dots, x_n]/I$ конечномерна над F .
2. $V(I) \subset F^n$ конечно.
3. Если G - базис Грёбнера идеала I , то

$$\forall i \exists m_i \geq 0 : x_i^{m_i} = LM(g)$$

для некоторого $g \in G$.

4. Для каждой переменной x_i исключаящий идеал $I \cap F[x_1, x_2, \dots, x_n]$ является ненулевым.

Идеал, удовлетворяющий данной теореме называется нульмерным

3.4 Нормальная форма полинома

Определение(1.1): Пусть G - базис Грёбнера идеала I . Будем называть $f \in F[x_1, \dots, x_n]$ редуцированным по отношению к G (или в нормальной форме по отношению к G), если не существует $g \in G$ старший член которого делит какие-либо члены из f .

Определение: Алгоритм редукции - алгоритм, вычисляющий нормальную форму по полинома f .

Определение: Базис Грёбнера G называется редуцированным, если $\forall g \in G$ g - редуцирован, по отношению к другим элементам G .

Утверждение: Данное определение редуцированного базиса Грёбнера совпадает с определением из лекции с точностью до нормировки старшего члена.

Определение: Пусть I - нульмерный идеал над $F[x_1, \dots, x_n]$ и G - его редуцированный базис Грёбнера. Натуральным базисом, определяемым G K -векторного пространства R/I , назовем базис $B(G)$, элементами которого являются мономы, редуцированные по отношению к G . Будем обозначать $D(I)$ размерность K -векторного пространства R/I (степень идеала).

Определение: Пусть $B(G)$ - натуральный базис для R/I . Тогда положим, что

$$M(G) = \{x_i b \mid b \in B(G), 1 \leq i \leq n, x_i b \notin B(G)\}$$

граница G .

Теорема(2.1): Пусть I - нульмерный идеал и G - редуцированный базис Грёбнера данного идеала, и $B(G)$ - натуральный базис Грёбнера для R/I , тогда $\forall m \in M(G)$ выполнено одно из данных условий:

1. $\forall x_i : x_i \mid m$ выполнено, что $m/x_i \in B(G)$, тогда и только тогда, когда m - это старший моном элементов G .
2. $m = x_j m_k$ для некоторых j и $m_k \in M(G)$

Доказательство:

1. Следует из определения редуцированного базиса Грёбнера и определения $B(G)$
2. Пусть x_j делит m и $m/x_j \notin B(G)$, тогда $m_k = m/x_j \in M(G)$. Заметим, по определению $M(G)$, что $m = x_j m_k = x_i b$. Тогда $i \neq j$ (если $i = j$, то $m_k = b$, что противоречит тому, что $m_k \in M(G)$) и $m_k/x_i = b/x_j \in B(G)$, так как $B(G)$, по определению, замкнуто относительно деления. Таким образом $m_k = x_i(b/x_j) \in M(G)$.

Следствие: Пусть k число образующих редуцированного базиса Грёбнера для нульмерного идеала. Тогда $k \leq nD(I)$

4 Алгоритм FGLM

4.1 Вычисление нормальной формы

Воспользуемся структурой векторного пространства, чтобы построить алгоритм, который найдёт координаты нормальных форм элементов R/I за полиномиальное время. Для этого рассмотрим n -линейные отображения ϕ_i определённые на $B(G)$ следующим образом:

$$\phi_i : m \rightarrow NormalForm(x_i m)$$

и изучим их свойства.

Определение: Обозначим $T(G) = (t_{ijk})$ матрицу $n \times D(I) \times D(I)$, в которой $t_{ijk} = j$ -ая координата относительно $B(G)$ редукции элементов из G $x_i b_k (b_k \in B(G))$.

Данная матрица будет использована для вычисления нормальной формы полинома.

