**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ  
Кафедра высшей алгебры и защиты информации**

ЛАЗУКО

Серафим Александрович

**ПРИМЕНЕНИЕ БАЗИСОВ ГРЁБНЕРА К РЕШЕНИЮ СИСТЕМ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ И К РЕШЕНИЮ ЗАДАЧ ОБ   
ИДЕАЛАХ**

Дипломная работа

|  |  |
| --- | --- |
|  | Научный руководитель:  доктор физ.-мат. наук,  профессор В.В. Беняш-Кривец |

Допущена к защите

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Зав. кафедрой высшей алгебры и защиты информации

доктор физ.-мат. наук, профессор В.В. Беняш-Кривец

Минск, 2020

Глава 1. Идеалы, аффинные многообразия и связь между ними

§1. Аффинные многообразия.

1. Определение и свойства аффинных многообразий.

**Определение 1.1**. *Пусть k – некоторое поле,*  *– полиномы в . Положим* *.*  *– называется* ***аффинным многообразием****, определенным полиномами .*

**Лемма 1.1** (Свойства аффинных многообразий). *Если* *- аффинные многообразия, то и также являются аффинными многообразиями*.

**Доказательство**. Пусть  и . Мы утверждаем, что

1. Если точка принадлежит , то как функции , так и функции обращаются в нуль в этой точке.
2. Если то все обращаются в нуль в этой точке; значит и все функции обращаются в нуль в . Таким образом, и, аналогично, . Следовательно, . С другой стороны, пусть . Если эта точка принадлежит *V*, то все доказано, если же нет, то для некоторого . Так как функции обращаются в нуль в при всех *j*, то все равны нулю в этой точке. Значит, и

Из этой леммы вытекает, что конечные объединения и пересечения аффинных многообразий являются аффинными многообразиями.

§2. Идеалы.

**Определение 2.1**. *Подмножество называется идеалом, если выполнены следующие условия:*

1. *Если , то ;*
2. *Если и , то .*

**Лемма 2.1**. *Пусть принадлежат кольцу ; тогда множество является идеалом в . Оно называется идеалом, порожденным полиномами , а полиномы – образующими этого идеала или его порождающими элементами.*

**Доказательство**. Прежде всего, , поскольку . Пусть теперь , и . Тогда из равенств

вытекает, что – идеал.

Идеал имеет интерпретацию на языке полиномиальных уравнений. Пусть . Рассмотрим систему уравнений

Из этих уравнений мы можем вывести другие, используя обычные алгебраические преобразования. Так, например, если мы умножим первое уравнение на , второе – на и т.д., а затем сложим произведения, то получим уравнение:

которое является следствием уравнений первоначальной системы. Отметим, что левая часть этого уравнения принадлежит идеалу , т.е. идеал можно рассматривать как множество всех «полиномиальных следствий» системы .

Идеал *I* называется ***конечно порожденным****,* если существуют полиномы , такие, что ; при этом множество полиномов называется *базисом* идеала *I.*

**Определение 2.2.** *Пусть – аффинное многообразие. Положим*

*–* ***идеал****.*

**Лемма 2.2.** *Пусть – аффинное многообразие. Тогда – идеал, который мы будем называть идеалом многообразия V.*

**Доказательство**.Ясно, что , так как нулевой полином обращается в нуль на и на *V* в частности. Пусть и . Пусть () – произвольная точка из *V.* Тогда  
Отсюда следует, что *I(V)* - идеал.

Глава 2. Базисы Грёбнера

§1. Основные задачи об идеалах.

В этой главе мы рассмотрим следующие задачи:

1. Задача описания идеала. Является ли произвольный идеал   
    конечно порожденным? Другими словами, верно ли, что для некоторых ?
2. Задача о принадлежности идеалу. Пусть , и пусть задан идеал . Принадлежит полином *f* идеалу *I* или нет? На геометрическом языке эта задача может быть сформулирована так: содержится ли многообразие в многообразии ?
3. Задача решения полиномиальных уравнений. Описать множество решений в системы полиномиальных уравнений  
   Конечно, это то же самое, что описать аффинное многообразие .
4. Задача неявного представления. Пусть *V –* подмножество в , заданное параметрически:  
   Если – полиномы (или рациональные функции) от переменных , то *V* будет аффинным многообразием или его частью. Задача состоит в том. Чтобы задать *V* полиномиальными уравнениями от переменных .

§2. Упорядочение мономов в .

Тщательное рассмотрение алгоритма деления в и алгоритма приведения системы (или матрицы) к ступенчатому виду методом исключения Гаусса показывает, что понятие ***упорядочения членов*** полинома является ключевым в обоих алгоритмах. Алгоритм деления полиномов от одной переменной имеет дело, как правило, со следующим упорядочением мономов:  
Аналогично, при приведении матрицы к ступенчатому виду мы систематически обращаем в нуль главные элементы, т.е. первые слева ненулевые элементы в строках. На языке линейных систем это означает следующий порядок переменных:  
Каждое уравнение записывается в порядке убывания членов. Более того, в ступенчатом виде уравнения системы записаны в порядке убывания старших (главных) членов.  
  
Отметим, что существует взаимно однозначное соответствие между мономами и *n*-наборами (*n*-векторами) показателей степеней   
 Упорядочение, которое мы определим на , определит и упорядочение на множестве мономов: если на в , то мы будем говорить что .

Так как полином есть сумма мономов, то мы должны уметь расположить его члены в порядке убывания (или по возрастанию). Для этого наше упорядочение должно быть ***линейным***. То есть для любой пары мономов должно выполнятся ровно одно из следующих условий

Далее мы должны учесть связь упорядочения с операциями сложения и умножения полиномов. Когда мы складываем полиномы, то после приведения подобных мы просто можем переписать члены суммы в требуемом порядке. Ситуация с произведением более сложная. Дистрибутивность умножения по отношению к сложению позволяет свести задачу к случаю умножения монома на полином.

