

Реферат

Данная работа посвящена разработке ряда методов для ускорения вычисления базиса Гребнера. Рассмотрены стратегии выбора S-полиномов в алгоритме Бухбергера. Разработан эффективный алгоритм вычисления нормальной формы полинома.

Ключевые слова: Моном, полином нескольких переменных, базис Гребнера, редукция полиномов, нормальная форма полинома, алгоритм Бухбергера.

Оглавление:

1. Введение

1.1 Определения и теория базисов Гребнера

1.2 Алгоритм Бухбергера

1.3 Важность задачи

2. Метод предсказания результата редукции с помощью многомерной диаграммы Юнга.

2.1 Диаграмма Юнга

2.2 Результаты

2.2.1 Результаты достоверности результатов редукций на основе диаграмм Юнга, относительно известного базиса Гребнера.

2.2.2 Результаты работы Алгоритма Бухбергера, использующего стратегию выбора s -полиномов, основанную на предсказании по диаграммам Юнга.

3. Эффективное вычисление нормальной формы.

3.1 Описание Алгоритма

3.2 Алгоритм 3

3.3 Результаты тестирования Алгоритма 3

4. Особенности программной реализации

1. Введение

Системы нелинейных алгебраических полиномиальных уравнений часто возникают в различных областях науки и техники: в построении формул, проблеме пересечения поверхностей в геометрии, в обратной кинематической задаче.

Численное решение систем полиномиальных уравнений сопряжено с рядом проблем: погрешности вычислений, невозможность распознать и тем более решить недоопределенные системы, возможная потеря корня.

Этих проблем можно избежать, если использовать символьные или гибридные численно-символьные методы. Для решения многих задач, описываемых полиномиальными системами уравнений, часто применяется метод базисов Гребнера.

Обычно для вычисления базиса Гребнера используют алгоритм Бухбергера [1], который, в худшем случае, имеет экспоненциальную сложность. Однако есть и другие способы. Например, вычисление инволютивных базисов, с помощью которых можно как решать ряд задач систем алгебраических полиномиальных уравнений, так и вычислить собственно базис Гребнера. Также Фужером был разработан наиболее быстрый, на сегодня, метод вычисления базиса Гребнера не использующий технику Бухбергера. Данная работа посвящена исследованию редукции полиномов и вычислению Базиса гребнера алгоритмом Бухбергера. Были рассмотрены следующие задачи:

- 1) Разработка методов предсказания редукции полиномов.
- 2) Разработка эффективного алгоритма редукции

1.1 Определения и теория базисов Гребнера.

Определение 1.1.1: *Мономом* от переменных x_1, \dots, x_n называется произведение вида

$$x_1^{\alpha_1} \dots x_n^{\alpha_n},$$

где показатели степеней $\alpha_1, \dots, \alpha_n$ – неотрицательные целые числа.

Будем использовать следующие обозначения для мономов. Пусть $\alpha = (\alpha_1, \dots, \alpha_n)$ – набор n неотрицательных целых чисел. Положим

$$x^\alpha = x_1^{\alpha_1} \dots x_n^{\alpha_n}$$

Через $|\alpha| = \alpha_1 + \dots + \alpha_n$ мы будем обозначать полную степень монома x^α .

Определение 1.1.2: *Полиномом* f от переменных x_1, \dots, x_n с коэффициентами из K называется конечная линейная комбинация мономов (с коэффициентами из K). Полином f будем записывать в виде

$$f = \sum_{\alpha} a_{\alpha} x^{\alpha}, a_{\alpha} \in K,$$

где суммирование идет по конечному множеству наборов $\alpha = (\alpha_1, \dots, \alpha_n)$. Множество всех полиномов от переменных x_1, \dots, x_n с коэффициентами из K обозначается $K[x_1, \dots, x_n]$.

Определение 1.1.3: *мономиальным упорядочением* на $K[x_1, \dots, x_n]$ называется любое бинарное отношение $>$ на $Z_{\geq 0}^n$, обладающее следующими свойствами:

- 1) $>$ является линейным упорядочением на $Z_{\geq 0}^n$
- 2) Если $\alpha > \beta$ и $\gamma \in Z_{\geq 0}^n$, то $\alpha + \gamma > \beta + \gamma$;
- 3) $>$ вполне упорядочивает $Z_{\geq 0}^n$, т.е. любое непустое подмножество в $Z_{\geq 0}^n$ имеет минимальный (наименьший) элемент (по отношению к упорядочению $>$).

Рассмотрим три наиболее часто употребляемых на практике упорядочения:

- 1) Лексикографическое упорядочение:

$x^{\alpha} \succ_{Lex} x^{\beta}$ если имеется $1 \leq k \leq n$, такой, что $\alpha_1 = \beta_1, \dots, \alpha_{k-1} = \beta_{k-1}, \alpha_k > \beta_k$

- 2) Упорядочение сначала по полной степени, а затем лексикографическое:

$x^\alpha \succ_{\text{DegLex}} x^\beta$ если $\sum_i \alpha_i > \sum_i \beta_i$ или $\sum_i \alpha_i = \sum_i \beta_i$ но имеется $1 \leq k \leq n$, такой, что $\alpha_1 = \beta_1, \dots, \alpha_{k-1} = \beta_{k-1}, \alpha_k > \beta_k$

3) Упорядочение сначала по полной степени, а затем обратное лексикографическое:

$x^\alpha \succ_{\text{DegRevLex}} x^\beta$ если $\sum_i \alpha_i > \sum_i \beta_i$ или $\sum_i \alpha_i = \sum_i \beta_i$ но имеется $1 \leq k \leq n$, такой, что $\alpha_n = \beta_n, \dots, \alpha_{k+1} = \beta_{k+1}, \alpha_k < \beta_k$

Определение 1.1.4: Подмножество $I \subset K[x_1, \dots, x_n]$ называется *идеалом*, если выполнены следующие условия:

- 1) $0 \in I$;
- 2) если $f, g \in I$, то $f + g \in I$;
- 3) если $f \in I$ и $h \in K[x_1, \dots, x_n]$, то $hf \in I$.

Определение 1.1.5: Пусть $f = \sum_{\alpha} a_{\alpha} x^{\alpha}, a_{\alpha} \in K$ -- ненулевой полином, и пусть $>$ - мономиальное упорядочение.

1) *Мультистепень* полинома f определяется как:

$$\text{multideg}(f) = \max (\alpha \in \mathbb{Z}_{\geq 0}^n : a_{\alpha} \neq 0)$$

(максимум берется по отношению к $>$)

2) *Старший коэффициент* полинома f – это

$$\text{LC}(f) = a_{\text{multideg}(f)} \in K$$

3) *Старший моном* полинома f – это

$$\text{LM}(f) = x^{\text{multideg}(f)}$$

(с коэффициентом 1).

4) *Старший член* полинома f – это

$$\text{LT}(f) = \text{LC}(f) * \text{LM}(f).$$

Определение 1.1.6: говорят, что *полином f редуцируется полиномом g к полиному f'* , если $f' = f - gu$, где u – моном и $\text{LT}(gu)$ совпадает с одним из

мономов f . Если полученный таким образом f' снова может быть редуцирован полиномом g , то можно получить полином $f'' = f' - u'g$. Этот процесс может продолжаться пока мы не получим полином r , нередуцируемый полиномом g . Такой полином будем называть *нормальной формой полинома f по модулю g* , и записывать как $r = NF(f, g)$.

Определение 1.1.7: *редукция* по конечному множеству полиномов $I \subset K[x_1, \dots, x_n]$ определяется как последовательная редукция по всем элементам F . Редуцируя f пока это возможно, получим нормальную форму по модулю F . $r = NF(f, F)$.

Определение 1.1.8: пусть $I \subset K[x_1, \dots, x_n]$ – ненулевой идеал.

1) обозначим через $LT(I)$ множество старших членов элементов из I , т.е.

$$LT(I) = \{ cx^\alpha : \exists f \in I, LT(f) = cx^\alpha \}.$$

2) Обозначим через $\langle LT(I) \rangle$ идеал, порожденный элементами из $LT(I)$.

Определение 1.1.9: Пусть задано мономиальное упорядочение. Конечное подмножество $G = \{g_1, \dots, g_s\}$ элементов идеала I называется его *базисом Гребнера* (или стандартным базисом), если

$$\langle LT(g_1), \dots, LT(g_s) \rangle = \langle LT(I) \rangle.$$

1.2 Алгоритм Бухбергера.

Определение 1.2.1: *S-полиномом* $p, q \in K[x_1, \dots, x_n]$ называется комбинация

$$Spol(p, q) = \frac{LCM(p, q)}{LT(p)} p - \frac{LCM(p, q)}{LT(q)} q$$

Где $LCM(p, q)$ – наименьшее общее кратное $LT(p)$ и $LT(q)$.

Алгоритм Бухбергера для построения базиса Гребнера основывается на теореме, гласящей, что G является базисом Гребнера тогда и только тогда, когда

$$NF(\text{Spol}(g_i, g_j), G) = 0, \forall g_i, g_j \in G$$

Приведем описание алгоритма построения базиса Гребнера, для полиномиального идеала, с использованием так называемых критериев, применение которых может привести к значительному сокращению

Алгоритм 2.2:

In: F – конечное множество полиномов

Out: G – базис Гребнера для $I(F)$

```

1:   $G = F$ 
2:   $B = \{(f_i, f_j), \text{ где } f_i, f_j \in G, i \neq j\}$ 
3:  while ( $B \neq \emptyset$ )
4:       $(f_1, f_2) \leftarrow B$ 
5:      if (Критерий1 ложен и Критерий2 ложен)
6:           $h = \text{Spol}(f_1, f_2)$ 
7:           $r = NF(h, G)$ 
8:          if ( $r \neq 0$ )
9:               $G = G \cup \{r\}$ 
10:         endif
11:     endif
12: endwhile
13: return  $G$ 

```

Определение 1.2.3: *Критерий1* позволяет исключить из рассмотрения каждую пару полиномов f_i, f_j , для которой в процессе работы алгоритма уже оказались рассмотренными две другие пары f_i, f_k и f_k, f_j такие что $LM(f_k)$ делит $LCM(LM(f_i), LM(f_j))$

Определение 1.2.4: *Критерий2* исключает из рассмотрения пары такие, что

$$LM(f_i)LM(f_j) = LCM(LM(f_i), LM(f_j));$$

Применение критериев 1 и 2 позволяет существенно повысить эффективность алгоритма, поскольку, во многих случаях, позволяет распознать редуцируемость S-полинома, соответствующего данной паре полиномов f_i, f_j к нулю без вычисления соответствующей нормальной формы. Корректность алгоритма и окончание его работы были доказаны Бухбергером.