Теорема: $T(G)$ вычисляется за $O(nD(I)^3)$

Доказательство: Рассмотрим $MB(G) = B(G) \cup M(G)$ и упорядочим его элементы согласно m_1 . Будем строить столбцы t_{i*k} в том порядке, в котором $x_i b_k$ появляется в $MB(G)$. Рассмотрим $m = x_i b_k$. Если $m \in B(G)$, значит, что m - не редуцирован по отношению к G и другие $t_{ijk} = 0$ для $j \neq k$, и $t_{ikk} = 1$. В ином случае $m \in B(G)$, тогда, согласно теореме(2.1), либо $\exists g \in G : g = m + \sum_{u=1}^{D(I)} a_u b_u$, тогда $t_{i*k} = (-a_1, \dots, -a_{D(I)})^t$, либо $m = x_l m'$, где $m' \in M(G)$ и $m' < m$. Во втором случае, координаты $m' = x_s b_h$ по отношению к $B(G)$, уже вычислены, и хранятся в t_{i*s} . Этого достаточно, чтобы добавить координаты $x_l b_v b_v \in B(G)$ умноженные на соответствующий коэффициент, то есть $x_i b_k = x_l x_s b_h = x_l \sum_v t_{svh} b_v = \sum_u \sum_v t_{svh} t_{luv} b_u$. В этом случае мы должны выполнить $D(I)^2$ операций чтобы посчитать t_{i*k} , и, в итоге, данную операцию мы будем повторять $nD(I)$ раз.

Замечание: Функция *NextMonom* возвращает следующий моном, который необходимо рассмотреть. О том, как эта функция работает будет сказано позднее, но пока будем считать что она выполняется за $O(1)$.

Замечание: Для $i \in \{1, \dots, n\}$ матрица, связанная с ϕ относительно $B(G)$, есть t_{i**} .

Данный алгоритм вычисляет конструкцию, описанную на предыдущей странице.

Algorithm 1 Matphi

```

1: Input
2:    $m_1$  ▷ мономиальное упорядочение
3:   Basis ▷ минимальный редуцированный базис Грёбнера нульмерного идеала
4: EndInput

5: Output
6:    $\phi[i, m, m']$  for  $i = 1, \dots, n$  ▷  $m, m' \in B(G)$  такие, что  $\phi[i, *, *]$  - это матрица применений
    $p \rightarrow NormalForm(x_i p)$ , где  $p$  - редуцированные многочлены
7: EndOutput

8: Subfunctions
9:   NextMonom ▷ Возвращает и удаляет первый элемент списка ListOfNexts. Возвращает nil, если список
   пуст.
10:  InsertNexts(monom) ▷ Добавляет monom в ListOfNexts и сортирует его, согласно  $m_1$ 
11: EndSubfunctions

12: LocalVariables
13:   ListOfNexts ▷ Список "следующих"мономов. Отсортирован относительно  $m_1$ 
14: EndLocalVariables

15:  $monom := 1$ ,  $ListOfNexts := []$ 
16: while  $monom \neq nil$  do
17:   if monom является произведением старших мономов каких-то элементов из Basis then
18:     let  $monom = x_j m$ , где  $m$  - редуцируемо относительно Basis
19:      $NormalForm[monom] := \sum \lambda_i * NormalForm(x_j m_i)$ 
20:     for all  $k$ , таких что  $monom = x_k m'$ , где  $m'$  - нередуцируемо относительно Basis do
21:        $\phi[k, m'', m'] :=$  коэффициент  $m''$  в  $NormalForm[monom]$ 
22:   else if monom - старший моном какого-то  $p \in Basis$  then
23:      $NormalForm[monom] := -rest(p)$ 
24:     for all  $j$ , таких, что  $monom = x_j m$  do
25:        $\phi[j, m', m''] :=$  коэффициент  $m'$  в  $NormalForm[monom]$ 
26:   else
27:      $NormalForm[monom] := monom$ 
28:      $InsertNexts(monom)$ 
29:     for all  $j$ , таких, что  $monom = x_j m$  do
30:

$$\phi[j, m', m] := \begin{cases} 1 & \text{if } m' = monom \\ 0 & \text{otherwise} \end{cases}$$

 $monom := NextMonom$ 

```

Доказательство корректности: Корректность алгоритма следует из доказательства теоремы(2.1). Данный алгоритм представляет собой рассмотрение случаев, рассмотренных в доказательстве данной теоремы и случая, когда полином является элементом $B(G)$.

4.2 Смена упорядочения

Теорема(2.2): Пусть I - нульмерный идеал и (G_1, m_1) - его редуцированный базис Грёбнера, построенный для мономиального упорядочения m_1 , m_2 - иное мономиальное упорядочение. Тогда можно построить базис Грёбнера (G_2, m_2) за $O(nD(I)^3)$.