Мы по требуем, чтобы упорядочение мономов обладало следующим свойством. Если , а – произвольный моном, то . В терминах векторов – показателей степеней это означает, что если в , то для любого , .

Теперь мы можем дать определение.

1. Мономиальное упорядочение.

**Определение 2.3**. ***Мономиальным упорядочением*** *на*  *называется любое бинарное отношение > на , обладающее следующими свойствами*:

1. > *является линейным (для любых можно однозначно определить одно из следующих отношений: , , ) упорядочением на .*
2. *Если и* ,  *то .*
3. *> вполне упорядочивает , т.е. любое непустое подмножество в имеет минимальный (наименьший) элемент (по соотношению к упорядочению >).*

**Лемма 2.3**.(условие вполне упорядоченности): *Упорядочение > на вполне упорядочивает это множество тогда и только тогда, когда каждая строго убывающая последовательность элементов из   
обрывается.*

**Доказательство**. Докажем эквивалентное утверждение: > не является вполне упорядочением тогда и только тогда, когда существует бесконечная строго убывающая последовательность элементов из .

Если > не есть вполне упорядочение, то существует непустое подмножество , которое не имеет минимального элемента. Возьмем в качестве произвольный элемент из *S.* Так как он не минимален, то в *S* найдется элемент . Так как не минимален, то в *S* найдется элемент . Продолжая этот процесс, мы получим бесконечную строго убывающую последовательность   
Обратно, если существует такая бесконечная строго убывающая последовательность, то множество является непустым подмножеством в , которое не имеет минимального элемента, т.е. > не является вполне упорядочением.

В качестве примера мономиального упорядочения рассмотрим обычное упорядочение натуральных чисел из :  
Все три условия определения 2.1 выполнены. Следовательно упорядочение мономов из по степени (1) является мономиальным упорядочением.

2. Лексикографическое упорядочение.

Нашим первым примером упорядочения n-векторов будет лексикографическое упорядочение (или сокращенно *lex*-упорядочение)

**Определение 2.4** *(лексикографическое упорядочение). Пусть . Мы говорим, что , если самая левая ненулевая координата вектора положительна. Мы будем писать , если .*

Вот несколько примеров:

1. , так как
2. , так как
3. Обычный порядок переменных является *lex-*упорядочением. Так как

то .

Работая с полиномами от двух или трёх переменных, мы обозначаем переменные через а не  *.* В дальнейшем мы также будем предполагать, что алфавитный порядок .

Надо проверить, что лексикографическое упорядочение удовлетворяет трем условиям определения 2.1.

**Предложение 2.1**. *Лексикографическое упорядочение на является мономиальным упорядочением.*

**Доказательство**. (i) Тот факт, что – линейное упорядочение, прямо следует из определения и из того, что обычное упорядочение на линейно.  
(ii) Пусть . Тогда самая левая ненулевая координата вектора . Пусть это, например, . Но и   
. Тогда , и самой левой ненулевой координатой опять является .  
(iii) Предположим, что не является вполне упорядочением. Тогда по лемме 2.1 должна существовать строго убывающая бесконечная последовательность

элементов из . Докажем, что это невозможно.

Рассмотрим первые координаты векторов . По определению лексикографического упорядочения они образуют невозрастающую последовательность неотрицательных целых чисел. Так как вполне упорядочено, то эта последовательность «стабилизируется», т.е. существует такое *k*, что первые координаты векторов одинаковы при .   
Начиная с , будем рассматривать вторые (а затем третьи и т.д.) координаты. Последовательность вторых координат векторов не возрастает; значит, она «стабилизируется». Продолжая это рассуждение, мы можем найти такое *l*, что у векторов равны все координаты. Значит, это одинаковые векторы, что противоречит строгому убыванию последовательности.

В случае лексикографического упорядочения переменная больше *любого монома*, который содержит только меньшие переменные, вне зависимости от его степени. Так, при упорядочении мы имеем . В ряде случаев нам будет необходимо учитывать также степени мономов и сравнивать сначала именно степени. Это можно сделать с помощью градуированного лексикографического упорядочения (сокращенно *grlex*-*упорядочения*).

**Определение 2.5** *(градуированное лексикографическое упорядочение). Пусть . Тогда мы говорим, что , если*

Таким образом, *grlex* сначала упорядочивает по степеням, а если степени равны, то использует лексикографическое упорядочение.

Вот пример:

, так как .

**Определение 2.6** *(градуированное обратное лексикографическое упорядочение grevlex). Пусть . Тогда мы говорим, что , если  
или и самая правая ненулевая координата вектора отрицательна.*

**Пример**, так как .

Чтобы объяснить связь между *grlex* и *grevlex*, отметим сначала, что оба эти упорядочения одинаково оценивают степень монома. В случае равенства степеней *grlex* использует *lex*-упорядочение, т.е. обращает внимание на самую левую (большую) переменную и «предпочитает» *большую* степень. Напротив, *grevlex* в случае равенства степеней обращает внимание на самую правую (меньшую) переменную и «предпочитает» меньшую степень.

Посмотрим, как мономиальные упорядочения могут помочь при работе с полиномами. Пусть , и пусть выбрано мономиальное упорядочение >. Тогда мы можем однозначно упорядочить члены полинома *f* в соответствии с >.

Пусть, например, Тогда:

1. При *lex*-упорядочении мы записываем полином *f* в порядке убывания членов так:
2. При *grlex*-упорядочении запись *f* такова:
3. При *grevlex*-упорядочении запись *f* такова:

**Определение 2.7.** *Пусть - ненулевой полином в , и пусть > - мономиальное упорядочение.*

1. *Мультистепень полинома f определяется так:  
   (максимум берется по отношению к >).*
2. *Старший коэффицент полинома f – это*
3. *Старший моном полинома f – это  
   (с коэффициентом 1).*
4. *Старший член полинома f – это*

Пусть, например, , как и выше, и пусть > обозначает lex-упорядочение. Тогда

**Лемма 2.4** (свойства мультистепени). *Пусть – ненулевые полиномы. Тогда* :

1. *Если , то   
   Если, кроме того, , то указанное неравенство становится равенством.*

§3. Алгоритм деления.