Однако нетрудно придумать примеры идеалов, когда даже наилучшие современные версии алгоритма затрачивают колоссальное время и ресурсы памяти для вычисления базиса Гребнера. Этому есть объективные причины:

- 1) Степени промежуточных полиномов, генерируемых в процессе вычисления, могут быть очень велики
- 2) Коэффициенты элементов базиса Гребнера могут быть очень сложными рациональными числами, даже если исходные коэффициенты – небольшие целые числа.

Много усилий было потрачено, чтобы теоретически оценить верхнюю границу степеней промежуточных полиномов в зависимости от степеней исходных элементов. На сегодня, для алгоритма Бухбергера, известна оценка сложности алгоритма $O(d^{2^n})$, где d – наибольшая степень полиномов, поступивших на вход алгоритма, а n – число переменных.

1.3 Важность задачи

Применения Базиса Гребнера упрощает решение таких задач как:

- 1) Задача принадлежности к идеалу
- 2) Решение полиномиальных уравнений
- 3) Задача нахождения неявного представления

Рассмотрим некоторые свойства базисов Гребнера.

Свойство 1.3.1: пусть $G = \{g_1, \dots, g_s\}$ – базис Гребнера идеала $I \subset K[x_1, \dots, x_n]$, и пусть $f \in K[x_1, \dots, x_n]$. Тогда существует единственный полином $r \in K[x_1, \dots, x_n]$, который обладает следующими свойствами:

- 1) ни один член полинома r не делится ни на один из старших членов $LT(g_1), \dots, LT(g_s)$;
- 2) существует $g \in I$, что $f = g + r$.

То есть r является остатком от деления f на G , не зависящим от порядка делителей G .

Следствие 1.3.2: пусть $G = \{g_1, \dots, g_s\}$ – базис Гребнера идеала $I \subset K[x_1, \dots, x_n]$, и пусть $f \in K[x_1, \dots, x_n]$. Тогда $f \in I$ в том и только в том случае, когда остаток от деления f на G равен нулю.

Теорема 1.3.3: Любой идеал обладает базисом Гребнера относительно любого допустимого упорядочения.

Теорема 1.3.4: Два идеала равны тогда и только тогда, когда они имеют один и тот же редуцированный базис Гребнера.

Теорема 1.3.5: Система полиномиальных уравнений несовместна тогда и только тогда, когда базис Гребнера (относительно любого упорядочения) содержит константу.

Базис Гребнера дает соответствующий алгоритмический инструмент для анализа основных алгебраических свойств систем полиномов. Лексикографический базис Гребнера имеет треугольную форму

относительно вхождения переменных. Это является наиболее удобным видом для вычисления корней численными и символьными методами. Вычисляемый, относительно любого упорядочения, базис Гребнера выдает информацию относительно совместности системы и размерности пространства решений. Следовательно базис Гребнера можно рассматривать как некое каноническое представление системы полиномов.

Известно, что более 90% времени алгоритм Бухбергера вычисляет редукции S -полиномов, при этом большая часть всех редукций равна 0. Именно поэтому так существенно влияют на алгоритм критерии. Однако кроме точных критериев возможен другой подход: стратегии выбора следующей пары из множества B (см алгоритм). Так если есть метод, позволяющий с некоторой долей вероятности определить результат редукции, то можно сначала брать преимущественно ненулевые редукции, после чего большая часть отложенных редукций будет пропущена уже на этапе проверки критериев. Так например Бухбергер в своей работе [1] предложил, что выгоднее выбирать такую пару $(i,j) \in B$, для которой моном $LCM(LT(f_i), LT(f_j))$ минимален. Соответствующий S -полином будет чаще выдавать ненулевые остатки (т.е. новые элементы базиса Гребнера) в начале работы алгоритма; следовательно, повышается вероятность того, что затем остатки $\overline{Spol(f_i, f_j)}^G$ будут нулевыми. Эта так называемая стратегия нормального выбора подробно рассмотрена в работах [2,3]. Наконец, еще есть сахарная стратегия, которая является усовершенствованием стратегии нормального выбора. Сахарная и ее вариант двойная сахарная стратегии рассматриваются в работе [4]. Также в работе [4] в качестве стратегии использовать модулярные методы: производить редукцию полиномов над полем остатков по некому модулю p . Очевидно, что если редукция в поле $\mathbb{Z}_p[x_1, \dots, x_n]$ не ноль, то и редукция в $K[x_1, \dots, x_n]$ также не ноль. Однако обратное не верно.

2. Метод предсказания результата редукции с помощью многомерной диаграммы Юнга.

Определение 2: *Диаграммой Юнга*, соответствующей показателю $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ называется «лестница» из n ступенек, у которой высота k -ой ступеньки равна α_k , а ширина — единице.

$$(4, 1, 1) \mapsto \begin{array}{|c|c|} \hline \square & \\ \hline \square & \square \\ \hline \square & \square \\ \hline \end{array}$$

$$(4, 2, 1, 0) \mapsto \begin{array}{|c|c|c|} \hline \square & \square & \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array}$$

2.1 Многомерная диаграмма Юнга

Определение 2.1.1: *образом* монома x^α в n -мерной целочисленной решетке Z_+^n назовем точку $\alpha = (\alpha_1, \dots, \alpha_n)$.

Определение 2.1.2: *Многомерной диаграммой Юнга* монома $x^\alpha = x^{\alpha_1} \dots x^{\alpha_n}$ назовем множество точек D целочисленной решетки Z_+^n таких, что:

$$\beta = (\beta_1, \dots, \beta_n) \in D(x^\alpha) \Leftrightarrow \forall i = 1 \dots n \beta_i \leq \alpha_i$$

Рассмотрим моном $x^3 y^4$:

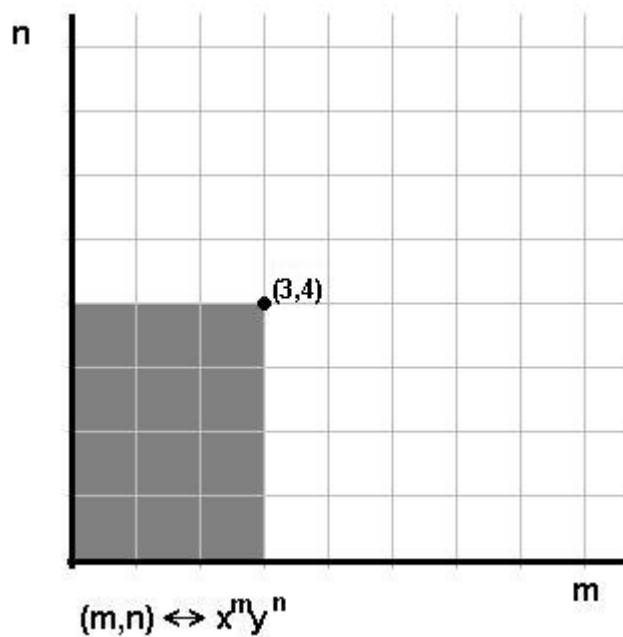


Рисунок 1

На рисунке 1 изображена диаграмма Юнга (двумерная) для монома x^3y^4 . На плоскости mn в точках с неотрицательными целочисленными координатами располагаются мономы $x^m y^n$. Как видно мономы, образы которых попадают в отмеченную фигуру, делят моном x^3y^4 .

Определение 2.1.3: *многомерной диаграммой Юнга* полинома $f = \sum_{\alpha} a_{\alpha} x^{\alpha}, a_{\alpha} \in K$ назовем объединение диаграмм Юнга входящих в полином мономов. Саму диаграмму от f далее будем обозначать как $D(f)$.

В качестве примера рассмотрим полином $5x^5y^2 - x^3y^3 - 3x^3y + xy + 2xy^5$ и его диаграмму Юнга.

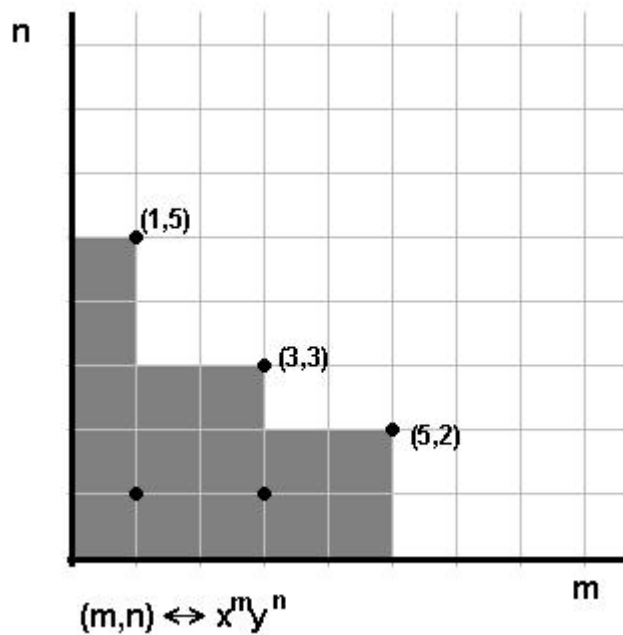


Рисунок 2

Определим операции над полиномами, использующие диаграммы Юнга.