Доказательство: Из (G_1, m_1) мы можем построить $B(G_1) = \{a_1, \dots, a_{D(I)}\}$, $M(G_1)$ и $T(G_1)$, как было показано в предыдущей главе. Построим матрицу C в которой i -й столбец будет координатами элемента $b_i \in B(G_2)$ относительно $B(G_1)$. Будем строить новый базис итеративно. Пусть $B(G_2) := \{1\}$ и $M(G_2) := \emptyset$. Пусть $m = \min_{m_2} \{x_j b_i | 1 \leq j \leq n, b_i \in B(G_2), x_j b_i \notin B(G_2) \cup M(G_2)\}$. Может возникнуть три случая:

1. m - старший терм g , какого-либо $g \in G_2$
2. m нужно добавить в $B(G_2)$
3. m нужно добавить в $M(G_2)$, но m - кратно $LT(G)$ для некоторого $g \in G$.

Проверка того, что m удовлетворяет третьему пункту - старший член g строго меньше чем m при любом допустимом упорядочении и g уже добавлен в $M(G_2)$. Поскольку по построению, $m = x_j b_i$ мы можем посчитать его координаты $c(m)_h$ относительно $B(G_1)$ используя матрицы C и $T(G_1) = (t_{ijk})$:

$$m = x_j b_i = x_j * \sum_k c_{ki} * a_k = \sum_k x_j * c_{ki} * a_k = \sum_k c_{ki} * \sum_h t_{jkh} * a_h = \sum_h \left(\sum_k t_{jkh} c_{ki} \right) = \sum_h c(m)_h a_h$$

Если вектор $c(m)$ линейно независим от векторов из C , то выполняется второй пункт, и мы нашли новый $g \in B(G_2)$. В противном случае, мы получаем элемент $g \in G_2$.

Algorithm 2 NewBasis

```
1: Input
2:    $m_2$  ▷ новое упорядочение
3:   oldBasis ▷ Базис Грёбнера нульмерного идеала относительно упорядочения  $m_1$ 
4: EndInput

5: Output
6:   newBasis ▷ Базис Грёбнера нульмерного идеала относительно упорядочения  $m_2$ 
7: EndOutput

8: Subfunctions
9:   NormalForm(polynom) ▷ Возвращает редуцированную форму полиному относительно упорядочения  $m_1$ 
10:  NextMonom() ▷ Возвращает первый элемент ListOfNexts или nil, если ListOfNexts пуст. Первый элемент удаляется
11:  InsertNexts(monom) ▷ Добавляет в ListOfNexts произведение монома со всеми переменными, после чего сортирует лист и удаляет дубли
12: EndSubfunctions

13: LocalVariables
14:   staircase ▷ Список ведущих мономов из NewBasis
15:   MBasis ▷ Список пар  $[a_i, b_i]$ , где  $[a_i]$  - список мономов в нормальной форме относительно нового базиса и  $b_i = NormalForm(a_i)$ , нормальная форма  $a_i$  относительно старого базиса.
16:   ListOfNexts ▷ Список "следующих" мономов. Отсортирован относительно  $m_1$ 
17: EndLocalVariables

18:  $MBasis := []$ ;  $staircase := []$ ;  $newBasis := []$ ;  $ListOfNexts := []$ ;  $monom := 1$ ;
19: while  $monom \neq nil$  do
20:   if  $monom$  не делится на какие-нибудь элементы из  $staircase$  then ▷ Проверка, это пункт 1 или 2
21:      $vector := NormalForm(monom)$  ▷
22:     if есть линейная зависимость  $vector + \sum_{v \in MBasis} \lambda_v second(v) = 0$  then
23:        $pol := monom + \sum_{v \in MBasis} \lambda_v first(v)$ 
24:        $newBasis := cons(pol, newBasis)$ 
25:        $staircase := cons(monom, staircase)$ 
26:     else
27:        $MBasis := cons([monom, vector], MBasis)$ 
28:       InsertNexts(monom)
29:    $monom := NextMonom$ 
```

Доказательство корректности:

Первое: Покажем, что *newBasis* - это базис Грёбнера того же идеала при упорядочении m_2 . Для этого покажем, что элементы из *MBasis* независимы по модулю идеала, порождаемого *oldBasis*. Пусть это не так тогда существует линейная комбинация элементов из *MBasis* $P = \sum \lambda_i * m_i$, такая, что $P \in \langle oldBasis \rangle$. Тогда

$$NormalForm(P) = \sum \lambda_i NormalForm(m_i) = 0$$

...