1. Алгоритм деления полинома от одной переменной.

Работая с алгоритмами, мы будем использовать «псевдокод», что облегчит нам понимание формальных структур. Описание псевдокода дано в **приложении А**. Псевдокод похож на язык программирования Паскаль, и алгоритмы, написанные на нем, легко компьютезируются.

Важнейшей частью алгоритма является понятие «старшего члена» полинома от одной переменной. Вот точное определение.

**Определение 2.8**. *Пусть – ненулевой полином,  
где и (т.е. ). Тогда называется старшим членом полинома f и обозначается .*

Теперь мы можем дать описание алгоритма деления.

**Предложение 2.2** (алгоритм деления). *Пусть – ненулевой полином. Тогда любой полином может быть записан в виде   
где и либо , либо Более того, q и r определены однозначно, и имеется алгоритм для их вычисления.*

**Доказательство.** Вот алгоритм вычисления *q* и *r,* записанный в   
псевдокоде:

Операции, подчиненные оператору циклу WHILE … DO, выполняются, пока выполняется условие, записанное между WHILE и DO; … и … - это операторы определения или переопределения значений и . И и являются *переменными* в этом алгоритме – на каждом шаге их значения меняются.

Приступим к доказательству корректности алгоритма. Заметим, сначала, что равенство выполняется при начальных значениях и . Далее, на каждом шаге после переопределения и это равенство должно выполнятся, потому что

Отметим, что выполнение циклического оператора WHILE … DO прекращается, когда утверждение «» становится ложным, т.е. когда или или . Если алгоритм прекращает работу, он выдает требуемые и .

Осталось доказать, что алгоритм обязательно остановится, т.е. что утверждение между WHILE … DO в какой-то момент станет ложным. Самым важным тут является тот факт, что полином или равен нулю, или имеет степень, меньшую, чем степень полинома *r*. Докажем это. Пусть  
и пусть . Тогда

и степень полинома *r* обязана уменьшиться (или *r* обращается в нуль). Так как степень конечна, то алгоритм останавливается после конечного числа шагов.

2. Основные следствия алгоритма деления в .

**Следствие 2.1**. *Пусть*  *– ненулевой полином. Тогда он имеет в k не более чем корней.*

**Доказательство.** Применим индукцию по . Если , то *f –* ненулевая константа, и утверждение справедливо. Пусть утверждение выполняется для всех полиномов степени , и пусть *f* имеет степень . Если *f* не имеет корней в *k,* то утверждение доказано. Пусть теперь – корень полинома . Поделим *f* на . Тогда по предположению имеем  
, где , так как имеет степень один. Положив в этом равенстве , получим , т.е. , и, значит степень полинома *q* равна .

Мы утверждаем, что любой корень полинома *f*, отличный от *a,* является корнем полинома *q*. Если – корень полинома *f*, то , откуда (так как *k -* поле). По предположению индукции *q* имеет не более корней; значит, *f* имеет не более *m* корней в *k.*

**Следствие 2.2.** *Пусть k - поле. Тогда каждый идеал в может быть представлен в виде для некоторого полинома . Более того, f определен однозначно с точностью до умножения на ненулевую константу из k.*

**Доказательство.** Пусть – некоторый идел. Если , то и утверждение доказано. Пусть теперь , и пусть – ненулевой полином минимальной степени (в множестве полиномов, содержащихся в *I*). Мы утверждаем, что . Включение очевидно, так как *I* - идеал. Рассмотрим теперь полином . В соответствии с алгоритмом деления,   
, где или , или . Так как *I* – идеал, то и, значит . Если , то , что противоречит выбору полинома *f*. Значит , т.е. , что доказывает равенство .

Теперь докажем единственность. Пусть . Так как , то для некоторого полинома *h*. Имеем

т.е. . Аналогично получаем, поменяв местами , что , т.е. . Из равенства следует, что . Значит, - ненулевая константа.

Идеал, порожденный одним элементом, называют ***главным идеалом****.* Таким образом, ввиду следствия 2.2 мы говорим, что является *областью главных идеалов* или сокращенно ОГИ.

3. Наибольший общий делитель полиномов.

**Определение 2.9**. *Наибольшим общим делителем полиномов называется полином h, такой, что*

1. *h делит и , и ;*
2. *если p – некоторый полином, который делит и , и , то р делит h.*

Наибольший общий делитель будет обозначаться через *.* Основные свойства наибольших общих делителей сформулированы в следующем предложении.

**Предложение 2.3**. *Пусть Тогда*

1. НОД *существует и единственен с точностью до умножения на ненулевую константу из k;*
2. НОД *является образующим идеала ;*
3. *существует алгоритм для вычисления* НОД*.*

**Алгоритм Евклида** позволяет вычислить наибольший общий делитель двух полиномов в .

Для начала, дадим необходимые определения. Пусть , . Запишем в виде , где *q* и *r* определены, как в предложении 1.1. Тогда *r* называется остатком от деления . Теперь мы можем дать описание алгоритма Евклида:

Переменными алгоритма являются *h* и *s.* Значением *h* является первый полином в каждом НОД, а значением *s –* второй. Переход от очередного НОД к следующему происходит так же, как и соответствующий переход в цикле алгоритма. Таким образом, на каждом шаге алгоритма   
. Работа алгоритма должна прекратиться, так как степени полинома *s* уменьшаются и в некоторый момент *s*  станет равным нулю. В момент остановки алгоритма , т.е. .