Определение: $D(f_1 + \dots + f_k) = \bigcup_{i=1}^k D(f_i)$, т.е. в качестве диаграммы Юнга от суммы полиномов будем брать объединение диаграмм Юнга от всех слагаемых.

Несложно заметить, что при умножении полинома на моном, вся диаграмма сдвинется на вектор, определенный $\alpha = (\alpha_1, \dots, \alpha_n)$ – степень монома.

Разработаем метод предсказания результата редукции, основанный на диаграммах Юнга. Для начала рассмотрим стандартный алгоритм редукции полинома в $K[x_1, \dots, x_n]$ по конечному множеству F .

Алгоритм редукции:

In: $F = f_1, \dots, f_s$ – множество полиномов

f – редуцируемый полином

Out: r – остаток от деления f на F .

1: $p = f$

2: **while** ($p \neq 0$)

3: $\text{bdiv} = \text{false}$

```

4:      i=1
5:      while (i<=s)
6:          if ( LT(fi) делит LT(p)) then
7:              p=p - fi*(LT(p)/LT(fi))
8:              bdiv=true
9:          else
10:             i++
11:          endif
12:      endwhile
13:      if (bdiv == false)
14:          p = p- LT(p)
15:          r = r + LT(p)
16:      endif
17: endwhile
15: return r

```

Будем в алгоритме редукции оперировать не реальными полиномами с их коэффициентами, а Диаграммами Юнга. Когда у нас возникает возможность редукции одной диаграммы с помощью другой, мы сравниваем диаграммы, если они оказались равны, то считаем, что результат редукции 0, если не равны, то выполняем операцию суммы диаграмм. Поскольку мы точно знаем что на седьмом шаге алгоритма мы избавляемся в полиноме от старшего члена, а в случае с диаграммами такого сокращения не произойдет, то добавим на этом шаге метод `deleteLT(p)`, здесь `p` уже не полином, а структура данных, представляющая диаграмму Юнга, проще говоря набор ~~Сравнение~~ диаграмм перед их сложением, необходимо постольку, поскольку, вообще говоря, иначе редукция диаграмм никогда не окажется равной нулю.

3. Метод предсказания результата редукции с помощью операций с полиномами с булевыми коэффициентами.

У представления полиномов их диаграммами Юнга есть существенный недостаток: теряется информация о части мономов, что может сыграть существенную роль в предсказании результата редукции. Постараемся как-то это учесть.

Будем рассматривать полиномы с коэффициентами в булевой алгебре $\langle \{0,1\}; +, *, ^- \rangle$, в ней $1+1=1$, $1+0=1$, $1*0=0$, а дополнение 1 есть 0. Очевидно, что с такими коэффициентами, сокращений мономов не будет происходить вообще, что, кстати, является вполне естественным для полиномов общего вида — таких полиномов, в которых совпадения коэффициентов редки. Однако, есть один существенный недостаток: сложность самой редукции таких объектов. Так относительно честной редукции данный метод отличается только тем, что он не производит непосредственной работы с коэффициентами. В работе алгоритма редукции с подобными полиномами, будем удалять старший моном после шага 7 алгоритма редукции.

4. Результаты

Тестирование методов предсказания производилось следующим образом: генерировалось некоторое количество полиномов G , далее случайным образом несколько раз выбирались два полинома f и g из G , и в случае ненулевой редукции $r = \overline{\text{Spol}(f,g)}^G$, r добавлялось в множество G . Таким образом мы имели нечто похожее на промежуточное состояние базиса Гребнера, по ходу алгоритма Бухбергера. Для тестирования стратегий угадывания, из сгенерированного множества G снова выбиралось два полинома f и g , из них составлялся s -полином. Далее проводилась редукция

$\overline{\text{Spol}(f,g)}^G$ с помощью стандартного алгоритма и оценивались результаты редукции с помощью двух, описанных выше методов.

Тест 1: количество переменных 5. Мономов в полиноме от 1 до 4, степень вхождения переменной в моном от 0 до 2.

Процент угадывания нулевой редукции с помощью диаграмм: 55.5%

Процент угадывания нулевой редукции с помощью полином с булевыми коэффициентами: 37.2%

Тест 2: количество переменных 10. Мономов в полиноме от 1 до 4, степень вхождения переменной в моном от 0 до 2.

Процент угадывания нулевой редукции с помощью диаграмм: 40.6%

Процент угадывания нулевой редукции с помощью полином с булевыми коэффициентами: 14%

Тест 3: количество переменных 8. Мономов в полиноме от 1 до 4, степень вхождения переменной в моном от 0 до 2.

Процент угадывания нулевой редукции с помощью диаграмм: 25.6%

Процент угадывания нулевой редукции с помощью полином с булевыми коэффициентами: 9%

Тест 4: здесь был взят уже просчитанный базис Гребнера для `cuslic5`, и на вход подавались случайные полиномы.

Процент угадывания нулевой редукции с помощью диаграмм: 11.2%

Процент угадывания нулевой редукции с помощью полином с булевыми коэффициентами: 6.8%

5. Эффективное вычисление нормальной формы.

5.1 Описание Алгоритма

Поскольку наиболее трудоемкое место в алгоритме Бухбергера это вычисление редукций полиномов, то после уменьшения количества рассматриваемых редукций наиболее актуален вопрос об ускорении самой редукции. Кроме того задача вычисления редукции, при известном базисе Гребнера, сама по себе является значимой. Например, как было сказано выше, задача принадлежности идеалу, которая сводится к проверке редуцируемости полинома к нулю.

Пусть есть $G=\{g_1,\dots,g_s\}$ – базис Гребнера идеала $I \subset K[x_1,\dots,x_n]$. Причем считаем, что G – минимальный и редуцированный.

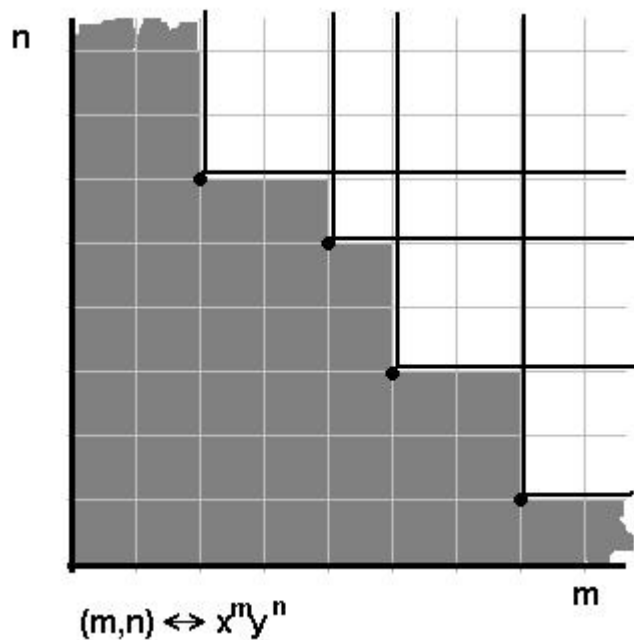


Рисунок 3

Рассмотрим Рисунок 3, черными точками на нем отмечены старшие члены элементов базиса Гребнера. В помеченной области располагаются все мономы x^α , для которых $NF(x^\alpha, G) = x^\alpha$, т.е. эти мономы уже являются редуцированными.

Тут следует заметить, что множество нередуцируемых мономов не всегда конечно.

Будем вычислять $NF(f,G)$ в фактор-алгебре $K[x_1, \dots, x_n]/G$. Целью нашей будет построить вычисления таким образом, чтобы в конце каждого шага, мы оставались в $K[x_1, \dots, x_n]/G$.

Известно что:

$$\text{если } c \in K, p \in K[x_1, \dots, x_n], \text{ и } r = NF(p,G), \text{ то } NF(c*p) = c*r \quad (3.0)$$

$$\text{если } m = x_i * m' \text{ и } p = NF(m',G), \text{ то } NF(m,G) = NF(x_i * p, G) \quad (3.1)$$

$$\text{если } f = p + q \text{ и } p' = NF(p,G), q' = NF(q,G) \text{ то } NF(f,G) = p' + q' \quad (3.2)$$

Воспользуемся эффективными методами вычисления полиномов, например схемой Горнера. Также возможно применение более изощренных схем, приводящих к уменьшению числа не скалярных умножений, но в этой работе мы ограничимся схемой Горнера. При этом схема вычисления полинома будет состоять из множества операций:

- 1) умножение полинома на переменную
– сложность этой операции будет рассмотрена ниже
- 2) скалярное умножение полинома на коэффициент
– простая операция, имеющая сложность k скалярных умножений, где k – количество мономов полинома.
- 3) сложение двух полиномов
– простая операция, в худшем случае имеет сложность k сложений коэффициентов + t слияний мономов, где k – меньшее количество мономов в полиномах, а t – количество мономов в результирующем полиноме (ограничено суммой количества мономов полиномов)

Замечание: то, что в сложении полиномов большую часть в сложность вносит сложение коэффициентов, следует из того, что по ходу вычислений редукций коэффициенты сильно растут и являются не целыми а дробными числами.