Второе: Покажем что алгоритм завершается. Заметим, что дан нульмерный идеал. Это значит, что максимальное число линейно независимых нередуцируемых конечно, и равно отношению размерности кольца многочленов к размерности идеала, то есть числу нередуцируемых одночленов для любого базиса Грёбнера. Таким образом, число итераций, которые увеличивают *MBasis* конечно. После того, как все элементы *MBasis* рассмотрены, *ListOfNexts* не будет больше увеличиваться, значит, количество оставшихся итераций конечно.

Реализация *NextMonom*: При вставке одночлена в *ListOfNexts* полезно помнить, что он получается как произведение переменной на одночлен с известной нормальной формой. При этом один и тот же моном можно получить несколько раз...

5 Примерная архитектура программы

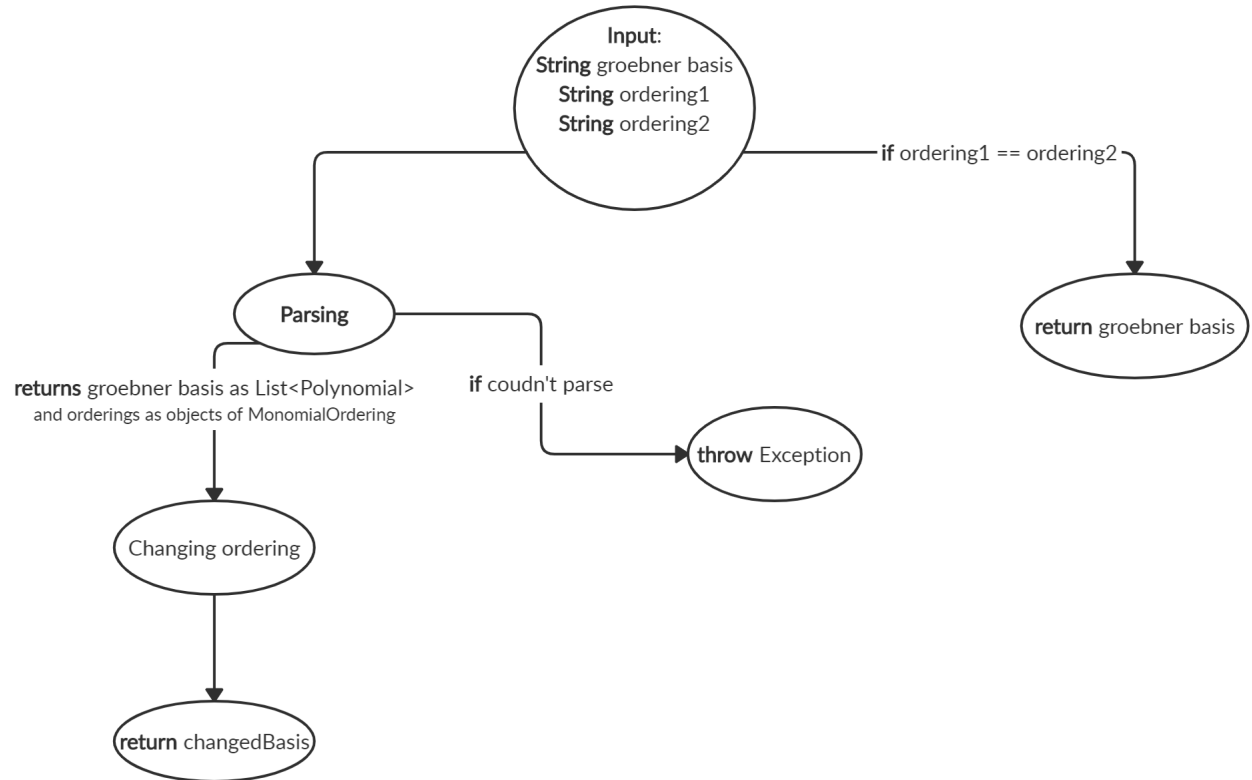


Рис. 1: Схема работы программы

6 Используемая литература

1. Faugère, J.C., Gianni, P., Lazard, D., Mora, T. Efficient computation of zero-dimensional gröbner bases by change of ordering (1993) Journal of Symbolic Computation, 16 (4), pp. 329-344.
2. <http://halgebra.math.msu.su/groebner.pdf>
3. Кокс Д., Литтл Дж., О'Ши Д. Идеалы, многообразия, кольца. Стр. 108. Стр. 153.
4. Презентация про FGLM алгоритм