**Определение 2.10**. ***Наибольшим общим делителем*** *полиномов называется полином h, такой, что*

1. *h делит ;*
2. *если p – некоторый полином, который делит , то р делит h.*

Такой полином *h* обозначается через *.* Основные свойства наибольшего общего делителя сформулированы в следующем предложении.

**Предложение 2.4**. *Пусть , . Тогда*

1. *существует и определен однозначно с точностью до умножения на ненулевую константу из k;*
2. *порождает идеал ;*
3. *Если , то ;*
4. *существует алгоритм для вычисления .*

**Доказательство.** Доказательство пп. (*i*) и (*ii*) аналогично доказательству тех же пунктов предложению 2.4. Докажем (*iii*). Пусть .   
Тогда  
Из (*ii*) следует, что

Чтобы доказать существование алгоритма, вычисляющего , нужно объединить п.(*iii*) и алгоритм Евклида.

4. Алгоритм деления в .

Выше мы рассматривали алгоритм деления для полиномов от одной переменной. Он может быть применен для решения задачи о принадлежности идеалу. Если , то, для того, чтобы узнать, принадлежит идеалу  
 или нет, мы делим :

где и или . Мы доказали, что в том и только том случае, когда . Таким образом, этот алгоритмический метод пригоден для проверки принадлежности полинома идеалу.

Для решения этой же задачи в случае нескольких переменных необходимо обобщить алгоритм деления алгоритм деления в на общий случай полиномиального кольца . Наша цель – научиться делить полином на полиномы . Это означает научиться представлять в виде

где «частные» и остаток *r* принадлежат . Чтобы корректно определить остаток, тут будут использованы мономиальные упорядочения.

**Теорема 2.1** (Алгоритм деления в ). *Зафиксируем некоторое мономиальное упорядочение > на , и пусть – упорядоченный s-набор полиномов из Тогда любой полином может быть записан в виде*

*где и или , или r - есть линейная комбинация мономов (с коэффициентами из k, ни один из которых не делится ни на один из старших членов . Мы называем r* ***остатком*** *от деления полинома f на F. Более того, если , то*

**Доказательство**. Ниже приведено формальное описание алгоритма:

В этом алгоритме переменная *p* на каждом шаге выполняет роль промежуточного делимого, переменная *r* выполняет роль остатка, а переменные выполняют роль частных. Логическая переменная «естьделение» говорит нам, делится ли старший член промежуточного переменного на какой-либо из . Каждый раз, когда мы находимся в главном цикле может произойти ровно одно из двух событий:

* (Шаг деления) Если никоторый член делит , то алгоритм продолжает работу, как в случае одной переменной.
* (Шаг вычисления остатка) Если никакой из не делит , то алгоритм прибавляет к остатку.

Чтобы проверить корректность алгоритма, мы сначала докажем, что равенство  
выполняется на каждом шаге. Очевидно, что оно выполнено для начальных значений . Пусть на некотором шаге оно имеет место. Если следующим является шаг деления, то некоторый делит и равенство  
показывает, что сумма не изменилась. Так как все остальные переменные остались теми же, то изначальное равенство выполняется и на этом шаге тоже. Если же следующим шагом является шаг вычисления остатка, то меняются и *p*, и *r*, но их сумма остается неизменной, так как   
И опять равенство выполняется на следующем шаге.

Далее, обратим внимание, что алгоритм прекращает работу, когда . В этом случае

Так как к *r* добавлялись только такие члены, которые не делятся ни на один из , то это означает, что удовлетворяют условиям теоремы 4.1 в случае остановки работы алгоритма.

Осталось доказать, что алгоритм в какой-то момент остановится. Для этого нужно заметить, что каждый раз, когда мы заново вычисляем переменную *p*, или ее мультистепень уменьшается (относительно заданного упорядочения), или *p* обращается в нуль. Чтобы доказать это, предположим сначала, что *p* изменилось в ходе шага деления:

Согласно свойству мультистепени мы имеем

так что *р* и имеют одинаковые старшие члены. Следовательно, их разность имеет строго меньшую мультистепень (если ). Пусть теперь *р* меняется в ходе шага вычисления остатка:

Очевидно, что здесь , т.е. в обоих случаях мультистепень уменьшается. Если алгоритм не останавливается, то мы получаем бесконечную строго убывающую последовательность мультистепеней. Но так как > является вполне упорядочением, то это противоречит лемме о свойстве мультистепени. Таким образом, в какой-то момент *p* обратится в нуль, и алгоритм остановится после конечного числа шагов.

Осталось установить связь между Каждый член полинома равен для некоторого значения переменной *р.* Начальное значение *p* есть *f*, и мы только что доказали, что мультистепень *p* строго убывает; значит, . Таким образом , если . Доказательство теоремы закончено.

**Пример**. Рассмотрим полином

при *lex-*упорядочение с . Применим алгоритм деления из теоремы 2.1 и поделим на

В результате выполнения алгоритма мы должны получить новое представление полинома в виде

Будем записывать делители частные и *r* в отдельный столбец

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | |  |
|  | | | |  |
|  | | |
|  | | |
|  | | |
|  | | |
|  |  | | |  |
|  |  | | |  |
|  |  | | |
|  |  | | |
|  |  |  | |
|  |  |  | |
|  |  |  | |

В результате получаем

§4. Мономиальные идеалы и лемма Диксона.

В этом параграфе рассмотрим задачу описания идеала для частного случая мономиальных идеалов.

1. Определение и свойства мономиальных идеалов.

**Определение 2.11.** *Идеал* *называется* ***мономиальным****, если существует подмножество (которое может быть бесконечным), такое, что I состоит из конечных сумм вида , где . Такой идеал I будет обозначаться через .*

Охарактеризуем все мономы, принадлежащие заданному мономиальному идеалу.