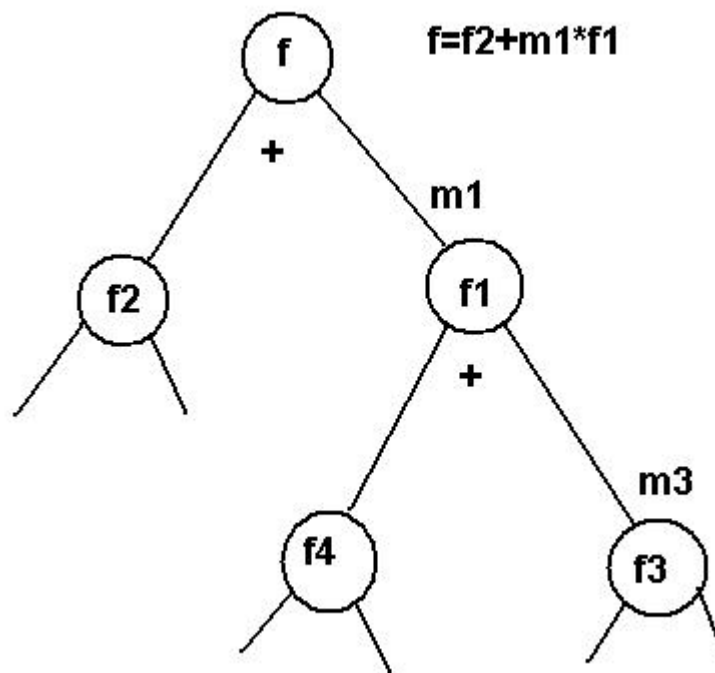


Рисунок 4

Итак у нас имеется процедура, позволяющая представить полином в виде дерева, листья которого – нередуцируемые мономы, а ребра, от родителя к правому сыну снабжены неким мономиальным коэффициентом m_i . Так что для вычисления значения родителя вычисляем по формуле:

$$fk = NF(f_{2k+2} + m_{2k+1} * f_{2k+1}, G) = f_{2k+2} + NF(m_{2k+1} * f_{2k+1}, G)$$

при этом моном m_{2k+1} представим в виде произведения переменных x_i, \dots, x_n , и будем несколько раз применять формулу 3.1.

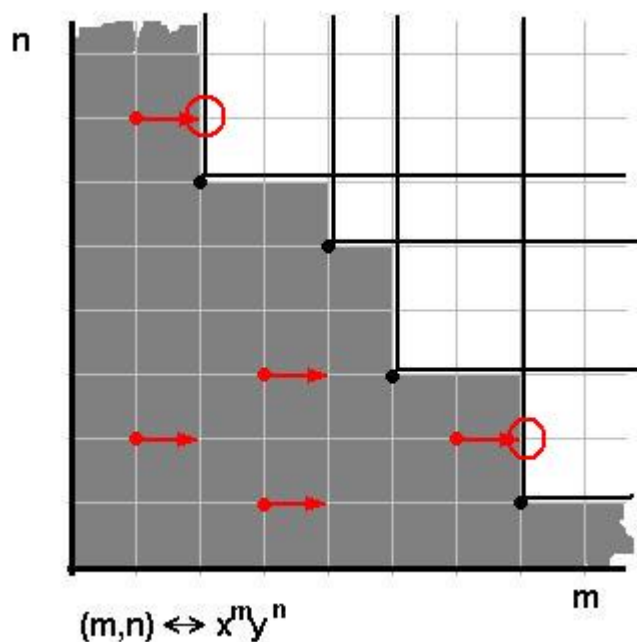


Рисунок 5

Пусть, на некотором шаге, у нас есть полином $f \in K[x_1, \dots, x_n]/G$, и нужно построить $NF(x*f, G)$. На рисунке 5 красными точками отмечены мономы полинома f , стрелками помечено, что произойдет после домножения на x . Как видим некоторые операции умножения монома на x в фактор-алгебре $K[x_1, \dots, x_n]/G$ будут тривиальными – моном по-прежнему останется редуцированным, но какая-то часть мономов, после домножения на x , может оказаться на границе области, т.е. выйти из $K[x_1, \dots, x_n]/G$, а значит потребуется проведение редукции. Пусть m_g – моном, оказавшийся на границе, мы вычисляем редукцию для m_g , и полученный результат сохраняем в кэш. И теперь, если на последующем шаге алгоритма нам вновь потребуется вычислять нормальную форму монома m_g , то достаточно взять предвычисленный результат из кэша. Таким образом в кэш попадут только мономы находящиеся на границе.

Замечание: даже из приведенной картинке видно, что максимальное количество кэшируемых таким образом нормальных форм иногда будет неограниченно, и действительное ограничение будет зависеть от степени исходного редуцируемого полинома.

Замечание: к сожалению, построение $NF(x_i * p, G)$, где $p \in K[x_1, \dots, x_n]/G$ не всегда является простой операцией, и, в некоторых случаях, произведение редукции по вышеизложенному алгоритму, по эффективности будет мало чем отличаться от стандартной редукции.

Определение 5.1.1: Если множество старших членов базиса Гребнера идеала I содержит в себе множество $\{x_1^{k_1}, \dots, x_n^{k_n}\}$, то такой идеал называется *нуль-мерным*.

Теперь, если мы нарисуем множество $K[x_1, \dots, x_n]/G$, где G – базис Гребнера нуль-мерного идеала, то, для случая 2х переменных, рисунок будем иметь подобную структуру:

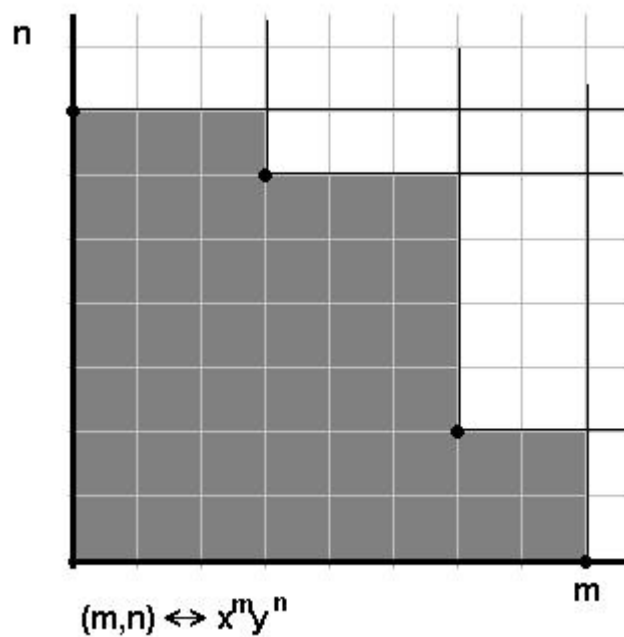


Рисунок 6

т.е носитель $K[x_1, \dots, x_n]/G$, будет конечным, а значит и использования алгоритма с кэшированием, предположительно, будет наиболее эффективно: количество кэшируемых элементов будет ограничено.

5.2 Алгоритм R

Приведем описание алгоритма на псевдокоде.

Основная функция – *optReduce*, рекурсивно строит древовидную схему вычисления полинома *p*, и вычисляет нормальную форму на каждом шаге.

Function: *optReduce*

In: Polynom *p* – редуцируемый полином

Out: Polynom *r* – нормальная форма *p*

```
1:  (p1,p2,m) <- CutPolynom(p)
2:  r1 = optReduce(p1)
3:  r2 = optReduce(p2)
4:  while (Deg(m) > 0)
5:    x = pickVariable(m)
6:    polynom tmp = 0
7:    foreach (monome mon in r2)
8:      tmp = tmp + GetNF(mon*x)
9:    r2 = tmp
10: endwhile
11: r2 = r2*m
12: r = r1 + r2
13: return r
```

Здесь функция *CutPolynom*(polynom *p*) разделяет полином на 2 полинома *p1* и *p2* и возвращает моном *m* такие что:

$$p = p1 + m \cdot p2.$$

Функция *pickVariable*(monome *m*) возвращает одну из переменных x_i , которая делит моном *m*, и присваивает моному *m* новое значение m/x_i .

Function *GetNF*

H – хэш предвычисленных нормальных форм мономов.

G – конечное множество полиномов, относительно которых вычисляется нормальная форма.

In: monome *m*

Out: Polynom p

```
1:  if (m  $\notin$  H)
2:    H[m] = Reduce(m, G)
3:  endif
4:  return H[m] * coefficient(m)
```

Здесь редукция монома вычисляется без учета коэффициента, т.к. в КЭШе хранятся нормальные формы мономов с коэффициентами 1.

Замечание: Поскольку операция проверки монома на нередуцируемость сопряжена с последовательной проверкой на делимость со всеми старшими мономами элементов множества G , в кэше хранятся не только элементы ‘границы’ мономов из $K[x_1, \dots, x_n]/G$, но и все, когда-либо запрашиваемые по ходу алгоритма, мономы из $K[x_1, \dots, x_n]/G$.

5.3 Результаты тестирования Алгоритма **R**

5.3.1 Сравнение алгоритма со стандартным алгоритмом редукции

Для тестирования алгоритма возьмем заранее просчитанный базис Гребнера. И будем подавать на вход стандартного алгоритма и алгоритма **R** полиномы состоящие из нескольких случайных мономов (под случайным мономом будем понимать моном с коэффициентов в некоем определенном диапазоне $[-a, a]$ и со степенями переменных также в некотором диапазоне $[0, \maxdeg]$).

Будем тестировать на примере лексикографического базиса Гребнера для системы cyclic5:

$$\begin{aligned} & x_1 + x_2 + x_3 + x_4 + x_5, \\ & x_1 * x_2 + x_2 * x_3 + x_3 * x_4 + x_4 * x_5 + x_5 * x_1, \\ & x_1 * x_2 * x_3 + x_2 * x_3 * x_4 + x_3 * x_4 * x_5 + x_4 * x_5 * x_1 + x_5 * x_1 * x_2, \end{aligned}$$

$$x_1 \cdot x_2 \cdot x_3 \cdot x_4 + x_2 \cdot x_3 \cdot x_4 \cdot x_5 + x_3 \cdot x_4 \cdot x_5 \cdot x_1 + x_4 \cdot x_5 \cdot x_1 \cdot x_2 + x_5 \cdot x_1 \cdot x_2 \cdot x_3, \\ x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 - 1;$$

Сам базис приведен в приложении. Как нетрудно увидеть, идеал порожденный системой `syslic5` является нуль-мерным.