**Лемма 2.5**. П*усть – мономиальный идеал. Тогда моном принадлежит I в том и только том случае, когда делится на некоторый моном .*

**Доказательство.** Если делится на некоторый , то по определению мономиального идеала . Докажем обратное. Пусть тогда   
, где , а . Если мы рассмотрим каждый член в равенстве справа делится на некоторый . Значит, и левая часть равенства, т.е. , обладает тем же свойством, потому что моном содержится как член хотя бы в одном слагаемом .

Теперь мы докажем, что принадлежность полинома мономиальному идеалу определяется мономами, линейной комбинацией которых является *.*

**Лемма 2.6**. *Пусть I – некоторый мономиальный идеал, а . Тогда следующие условия эквивалентны:*

1. *;*
2. *каждый член полинома принадлежит I;*
3. *является k-линейной комбинацией мономов из I.*

Следствием п(iii) является тот факт, что мономиальный идеал однозначно определен своими мономами. То есть мы имеем следующее утверждение.

**Следствие 2.3**. *Два мономиальных идеала совпадают в том и только том случае, когда совпадают множества мономов, содержащихся в них.*

2. Лемма Диксона.

**Теорема 2.2**. *Любой мономиальный идеал может быть представлен в виде , где . В частности, I имеет конечный базис.*

**Доказательство**. Доказательство проводится индукцией по *n* - числу переменных. Если , то *I* порожден мономами , где . Пусть   
 – наибольший элемент в Тогда для всех имеем . Таким образом, делит все образующие , т.е. .

Пусть и теорема справедлива для . Обозначим переменные через , так что мономы в будут записываться в виде , где , а .

Пусть – мономиальный идеал. Рассмотрим идеал  
 , порожденный такими мономами , что для некоторого . Так как – мономиальный идеал в , то по предположению индукции он конечно порожден, . Идеал может рассматриваться, как «проекция» идеала в .

По определению для каждого , существует такое, что . Пусть *m –* наибольшее из . Для каждого , рассмотрим идеал , порожденный такими мономами , что . Неформально можно сказать, что – это «срез» идеала , порожденный мономами, которые содержат *y* точно в степени *l.* По предположению индукции,конечно порожден,

Мы утверждаем, что *I* порожден мономами, перечисленными в следующем списке:

Докажем, что каждый моном в *I* делится хотя бы на один моном из списка. Пусть . Если , то по определению моном делится на некоторый моном . С другой стороны, если , то по определению идеала моном делится на некоторый моном Из леммы 2.5 следует, что мономы из списка порождают идеал, содержащий те же мономы, которые содержит *I*. Тогда по следствию 2.3 эти идеалы совпадают, и наше утверждение доказано.

Чтобы закончить доказательство теоремы, нам нужно доказать, что конечное множество образующих можно выбрать из заданного множества образующих идеала *I*. Будем обозначать переменные, как и раньше, . Тогда . Нам нужно доказать, что , где . Так как , то по лемме 2.5 каждый моном делится на некоторый моном , где . Теперь очевидно, что   
.

Лемма Диксона применяется для доказательства следующего важного утверждения о мономиальных упорядочениях на .

**Следствие 2.4**. *Пусть > - некоторое отношение на удовлетворяющее следующим условиям:*

1. *> - линейное упорядочение на ;*
2. *если и , то .*

*Тогда > является вполне упорядочением в том и только том случае, когда для всех .*

**Доказательство.** . Пусть > является вполне упорядочением, и пусть – наименьший элемент в . Достаточно доказать, что . Это просто: если , то по (ii) мы можем прибавить к обеим частям и получить , что противоречит тому, что – наименьший элемент в .

. Пусть для всех , и пусть – некоторое непустое множество. Нам нужно доказать, что в существует наменьший элемент. Рассмотрим мономиальный идеал . По лемме Диксона существуют мономы , такие, что . Пусть (в противном случае переименуем мономы). Мы утверждаем, что – наименьший элемент множества . Докажем это. Рассмотрим произвольный элемент . Тогда . По лемме 2.5 моном делится на некоторый моном , т.е. . Тогда и по (ii) мы имеем

Значит, – наименьший элемент в *A.*

§5. Теорема Гильберта о базисе и базисы Грёбнера.

В этом параграфе мы дадим полное решение *задачи описания идеала*. Для каждого идеала *I* мы можем определить его *идеал старших членов* следующим образом.

**Определение 2.12**. *Пусть – ненулевой идеал.*

1. *Обозначим через множество старших членов элементов из I, т.е.*
2. *Обозначим через идеал, порожденный элементами из*

Отметим важный момент в определении . Пусть *I* конечно порожден, . Тогда и могут быть *разными* идеалами. Конечно, ; поэтому .

Докажем, что – мономиалный идеал.

**Предложение 2.5**. *Пусть – некоторый идеал. Тогда:*

1. *– мономиальный идеал;*
2. *существуют полиномы , такие, что*

**Доказательство**. (i) Старшие мономы элементов порождают мономиальный идеал Так как отличается от на ненулевой множитель из поля *k*, то этот идеал совпадает с идеалом . Таким образом, – мономиальный идеал.

(ii) Так как порожден мономами , , то по лемме Диксона для конечного набора . Так как отличается от на ненулевой множитель из поля *k,* то .

1. Теорема Гильберта о базисе.

Используя предложение 2.5 и алгоритм деления, мы можем доказать конечную порожденность *любого* полиномиального идеала. Это дает утвердительный ответ на вопрос об описании идеала. Пусть – некоторый идеал, и пусть – его идеал старших членов. Будем также считать, что задано некоторое мономиальное упорядочение, используемое в алгоритме деления.

**Теорема 2.3** (Гильберта о базисе). *Каждый идеал является конечно порожденным, т.е. , где .*

**Доказательство**. Если , то наше порождающее множество состоит из одного элемента – нулевого полинома. Если – ненулевой идеал, то порождающее множество мы будем строить следующим образом. Из предложения 2.5 вытекает, что существуют полиномы , такие, что . Мы утверждаем, что .

Так как каждый принадлежит *I*, то . Пусть теперь – некоторый элемент. Применим алгоритм деления и поделим на . В результате будет представлен в виде

Где ни один член полинома *r* нельзя поделить ни на один из . Мы утверждаем, что . Имеем

Если , то . Тогда по лемме 2.5 должен делится хотя бы на один . Но это противоречит определению остатка. Значит , т.е.

Откуда . Теорема доказана.

2. Базисы Грёбнера.

**Определение 2.13**. *Пусть задано мономиальное упорядочение. Конечное подмножество элементов идеала I называется его* ***базисом Грёбнера*** *(или стандартным базисом), если*

Это определение можно переформулировать так: множество называется базисом Грёбнера идеала *I* в том и только том случае, когда старший член любого элемента из *I* делится на хотя бы один старший член .

Из доказательства теоремы 2.3 вытекает следующий результат.

**Следствие 2.5**. *Пусть задано некоторое мономиальное упорядочение. Тогда любой ненулевой идеал обладает базисом Грёбнера. Более того, базис Грёбнера идеала является его базисом.*

**Доказательство**. Пусть – ненулевой идеал в – множество, построенное в теореме 2.3. Это множество является базисом Грёбнера по определению. Что касается второго утверждения, то, как доказано в теореме 2.3, если , то , т.е. является базисом в .

**Упражнение**. Пусть используется grlex-упорядочение с . Верно ли, что множество является базисом Грёбнера? Объясните ваш ответ.

**Решение.**

*;*

*;*

*.*

Рассмотрим следующий полином:

.

Очевидно, что . Значит, .

Но, с другой стороны, не делится ни на один из старших членов полиномов . А именно, не делится ни на , ни на , ни на . Таким образом, имеем следующее:

Мы получили, что . Но по определению базисом Грёбнера является такое конечное подмножество элементов идеала *I*, для которого

Значит, рассматриваемое нами множество не является базисом Грёбнера.

3. Свойства базисов Грёбнера.

**Предложение 2.6**. *Пусть – базис идеала , и пусть . Тогда существует единственный полином , который обладает следующими свойствами:*

1. *ни один член полинома не делится ни на один из старших членов ;*
2. *существует , такой, что .*

*То есть является остатком от деления на , не зависящим от порядка делителя в .*

**Доказательство**. Алгоритм деления позволяет записать *f*  в виде , где удовлетворяет условию (i). Условие (ii) также выполняется, так как . Существование полинома доказано.

Докажем единственность. Пусть , где удовлетворяют условиям (i), (ii). Тогда . Поэтому если , то . Тогда по лемме 2.5 делится на какой-то старший член . Но это невозможно в силу условия (i). Значит, , и единственность доказана.

**Определение 2.14**. *Остаток называется* ***нормальной формой*** *полинома . Его единственность характеризует базисы Грёбнера.*

Стоит отметить, что хотя остаток и единственен, но «частные» , вычисляемые алгоритмом деления в зависят от порядка делителей даже в случае базиса Грёбнера.

**Следствие 2.6** (Условие принадлежности идеалу). *Пусть – базис Грёбнера идеала , и пусть . Тогда в том и только том случае, когда остаток от деления полинома на равен нулю.*

**Доказательство**. Если остаток равен нулю, то, как уже отмечалось, . Обратно, пусть . Тогда равенство удовлетворяет обоим условиям предложения 2.6. Из единственности представления полинома в таком виде следует, что 0 является остатком от деления на *.*

Данное следствие позволяет построить алгоритм принадлежности к идеалу: необходимо найти остаток от деления полинома на базис Грёбнера идеала. Как построить этот базис будет обсуждаться в следующих параграфах

**Определение 2.15**. *Остаток от деления полинома на упорядоченный* ***s****-набор будет обозначаться . Если является базисом Грёбнера идеала , то по предложению 2.6 его можно рассматривать как (неупорядоченное) множество.*

**Определение 2.16**. *Пусть – ненулевые полиномы.*

1. *Пусть и . Положим для любого i. Тогда – называется* ***наименьшим общим кратным*** *мономов и . Используется обозначение ,.*
2. ***S-полиномом*** *от называется комбинация*

Следует отметить, что в знаменателе стоят не мономы, а старшие члены. S-полином специально «сконструирован» для сокращения старших   
членов.

**Лемма 2.7**. *Рассмотрим сумму , где , a для всех . Если , то является линейной комбинацией с коэффицентами в*  *S-полиномов . Более того, мультистепень каждого меньше .*

Если удовлетворяют условиям леммы 2.7, то

Используя S-полиномы и лемму 2.7 мы можем теперь доказать следующий критерий (принадлежащий Бухбергеру) того, что базис идеала является базисом Грёбнера.

4. Критерий Бухбергера.

**Теорема 2.4**. *Пусть I – некоторый полиномиальный идеал. Тогда базис – базис идеала является базисом Грёбнера в том и только том случае, когда для всех пар остаток от деления (в   
любом порядке) равен нулю.*

**Доказательство**. Пусть – ненулевой полином. Мо должны доказать, что если остатки от деления всех S-полиномов на равны нулю, то . Сначала наметим общую стратегию доказательства.

Так как , то существуют полиномы , такие, что

Из леммы 2.4 следует, что

Если здесь нет равенства, то, следовательно, произошло сокращение старших членов в (2). Лемма 2.7 позволяет выразить это в терминах S-полиномов. Тогда наше условие, что S-полиномы на выражения с меньшим числом сокращений, т.е. мы получим выражения для с меньшим числом сокращаемых старших членов. Продолжая этот процесс, мы в конце концов получим выражение типа (2) для , причем в (3) будет иметь место равенство. Тогда для некоторого *i*, т.е. делится на некоторый . Значит, , что и требуется доказать.