Тестировать алгоритм **R** будем двумя способами. В одном случае кэшированные нормальные формы будут накапливаться алгоритмом **R**, очевидно что при этом алгоритм будет наиболее эффективен. Так, если считать число шагов редукции (за шаг редукции считаем операцию вычитания из редуцируемого полинома полинома множества редукции помноженного на моном – шаг 7 алгоритма редукции), то через несколько экспериментов это число для алгоритма **R** становится равным 0, т.е. алгоритм перестает считать редукции и занимается только вычислением полинома. А во втором, перед вычислением нормальной формы полинома, кэш будет стираться, данный вариант в таблице обозначен как **R***.

Параметры случайного полинома			Алгоритм R*	Алгоритм R	Стандартный алгоритм
Максимальная степень переменной	Число мономов	Интервал коэффициентов			
5	10	-1000...1000	355 мс	23 мс	188 мс
10	10	-1000...1000	600 мс	62 мс	16 389 мс
15	3	-10...10	430 мс	38 мс	80 720 мс
50	5	-1000...1000	779 мс	226 мс	---
100	5	-1000...1000	1103 мс	488 мс	---

5.3.2 Сравнение алгоритма **R** с пакетом Maple 9.5

Будем сравнивать по времени выполнения нахождения нормальной формы полинома относительно все того же лексикографического базиса cycle5. Для краткости будем писать результаты в виде AR=... sec, Maple=...sec.

1) $p = -5x_1^9x_3^{12}x_5^9 - 2x_1^2x_4^4 + 6x_2^5x_3^{12}x_4$ – полином степени 30.

AR=0.5 sec

Maple = 41.8 sec

2) $p = -5x_1^4 - 2x_1^2x_4^4 + 1x_1x_2^3x_4^3 + 4x_2^3x_4^3x_5 + 6x_3^2x_4x_5^2$ – полином степени 7.

AR = 0.25 sec

Maple=0.21 sec

3) $p = 4x_1^4x_3^3x_5 - 2x_1^4x_4^3 + 7x_2^4x_3^2 - 4x_2x_3x_5^4$ – полином степени 8.

AR = 0.33 sec

Maple = 0.51 sec

4) $p = 8x_1^{14}x_5^2 - 8x_2^8x_3^7x_4^7$ – полином степени 22.

AR = 0.562 sec

Maple = 53.8 sec

Эффективность Алгоритма R на высоких степенях очевидна.

6. Особенности программной реализации

Существует множество хороших и быстрых программных комплексов для решения задач компьютерной алгебры, и построение конкурентоспособного продукта – довольно трудная и объемная задача. Поэтому целью работы не было построение «самого быстрого» пакета вычисления базиса Гребнера и редукции полиномов, а основной задачей было построения пакета реализующего несколько стандартных алгоритмов и исследовать возможную эффективность применения изложенных в работе. В качестве языка программирования использовался C++, среда разработки – Visual Studio .NET 2003.

6.1 Числа

Учитывая рост коэффициентов полиномов при операциях редукции, возникла необходимость использования библиотеки для работы с числами неограниченной точности. После экспериментов с разными открытыми библиотеками решено было использовать библиотеку GMP (GNU Multiprecision Library). Эта библиотека широко используется в областях, где необходимы быстрые вычисления над числами большой длины: в криптографии, при нахождении простых чисел, в системах компьютерной алгебры. Ключевыми характеристиками этой библиотеки являются:

- 1) оптимизированный и хорошо протестированный код
- 2) Наличие всех современных алгоритмов для операций над целыми числами (в частности умножение с использованием методов Фурье, Карацубы, различные варианты нахождения наибольшего общего делителя)
- 3) Библиотека существует в вариантах для большего числа платформ и операционных систем.
- 4) Бесплатность для некоммерческого применения

Включение файла “gmp.h” в заголовочную часть программы определяет следующие типы для работы с числами неограниченной точности:

- * mpz_t – целое
- * mpq_t – рациональное число
- * mpf_t – число с плавающей точкой
- * mpn_t – внутренний тип, представляющий собой беззнаковое целое

Для коэффициентов мономов полинома в программе используется тип mpq_t.

Опишем функции необходимые для правильной работы с библиотекой GMP использованные в работе. Прежде чем использовать переменную ее нужно проинициализировать, а после выполнения требуемых действий очистить. Все функции работы с mpq_t начинаются с префикса “mpq_”:

- * mpq_init(mpq_t) – инициализация
- * mpq_clear(mpq_t) – очистка памяти, занимаемой переменной
- * mpq_set_si(mpq_t, long, unsigned long) – установка значения числителя и знаменателя.
- * mpq_neg(mpq_t, mpq_t) – обратное значение
- * mpq_canonicalize(mpq_t) – канонизация (сокращение) рационального числа
- * mpq_***(mpq_t, mpq_t, mpq_t) – арифметическая операция с рациональными числами, где *** это add, sub, mul или div.

6.2 Мономы

Моном в программе реализован как сортированный по переменной список пар (идентификатор переменной, степень) и коэффициент монома:

```
template <class Number> class Monome. Вычисления, производимые с
```

мономами, необходимые для текущей реализации (далее k – количество переменных в мономе):

1) проверка делимости двух мономов

реализовано как последовательное сравнение степеней переменных.

```
template <class Number>
bool Monome::IsDividedTo(const Monome<Number>&p) const
```

2) лексикографическое сравнение мономов

реализовано как последовательное сравнение списков переменных.

```
template <class Number>
bool Monome::IsGreaterThan(const Monome<Number>&p) const
```

3) умножение мономов

реализовано как слияние двух сортированных списков, со сложением степени, в случае равенства идентификаторов переменной, и перемножением коэффициентов.

```
template <class Number>
bool Monome::Mul(const Monome<Number>&p)
```

4) деление мономов

аналогично умножению.

```
template <class Number>
bool Monome::Div(const Monome<Number>&p)
template <class Number>
bool Monome::DivSignature(const Monome<Number>&p) //коэффициент
не меняется
```

5) извлечение из монома одной переменной

реализовано как понижение степени последнего элемента списка переменных, с удалением, если степень стала = 0.

```
template <class Number>
Monome<Number> Monome::PickOneVariable()
```

6) вычисление хэш-кода монома

метод необходим для оптимизированного вычисления нормальной формы. Результат зависит только от степеней переменных монома, и не зависит от коэффициента, реализован как последовательное сложение степени

переменной, умноженной на коэффициент, зависящий от идентификатора переменной. Метод считает значение по запросу, и пересчитывает его только при изменении монома.

```
template <class Number>
unsigned long Monome::GetHash() const
```

6.3 Полиномы

Полиномы в программе – основной тип данных для внешнего пользователя. Полином представлен как сортированный список мономов (сортировка по используемому мономиальному упорядочению):

```
typedef RationalNum          cf_type;
typedef Monome<cf_type> pm_type;
class Polynom;
```

Операции с полиномами:

- 1) сложение полиномов, (также сложение полинома и монома)

реализовано как слияние двух сортированных списков, со сложением коэффициентов мономов в случае равенства сигнатур.

```
Polynom& Polynom::Add(const pm_type& p)
Polynom& Polynom::Add(Polynom& p)
```

- 2) Умножение полинома на моном

Реализовано как последовательное умножение мономов. Также реализована возможность умножения полиномов, но в алгоритмах такого рода операции не нужны.

```
Polynom& Polynom::Mul(cf_type c)
Polynom& Polynom::Mul(pm_type& p)
Polynom& Polynom::Mul(Polynom& p)
```

- 3) Взятие старшего члена

Реализовано как возвращение первого элемента списка мономов.

```
pm_type Polynom::getLT() const
```

- 4) Редукция полинома с помощью списка полиномов.

Здесь реализован алгоритм деления полинома на конечное множество полиномов.

```
void Polinom::Reduce(list<Polynom>& in, Polynom& rest)
```

6.4 Диаграммы Юнга

Эта структура служит для операций не над полиномом, а над их диаграммами. Диаграмма представлена все тем же сортированным списком мономов, при этом коэффициенты мономов не учувствуют в операциях:

```
class Diagramm;
```

Программе необходимы:

- 1) инициализация диаграммы полиномом.

```
Diagramm & Diagramm::operator=(Polynom& p)
```

- 2) Сумма диаграмм

```
Diagramm & Diagramm::Add(Diagramm & p)
```

- 3) Умножение диаграммы на моном

```
Diagramm & Diagramm::Mul(Monome<BiNumber>& p)
```

- 4) Сравнение диаграмм

```
bool Diagramm::IsEqualSignature(Diagramm & p)
```

- 5) Удаление из диаграммы старшего члена

```
Diagramm & Diagramm::RemoveLT()
```

6.5 Хэш таблица нормальных форм мономов

Поскольку у монома имеется метод получения хэш кода, то эта структура представляет собой хэш таблицу на фиксированное, большое количество ячеек, и стандартный набор операции:

- 1) проверка на наличие монома в таблице
- 2) взятие полинома по ключу – моному
- 3) запись полинома по ключу – моному

Класс, предоставляющий оптимизированные операции и хранящий в себе хэш назван:

```
class Reduction;
```

Он содержит две основные операции:

- 1) установка множества, по которому будет производиться редукция

```
void Reduction::setBase(PolynomList g)
```

- 2) собственно редукция, имплементирующая алгоритм 3

```
Polynom Reduction::normalF(const Polynom& p)
```

7. Вопросы охраны труда и эргономики

Для решения задач системного программирования необходимо использование вычислительной техники. Практика работы с вычислительной техникой показывает, что наиболее эффективной работы и улучшения условий ее эксплуатации можно добиться при одновременном использовании большого числа электронно-вычислительных машин, сосредоточенных в вычислительном центре (ВЦ), и при тесном сотрудничестве специалистов различного профиля.