Приступим к подробному изложению доказательства. Рассмотрим (2). Пусть , и положим . Теперь неравенство (3) имеет вид . Рассмотрим *все* способы, какими может быть записано в виде (2). Для каждого способа мы буем иметь свое . Так как мономиальное упорядочение является вполне упорядочением, то мы можем выбрать такое выражение (2), для которого ***минимально***.

Мы покажем, что если минимально, то . Тогда в (3) равенство имеет место и, как мы видели выше, отсюда следует, что . Это и доказывает теорему.

Осталось доказать, что . Мы докажем это от противного. Если равенство места не имеет, то . Перепишем (2) в следующем виде:

Мономы во второй и третьей суммах в самой правой части равенства имеют мультистепени . Поэтому предположение означает, что первая сумма также имеет мультистепень .

Если , то сумма

Имеет в точности тот вид, который описан в условии леммы 2.7 с . Теперь из леммы 2.7 следует, что эта сумма есть линейная комбинация S-полиномов . Но

Где Значит, существуют константы , такие, что

Теперь вспомним, что, согласно нашему предположению, остаток от деления на равен нулю, т.е. каждый S-полином может быть записан в виде

Где . Из алгоритма деления также следует, что

Для всех *i,j,l.* Значит, можно сказать, что если остаток равен нулю, то существует такое представление в виде комбинации , что старшие члены слагаемых этой комбинации не сокращаются.

Умножим теперь (6) на и получим

где . Теперь из (7) и леммы 2.7 следует, что

Если мы подставим полученное нами выражение для в (5), то получим равенство

Но по (8) для всех *i*

Теперь, чтобы завершить доказательство, осталось подставить равенство в (4) и получить выражение для *f* в виде полиномиальной комбинации полиномов , где *все*  члены имеют мультистепень . Этот факт противоречит минимальности .

Применение критерия Бухбергера позволяет легко устанавливать, является данный базис базисом Грёбнера или нет.

**Упражнение.** Определить, являются ли следующее множество базисом Грёбнера для идеала, который он порождает.

*grlex-*упорядочение.

**Решение**.

Воспользуемся теоремой (критерием Бухбергера):  
Пусть *I* – некоторый идеал. Тогда базис идеала *I* является базисом Грёбнера в том и только том случае, когда для всех пар остаток от деления в любом порядке на равен нулю.

;

;

*.*

.

не является базисом Грёбнера.

является базисом Грёбнера для идеала, порожденного .

§6. Алгоритм Бухбергера.

В этом параграфе будет решаться следующая задача: как построить базис Грёбнера заданного идеала ?

**Теорема 2.5**. *Пусть дан некоторый ненулевой полиномиальный идеал Тогда базис Грёбнера для может быть построен за конечное число шагов с помощью следующего алгоритма:*

**Доказательство**. Сначала введем удобные обозначения. Если , то через и будем обозначать следующие идеалы:

Докажем сначала, что условие выполняется на каждом шаге алгоритма. Это верно в начале работы алгоритма. Далее, при каждом расширении множества мы добавляем остаток , где . Если , то и принадлежат . А так как мы делим на , то и остаток принадлежит ; значит . Кроме того, содержит исходный базис , а следовательно, является базисом идеала

Алгоритм заканчивает работу, когда , т.е. когда для всех . Следовательно, является базисом Грёбнера для по теореме 2.4.

Осталось доказать, что алгоритм в какой-то момент останавливается. Во время выполнения каждого основного цикла множество состоит из (старое ) и ненулевых остатков от деления S-полиномов от элементов из на , т.е.

так как . Мы утверждаем, что если , то строго меньше, чем . Докажем это. Пусть ненулевой остаток *r* от деления S-полиномов на был добавлен к . Тогда, так как *r* – остаток, не делится ни на один старший член элемента из , т.е. . Однако . Утверждение доказано.

По (1) идеалы , получающиеся в результате последовательных выполнений основного цикла, образуют возрастающую цепь в . Тогда условие обрыва возрастающих цепей утверждает, что эта цепь стабилизируется, т.е. условие станет выполнятся после конечного числа итераций основного цикла. Это означает, что условие станет выполняться и алгоритм остановится через конечное число шагов.

**Лемма 2.8**. *Пусть – базис Грёбнера полиномиального идеала , и пусть . Тогда также является базисом Грёбнера для .*

**Доказательство**. Мы знаем, что . Если , то . Следовательно, является базисом Грёбнера по определению.

Подберём константы т сделаем все старшие коэффициенты единицами, а также исключим из все , такие, что В результате мы получим *минимальный* базис Грёбнера.

**Определение 2.17**. *Минимальным базисом Грёбнера полиномиального идеала называется его базис Грёбнера , такой, что*

1. *для всех ;*
2. *для всех .*

Идеал может иметь несколько минимальных базисов Грёбнера. Поэтому введем понятие редуцированного базиса Грёбнера.

**Определение 2.18**. *Редуцированным базисом Грёбнера полиномиального идеала называется его базис Грёбнера , такой, что*

1. *для всех ;*
2. *никакой моном никакого не пренадлежит .*

Редуцированные базисы обладают следующим полезным свойством.

**Предложение 2.7**. *Пусть – полиномиальный идеал, и пусть задано некоторое мономиальное упорядочение. Тогда существует единственный редуцированный базис Грёбнера идеала .*

**Доказательство**. Пусть – некоторый минимальный базис Грёбнера для . Элемент называется *редуцированным для ,* если никакой моном из не принадлежит . Будем преобразовывать до тех пор, пока все его элементы не станут редуцированными.

Отметим сначала, что если редуцирован для , то редуцирован для любого другого минимального базиса Грёбнера (идеала ), содержащего и имеющего то же множество старших членов. Это утверждение справедливо, так как определение редуцированности оперирует только старшими членами.