Критериями выбора оптимального варианта организации работы в ВЦ является техническая эффективность и соответствие требованиям эргономики и охраны труда. Эргономикой изучаются возможности и особенности деятельности человека в процессе труда с целью создания таких условий, методов и организаций трудовой деятельности, которые делают трудовой процесс наиболее производительным и вместе с тем обеспечивают безопасность и удобство работающему, сохраняют его здоровье и работоспособность.

7.1 Общая характеристика условий труда

Разработка программного продукта производится на рабочем месте программиста, т.е. в помещении, оснащённом соответствующим для этого оборудованием. К данному оборудованию относится компьютер класса Pentium 4. К используемым материалам относятся: магнитные и лазерные диски, на которые копируется информация. Все рабочие места не реже двух раз в пять лет подлежат аттестации. По результатам аттестации рабочие места подразделяются на три группы:

- аттестованные – рабочие места, показатели которых полностью соответствуют предъявленным требованиям;
- подлежащие рационализации – рабочие места, несоответствующие показатели которых могут быть доведены до уровня этих требований в процессе рационализации;

- подлежащие ликвидации – рабочие места, показатели которых не соответствуют и не могут быть доведены до уровня установленных требований [14].

В результате проведенной аттестации рабочего места программиста имеем следующие данные по условиям труда.

Микроклимат производственного помещения определяется температурой (°C), относительной влажностью (%), скоростью движения воздуха (м/с). Согласно [14] к категории 1а относятся работы, производимые сидя и не требующие физического напряжения, при которых расход энергии составляет до 120 ккал/ч; к категории 1б относятся работы, производимые сидя, стоя или связанные с ходьбой и сопровождающиеся некоторым физическим напряжением, при которых расход энергии составляет от 120 до 150 ккал/ч. Разработка программного продукта, производимая на данном рабочем месте, относится по величине энергозатрат к категории 1а.

7.2 Эргономические требования

Рассмотрим более подробно требования эргономики, предъявляемые к рабочему месту оператора при работе с видеотерминалами (ВТ).

Под рабочим местом понимается зона, оснащенная необходимыми техническими средствами, в которой совершается трудовая деятельность исполнителя или группы исполнителей, совместно выполняющих одну работу или операцию. Организацией рабочего места называется система мероприятий по оснащению рабочего места средствами и предметами труда и их размещению в определенном порядке. В соответствии с требованиями эргономики, рабочее место должно быть приспособлено для конкретного вида деятельности и для работников определенной квалификации с учетом их физических и психических возможностей и особенностей. Конструкция рабочего места должна обеспечивать быстроту, безопасность, простоту и экономичность технического обслуживания в нормальных и аварийных условиях; полностью отвечать функциональным требованиям и

предполагаемым условиям эксплуатации. При конструировании производственного оборудования необходимо предусматривать возможность регулирования отдельных его элементов с тем, чтобы обеспечивать оптимальное положение работающего.

При организации рабочего места учитываются антропометрические данные программистов, а также предусматривается соответствующее размещение элементов оборудования в зависимости от характера выполняемой работы.

При размещении ВТ на рабочем месте учитываются границы полей зрения программиста, которые определяются положением глаз и головы. Различают зоны зрительного наблюдения в вертикальной плоскости, ограниченные определенными углами, в которых располагают экран ВТ (45° - 60°), пюпитр (35° - 45°) и клавиатура. При периодическом наблюдении за экраном рекомендуется располагать элементы оборудования так, чтобы экран находился справа, клавиатура – напротив правого плеча, а документы – в центре угла обзора. При постоянной работе экран должен быть расположен в центре поля обзора, документы – слева на столе или специальной подставке [14]. Рабочий стол должен иметь стабильную конструкцию. Плоскость стола выбирается в зависимости от размера документов. При больших размерах документов она должна быть 160 x 90 см. Плоскость стола, а также сидение программиста должны регулироваться по высоте. Высоту плоскости стола необходимо регулировать в диапазоне 65-85 или 68-84 см. При этом высота от горизонтальной линии зрения до рабочей поверхности стола при выпрямленной рабочей позе должна быть 40-50 см.

Табл. 8.1. Оптимальная высота сидения и подставки для ног в зависимости от роста человека (см).

150	155	160	165	170	175	180
59/22	57/18	55/14	53/10	51/6	49/2	47/0
58/21	56/17	54/13	52/9	50/5	48/1	46/0
57/20	55/16	53/12	51/8	49/4	47/0	45/0
56/19	54/15	52/11	50/7	48/3	46/0	44/0
55/18	53/14	51/10	49/6	47/2	45/0	43/0
54/17	52/13	50/9	48/5	46/1	44/0	42/0
53/16	51/12	49/8	47/4	45/0	43/0	41/0
52/15	50/11	48/7	46/3	44/0	42/0	40/0
51/14	49/10	47/6	45/2	43/0	41/0	39/0
50/13	48/9	46/5	44/1	42/0	40/0	38/0
49/12	47/8	45/4	43/0	41/0	39/0	37/0
48/11	46/7	44/3	42/0	40/0	38/0	36/0
47/10	45/6	43/2	41/0	39/0	37/0	35/0
46/9	44/5	42/1	40/0	38/0	36/0	34/0
45/8	43/4	41/0	39/0	37/0	35/0	33/0

Высота сидения от пола должна регулироваться в пределах 42-55 см. По желанию программиста может быть установлена подставка для ног размером 40x30x15 см и углом наклона 0-20° с нескользящим покрытием и неперемещаемая по полу (Табл. 8.1).

Покрывтие стола должно быть матовым (с коэффициентом отражения 20-50%) и легко чиститься; углы и передняя грань доски должны быть закругленными. Высота пространства под столом для ног рекомендуется порядка 60 см на уровне колен и не менее 80 см на уровне ступней.

Тип рабочего кресла выбирается в зависимости от продолжительности работы. При длительной работе кресло должно быть массивным, при кратковременной – легкой конструкции, свободно отодвигающееся.

Сидение должно быть удобным, иметь закругленные края, наклоняться по отношению к горизонтали вперед на 2° и назад на 14°. Его размеры не должны превышать 40x40 см. Кроме того, сидение должно быть покрыто латексом толщиной около 1 см, сверху которого накладывается влагонепроницаемый материал (меланжевая ткань, натуральное волокно).

Высота спинки кресла рекомендуется 48-50 см от поверхности сидения и с регулировкой в передне-заднем направлении. На высоте 10-20 см от сидения спинка должна быть оборудована поясничным опорным валиком.

Подлокотники рекомендуются лишь при эпизодической работе с ВТ, при постоянной работе подлокотники ограничивают движение, а, следовательно, в этой ситуации креслом с подлокотниками пользоваться не рекомендуется [14].

7.3 Микроклиматические условия

Устанавливаемые ГОСТ12.1.005-88 и СанПиН 2.2.2/2.4.1340-03 оптимальные нормы температуры (23-25°C) и относительной влажности (60-40%) для теплого периода года, а также (22-24 °C) и (60-40 %) соответственно для холодного и переходного, скорости движения воздуха соблюдаются благодаря наличию системы кондиционирования, работающей в режиме автоматического регулирования. Скорость движения воздуха не превышает допустимого значения (<0.1 м/с, как для холодного, так и для ~~теплого~~ оптимального периода). допустимые параметры микроклимата согласно [14] приведены соответственно в Табл. 8.2 и Табл. 8.3.

Табл. 8.2. Оптимальные параметры микроклимата.

Период года	Категория работ	Температура, °C	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный и переходный	Легкая Ia	22-24	40-60	0.1
	Легкая Ib	21-23	40-60	0.1
Теплый	Легкая Ia	23-25	40-60	0.1
	Легкая Ib	22-24	40-60	0.2

Табл. 8.3. Допустимые параметры микроклимата.

Период года	Категория работ	Температура, °C	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный и переходный	Легкая	19-25	не более 75	<0.2
Теплый	Легкая	Не более чем на 3°C выше средней температуры наружного воздуха в 13 ч. самого жаркого месяца, но не более 28°C	24°C: <75 25°C: <70 26°C: <65 27°C: <60 28°C: <55	<0.2-0.5

В помещении отсутствуют выделения вредных веществ, поэтому вентиляция служит для удаления насыщенного углекислым газом и пылью воздуха и подачи свежего. Концентрация пыли должна быть не более 0.3 мг/м³ при размере частиц не более 5 микрон [14].

В Табл. 8.4 приведены нормы подачи свежего воздуха в помещение в зависимости от объема на одного работающего, в соответствии с [14].

Табл. 8.4. Нормы подачи воздуха.

Объем помещения на одного работающего, м ³	Норма подачи воздуха, м ³ /ч
До 20	не менее 30
20 – 40	не менее 20
от 40	естественная вентиляция

Объем данного помещения составляет $7 \times 9 \times 3,5 = 396$ м³, количество работающих – 12 человек, следовательно, необходимо предусмотреть наличие кондиционера.

7.4 Уровень шума

На рабочем месте программиста источником шума является вычислительная машина, производящая постоянный небольшой шум. Шум представляет собой сочетание звуков, различных по интенсивности и частоте в частотном диапазоне 16-20 кГц, не несущих полезной информации. Шум вредно воздействует не только на органы слуха, но и на весь организм человека в целом через центральную нервную систему. Шум – причина преждевременного утомления, ослабления внимания, памяти.

Характеристикой постоянного шума на рабочих местах является уровень звукового давления в децибелах в установленных октавных полосах:

$$L = 20 \lg \left(\frac{P}{P_0} \right)$$

где P – среднеквадратическое значение звукового давления (Па), $P_0 = 2 \cdot 10^{-5}$ Па.

Характеристикой непостоянного шума является интегральный критерий – эквивалентный уровень звукового давления L (дБА), измеряемый по шкале А, имитирующей кривую чувствительности уха человека. При нормировании по предельному спектру шума учитываются уровни звукового давления в восьми октавных полосах частот со среднегеометрическими частотами 63, 125, 250, 500, 1000, 2000, 4000, 8000 Гц. Этот метод является основным для постоянных шумов.

Постоянным считается шум, уровень звука которого за 8-часовой рабочий день изменяется во времени не более чем на 5дБА.

В Табл. 8.5 приведены предельно допустимые значения уровня шума для данного производственного помещения в отдельных октавных полосах, в соответствии с [14].

Табл. 8.5. Предельно-допустимые уровни шума.

Среднегеометрические частоты октавных полос, Гц							
63	125	250	500	1000	2000	4000	8000
Уровни звукового давления, дБ							
71	61	54	49	45	42	40	38

Предельно допустимый эквивалентный уровень звука 50 дБ (для помещения в котором осуществляется эксплуатация ЭВМ, [14]). Для данного помещения уровень звука не превышает допустимого, поэтому применение специальных звукоизоляционных материалов не требуется. Воздействие вибрации, инфра- и ультразвука в помещении отсутствует.

7.5 Освещение

Помещения с видеотерминалами (ВДТ) и ПЭВМ должны иметь естественное и искусственное освещение. Естественное освещение должно осуществляться через светопроемы, ориентированные преимущественно на север и северо-восток [14].

Искусственное освещение в помещениях эксплуатации ВДТ и ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, допускается применение системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов). Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк [14].

Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Освещение рабочих поверхностей осуществляется с помощью люминесцентных ламп ЛБ белого света, которые предназначены для помещений с нормальными условиями труда.

7.6 Обоснование и выбор системы освещения

При разработке программного продукта используется компьютер класса *Pentium 4* и осуществляется ввод данных с клавиатуры или мыши, визуальный контроль символьной и графической информации, отображаемой на дисплее, а также производится работа с текстовыми документами. Следовательно, для освещения необходимо применение системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов). Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 400-500 лк [14]. Согласно [14] рекомендуется норма освещенности рабочего места оператора в системе общего освещения – 150-200 лк.

На рабочем месте обычно применяют совмещенное (естественное и искусственное) освещение, причем естественное – в виде бокового. Наиболее благоприятна северная ориентация светопроемов в помещениях с рабочими

местами программистов ЭВМ. При выполнении работ указанного характера искусственное освещение осуществляется в виде одного общего освещения. В связи с тем, что при работе с ПЭВМ особую роль играет равномерность распределения световой энергии (для исключения частых переадаптаций глаз программиста), светильники общего освещения следует располагать над рабочими поверхностями в равномерно прямоугольном порядке.

Для искусственного освещения помещений следует использовать главным образом люминесцентные лампы, у которых высокая световая отдача (до 120 лм/Вт и более), продолжительный срок службы (до 10000 ч.), малая яркость светящейся поверхности, близкий к естественному спектральный состав излучаемого света, что обеспечивает хорошую светопередачу. Наиболее приемлемыми являются лампы люминесцентные ЛБ (белого света) и ЛТБ (тепло-белого) мощностью 40 или 80 Вт. Т. к. данное помещение является помещением с нормальными условиями среды, то это позволяет использовать в системе освещения светильники ЛСПО 2 (пыле- и водонезащищенное исполнение) с люминесцентными лампами ЛХБ 40 Вт. со световым потоком 2780 лм по ГОСТ 6825-74.

7.7 Излучения

К числу вредных факторов можно отнести излучение электронно-лучевой трубки дисплея, которое подразделяется на несколько типов излучений, в т. ч. рентгеновское, радиочастотное, видимое и ультрафиолетовое. Уровни данных излучений не превышают допустимые нормы, но работа за компьютером, особенно длительное время, вредит зрению. Поэтому применяются защитные экраны для дисплеев, которые фильтруют излучения, причиняющие вред. Защитные экраны снижают утомляемость глаз. Допустимые уровни излучения регламентированы “ Гигиенические требования к персональным электронно-вычислительным машинам и организации работы” (СанПиН 2.2.2/2.4.1340-03), введенными с 30.05.03.

Табл. 8.6. Допустимые значения параметров неионизирующих электромагнитных излучений.

Наименование параметров		ВДУ ЭМП
Напряженность электрического поля	В диапазоне частот 5 Гц – 2 кГц	25 В/м
	В диапазоне частот 2 кГц – 400 кГц	2,5 В/м
Плотность магнитного	В диапазоне частот 5 Гц – 2 кГц	250 нТл
	В диапазоне частот 2 кГц – 400 кГц	25 нТл
Электростатический потенциал экрана видеомонитора		500 В

Для дисплеев на ЭЛТ частота обновления изображения должна быть не менее 75 Гц при всех режимах разрешения экрана, гарантируемых нормативной документацией на конкретный тип дисплея и не менее 60 Гц для дисплеев на плоских дискретных экранах (жидкокристаллических, плазменных и т.п.).

7.8 Электробезопасность

По степени поражения людей электрическим током помещение относится к категории помещений без повышенной опасности, поскольку является сухим, нежарким, непыльным, с нетокопроводящим полом, возможность случайного одновременного прикосновения к токоведущим частям и заземленным конструкциям присутствует лишь в случае грубейшего нарушения ТБ при техническом обслуживании ЭВМ и ПУ. В связи с этим требуется соблюдение необходимых мер предосторожности, в т. ч. использование средств индивидуальной защиты (при техническом обслуживании оборудования), что закреплено в инструкции по ТБ, в соответствии с ГОСТ 121-038-82. Электроснабжение осуществляется от однофазной сети переменного тока номинальным напряжением 220 В и частотой переменного тока 50 Гц с заземленной нейтралью. Предельные отклонения по напряжению и частоте соответствуют ГОСТ 21552-84. Для защиты от поражения электрическим током выполнено заземление корпусов оборудования. Все оборудование имеет предохранители в цепи питания.

При прикосновении в ВЦ к любому из элементов ПЭВМ могут возникнуть разрядные токи статического электричества. Для снижения возникающих статических зарядов в ВЦ покрытие полов следует выполнять из однослойного поливинилхлоридного линолеума. Не рекомендуется носить одежду из синтетических тканей. К общим мероприятиям защиты от статического электричества в ВЦ можно отнести общее и местное увлажнение воздуха (до 50%), ионизацию воздуха.

7.9 Мероприятия по созданию благоприятных условий труда

Помещение оснащено кондиционером, работающим в автономном режиме (со встроенными холодильными машинами и электрическими воздухонагревателями). Кроме того, данная система кондиционирования содержит технические средства для очистки, увлажнения (осушения) и перемещения воздуха, а также автоматического регулирования его температуры, влажности, давления, состава, скорости движения. Благодаря этому в помещениях создается и поддерживается состояние микроклимата, наиболее благоприятное для самочувствия людей и работы оборудования в данный период и для данной категории работ (температура 20-25°C, относительная влажность 60-40%, скорость движения воздуха <0.2 м/с), в соответствии с требованиями ГОСТ 12.1.005, ГОСТ 12.2.028, [14].

Для снижения уровня шума и вибрации оборудование и приборы устанавливаются на специальные амортизирующие прокладки, стены и потолки помещений, где стоят ЭВМ, печатающие и другие периферийные устройства, облицовываются звукопоглощающими материалами (специальные перфорированные плиты и панели). Также можно использовать подвесные акустические потолки (ГОСТ 12.1.003-83 и ГОСТ 12.1.012-90).

Для исключения бликов отражения на экранах применяется специальное антибликовое матовое покрытие экрана монитора. В качестве индивидуального средства защиты от излучения электронно-лучевой трубки

дисплея следует использовать защитный фильтр, который предназначен для защиты пользователей ЭВМ от вредных воздействий излучений мониторов в осевом направлении и позволяет снизить их до уровней, безопасных для организма человека.

К техническим средствам, обеспечивающим безопасность работ, относятся: система пожарной сигнализации, системы кондиционирования, система аварийного отключения электропитания.

Выбор средств индивидуальной защиты регламентирован ГОСТ 12.3.002-75. Поскольку данное производственное помещение не относится к категории помещений с неблагоприятными условиями труда, то средства индивидуальной защиты практически не применяются, если не считать защитные фильтры на экранах дисплеев, соответствующих требованиям санитарно-гигиенических норм N 2151-80 и N 4559-88 Министерства Здравоохранения РФ.

7.10 Методика и приборы контроля параметров среды

Метрологическое обеспечение в области безопасности труда осуществляется на основе положений ГОСТ 12.0.005-84 «Метрологическое обеспечение в области безопасности труда. Основные положения», системы стандартов безопасности труда, санитарных норм и правил Минздрава РФ и др. Приборы для контроля микроклимата, шума, освещенности, концентрации пыли, сопротивления изоляции приведены в Табл. 8.7.

Уровень освещенности следует проверять не реже одного раза в год в контрольных точках производственного помещения после очередной чистки светильников и замены перегоревших ламп.

Табл. 8.7. Характеристики приборов контроля.

Измеряемая величина, единица измерений	Допустимая погрешность измерений	Рабочие средства измерений			Погрешность измерений
		Тип	Частотный Диапазон	Динамический диапазон	
Температура, °С и относительная влажность воздуха, %	0,2°С 5,0%	Аспирационный Психрометр МВ – 4М	—	-35...51°С 10...100%	0,1°С 5%
Скорость движения Воздуха, м/с	0,1 м/с	Кататермометр	—	0,02...1 м/с	—
Уровень звукового давления в октавных полосах частот, дБ	—	Шумомер типа 00017 Роботрон	2 – 20000 Гц	25 – 140 дБ	1 дБ
Освещенность, Лк	5%	Люксметр Ю – 116	0,38 – 0,7 мкм	50 – 100000 лк	20%
Сопротивление изоляции, МОм	1%	Мегаомметр М 110211	—	0 – 200 МОм	1%

Фактическая освещенность должна быть больше или равна нормируемой освещенности умноженной на коэффициент запаса (0.9-1.2). При несоблюдении этого соотношения осветительная установка не пригодна для дальнейшей эксплуатации, и ее следует реконструировать или капитально ремонтировать.

7.11 Общая оценка фактических условий труда

Оценка условий труда в зависимости от факторов среды (химических, биологических, физических) производится для каждого для каждого из них в отдельности. Также в отдельности оценивают условия труда по показателям тяжести и напряженности трудового процесса (Табл. 8.8-8.9).

Общая оценка условий труда в зависимости от факторов среды (химических, биологических, физических) и трудового процесса (тяжести и напряженности труда) производится по наиболее высокому классу и степени вредности всех рассмотренных факторов. В случае если три и более факторов относятся к классу 3.1, то общая оценка условий труда соответствует классу-3.2; а при наличии двух и более факторов классов 3.2, 3.3, 3.4 условий труда оценивается на одну степень выше.[15]

Табл. 8.8. Оценка условий труда по показателям тяжести трудового

процесса

№ п/п	Класс условий труда
1. Физическая, динамическая нагрузка	
1.1	1
1.2	1
2. Масса поднимаемого и перемещаемого груза вручную	
2.1	1
2.2	1
2.3	1
3. Стереотипные рабочие движения	
3.1	1
3.2	1
4. Статическая нагрузка	1
5. Рабочая поза	1
6. Наклоны корпуса	1
7. Перемещение в пространстве	1

Табл. 8.9. Оценка условий труда по показателям напряженности трудового процесса

№ п/п	Класс условий труда
1. Интеллектуальные нагрузки	
1.1	3.1
1.2	2
1.3	3.1
1.4	2
2. Сенсорные нагрузки	
2.1	3.1
2.2	2
2.3	1
2.4.1	2
2.4.2	1
2.4.3	3.2
2.4.4	1
3. Эмоциональная нагрузка	
3.1	2
3.2	1
3.3	1
4. Монотонность нагрузок	
4.1	3.1
4.2	2
5. Режим работы	
5.1	1

5.2	1
-----	---

Табл. 8.10. Общая оценка напряженности трудового процесса (на основании учета числа показателей напряженности)

К л а с с ы у с л о в и й т р у д а					Общая оценка напряженности трудового процесса класс условий труда
Оптимальный труда легкой степени 1)	Допустимый труда средней степени 2)	Вредный (напряженный труд)			
		1 степени 3.1	2 степени 3.2	3 степени 3.3	
≤10	≥6	-	-	-	2
1 и 2 кл. ≤10		≥6	-	-	3.1
1 и 2 кл. =10		5	1	-	3.1
1 и 2 кл. =10		4	2	-	3.1
1 и 2 кл. =10		3	3	-	3.1
1 и 2 кл. =10		2	4	-	3.1
1 и 2 кл. =10		1	5	-	3.1
1,2,3.1 кл. =10			6	-	3.2
1,2,3.1 кл. ≤9			≥7		3.3

По таблице 8.10. и данных из таблицы 8.9. определим общую напряженность трудового процесса. Класс условий труда по напряженности равен 3.1.

Табл. 8.11. Оценка условий труда по степени вредности и опасности.

Ф а к т о р		К л а с с ы у с л о в и й т р у д а						
		1 класс оптима льный	2 класс допу стим ый	3 класс вредный				4 класс опасны й
				3.1 1-ой степен и	3.2 2-ой степен и	3.3 3-ей степен и	3.4 4-ой степен и	
Химический								
Биологический								
Ф и з и ч	Аэрозоли Ф							
	Шум		X					
	Вибрация лок.							
	Вибрация общ.							
	Инфразвук							

е с к и й	Ультразвук							
	ЭМИ		X					
	Иониз. излуч.			X				
	Микроклимат		X					
	Освещенность		X					
Тяжесть труда		X						
Напряженность труда				X				
Общая оценка условий труда				X				

После оценки всех условий труда подсчитан класс условий труда по вредности и опасности. Условия труда относятся к классу 3.1.

Приложение

1. Характеристики тестовой машины:

Процессор: Intel Pentium 4 CPU 3.0 GHz

Объем памяти: 1 Gb.

2. Лексикографический базис Гребнера системы cyclic5:

$x_1 + x_2 + x_3 + x_4 + x_5,$
 $275*x_2^2 + 825*x_2*x_5 + 550*x_4^6*x_5 + 1650*x_4^5*x_5^2 + 275*x_4^4*x_5^3 - 550*x_4^3*x_5^4 + 275*x_4^2 -$
 $566*x_4*x_5^{11} - 69003*x_4*x_5^6 + 69019*x_4*x_5 - 1467*x_5^{12} - 178981*x_5^7 + 179073*x_5^2,$
 $275*x_2*x_3 - 275*x_2*x_5 + 275*x_3^2 + 550*x_3*x_5 - 330*x_4^6*x_5 - 1045*x_4^5*x_5^2 -$
 $275*x_4^4*x_5^3 + 275*x_4^3*x_5^4 - 550*x_4^2 + 334*x_4*x_5^{11} + 40722*x_4*x_5^6 -$
 $40726*x_4*x_5 + 867*x_5^{12} + 105776*x_5^7 - 105873*x_5^2,$
 $275*x_2*x_4 - 275*x_2*x_5 - 110*x_4^6*x_5 - 440*x_4^5*x_5^2 -$
 $275*x_4^4*x_5^3 + 275*x_4^3*x_5^4 + 124*x_4*x_5^{11} + 15092*x_4*x_5^6 - 15106*x_4*x_5 + 346*x_5^{12} + 42218*x_5^7 -$
 $42124*x_5^2,$
 $55*x_2*x_5^5 - 55*x_2 + x_5^{11} + 143*x_5^6 - 144*x_5,$
 $275*x_3^3 + 550*x_3^2*x_5 - 550*x_3*x_5^2 + 275*x_4^6*x_5^2 + 550*x_4^5*x_5^3 - 550*x_4^4*x_5^4 + 550*x_4^2*x_5 -$
 $232*x_4*x_5^{12} - 28336*x_4*x_5^7 + 28018*x_4*x_5^2 - 568*x_5^{13} - 69289*x_5^8 + 69307*x_5^3,$
 $275*x_3*x_4 - 275*x_3*x_5 + 440*x_4^6*x_5 + 1210*x_4^5*x_5^2 - 275*x_4^3*x_5^4 + 275*x_4^2 - 442*x_4*x_5^{11} -$
 $53911*x_4*x_5^6 + 53913*x_4*x_5 - 1121*x_5^{12} - 136763*x_5^7 + 136674*x_5^2,$
 $55*x_3*x_5^5 - 55*x_3 + x_5^{11} + 143*x_5^6 - 144*x_5,$
 $55*x_4^7 + 165*x_4^6*x_5 + 55*x_4^5*x_5^2 - 55*x_4^2 - 398*x_4*x_5^{11} - 48554*x_4*x_5^6 + 48787*x_4*x_5 - 1042*x_5^{12} -$
 $127116*x_5^7 + 128103*x_5^2,$
 $55*x_4^2*x_5^5 - 55*x_4^2 - 2*x_4*x_5^{11} - 231*x_4*x_5^6 + 233*x_4*x_5 - 8*x_5^{12} - 979*x_5^7 + 987*x_5^2,$
 $x_5^{15} + 122*x_5^{10} - 122*x_5^5 - 1;$

Литература

1. Бухбергер Б.: Базисы Гребнера. Алгоритмический метод в теории полиномиальных идеалов. Компьютерная алгебра. Символьные и алгебраические вычисления. М., Мир, 1986
2. Becker T., Weispfenning V. (1993), Groebner Bases, Springer-Verlag, New York-Berlin-Heidelberg
3. Gebauer, Moller H.M. (1988), On an installation of Buchberger's algorithm, Computation Aspects of Commutative Algebra, ed. by Robbiano L, Academic Press, New York.
4. Giovini A., Mora T., Niesi G., Robbiano L., Tarveso C. (1991) "One sugar cube, please", or Selection strategies in the Buchberger algorithm, in: ISSAC 1991, Proc. Of the 1991 Intern. Symp. On Symbolic and Algebraic Comput., ad by Watt S., ACM Press, New York.
5. Кокс Д., Литтл Дж., О'Ши Д. Идеалы, многообразия и алгоритмы. Введение в вычислительные аспекты алгебраической геометрии и коммутативной алгебры. Мир 2000.
6. Bachmann O., Schoenemann H. Monomial Representation for Groebner Bases Computaion. Centre for Computer Algebra, Department of Mathematics, University of Kaiserslautern.
7. Faugere J-C. A new efficient algorithm for computing Groebner Bases (f4). Journal of Pure and Applied Algebra 139 (1999).
8. Lakshman Y. On the Complexity of Computing a Groebner Basis for the Radical of a Zero Dimension Ideal. Department of Computing Scince Rensselaer Polytechnic Institute Troy, NY 12180.
9. Singer M. (ed.) Computer algebra and differential equations. North Carolina State University. Raleigh, USA.
10. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. СанПин 2.2.2/2.4.1340-03.