Пусть . Положим и . Мы утверждаем, что также является минимальным базисом Грёбнера для Чтобы доказать это, отметим сначала, что . Поэтому . Так как , то - базис Грёбнера для (минимальность очевидна). Наконец, редуцирован для по построению.

Преобразуем таким образом каждый элемент из . Отметим теперь, что, что базис Грёбнера может измениться при каждом преобразовании, но, как только элемент стал редуцированным, то он и останется таковым при дальнейших преобразованиях элементов из (так как старший член не меняется). В конце концов мы получим редуцированный базис Грёбнера.

Докажем единственность. Пусть и – редуцированные (а, значит, и минимальные) базисы Грёбнера для . Минимальные базисы идеала имееют одно и то же множество старших членов:

Таким образом, для данного найдется , такой, что Если мы докажем, что из этого следует равенство , то тем самым и равенство , и единственность редуцированного базиса будут доказаны.

Рассмотрим разность . Эта разность принадлежит , а так как – базис Грёбнера, то . Но мы знаем также, что . Значит, старшие члены сократились, а оставшиеся члены не делятся ни на один элемент из так как редуцированные. Поэтому . Доказательство завершено.

Следствием данного предложения является *алгоритм проверки равенства идеалов*: порождают ли два множества один и тот же идеал? Для ответа достаточно задать мономиальное упорядочение и вычислить редуцированные базисы Грёбнера для . Идеалы совпадают в том и только том случае, когда совпадают их редуцированные базисы.

**УПРАЖНЕНИЯ**

**Упражнение 2.1.** Вычислить базисы Грёбнера для следующих идеалов:

Использовать *lex*-упорядочение.

**Решение.**

;

;

.

1. .  
   .
2. ;  
   :  
   .  
   .
3. . Заметим, что , а значит, можно «выбросить» из . Таким образом, на этом шаге .
4. :  
   .  
   .
5. :  
   .  
      
   .

Таким образом, базис Грёбнера идеала имеет вид:

;

;

;

1. :  
   .  
   .
2. .
3. :  
   .  
   .
4. .  
    Значит, .
5. :  
   .  
   .
6. .
7. Значит, .
8. :  
   .  
   .
9. :  
   .  
   .

Таким образом, базис Грёбнера идеала имеет вид:

;

;

;

1. .
2. :  
   .  
   .
3. .
4. ;  
   :  
   .  
   .
5. :  
   .  
   .

Таким образом, базис Грёбнера идеала имеет вид:

Глава 3. Применения базисов Грёбнера

§1. Теоремы об исключении и продолжении.

**Определение 3.1**. *Пусть дан идеал . Тогда l-м* ***исключающим идеалом*** *называется идеал в , равный*

**Теорема 3.1** (об исключении). *Пусть – идеал и G – его базис Грёбнера по отношению к lex-упорядочению с . Тогда для любого множество*

*является базисом Грёбнера l-го исключающего идеала .*

**Доказательство**. Зафиксируем *l* в интервале между 0 и *n.* Так как по построению, то достаточно доказать, что

(по определению базиса Грёбнера). Включение в одну сторону очевидно. Для доказательства другого включения нам достаточно доказать, что старший член , где – произвольный полином из , делится на некоторый старший член для некоторого .

Докажем это. Отметим сначала, что принадлежит также и *I, т.е.* делится на для некоторого (так как является базисом Грёбнера идеала *I*). Так как , то содержит только переменные . Теперь решающее замечание: так как используется *lex*-упорядочение с , то любой моном, содержащий хотя бы одну из переменных , больше всех мономов из . Поэтому из включения следует, что . Значит, .

**Теорема 3.2** (о продолжении). *Пусть , и пусть – первый исклющающий идеал для . Для каждого запишем в виде*

*где , а – ненулевые полиномы. Рассмотрим частичное решение . Тогда если , то существует , такое, что .*

§2. Суммы, произведения и пересечения идеалов.

1. Суммы идеалов.

**Определение 3.2**. *Пусть – идеалы кольца .* ***Сумма*** *идеалов – это множество*

**Предложение 3.1**. *Если – идеалы в , то также идеал в , причем – это наименьший идеал, содержащий . Кроме того, если и , то*

*.*

**Доказательство**. Прежде всего, . Далее, пусть . Тогда где . Имеем , так как и по определению идеала. Пусть теперь , а – произвольный полином. Тогда , где , . Имеем , так как и по определению идеала. Таким образом – идеал.

Если – некоторый идеал, содержащий , то содержит все элементы и все элементы . Так как – идеал, то он содержит все суммы . Значит, . Таким образом, каждый идеал, содержащий , обязан содержать и , т.е. – наименьший из идеалов с этим свойством. Наконец, если и , то идеал содержит . Поэтому . Обратное включение очевидно.  
Значит .

**Следствие 3.1**. *Пусть . Тогда*

**Теорема 3.3**. *Пусть – идеалы в . Тогда*

*.*

2. Произведение идеалов.

**Определение 3.3**. *Пусть – идеалы в . Тогда их* ***произведение***  *– это идеал, порожденный всеми полиномами вида где и .*

Такимобразом, произведение идеалов – это множество

Докажем, что это множество является идеалом. В самом деле, . Очевидно, что если , то и . Наконец, если и – произвольный полином, то

так как для всех . Отметим, что множество произведений не является идеалом – оно не замкнуто относительно сложения.

**Предложение 3.2**. *Пусть и . Тогда порождается множеством всех произведений образующих идеалов :*

**Доказательство**. Очевидно, что идеал, порожденный произведениями , содержится в . Докажем обратное включение. Любой полином из является суммой полиномов вида , где , . Но могут быть выражены через образующие:

где – некоторые полиномы. Тогда и любая сумма полиномов этого вида есть сумма , где .

**Теорема 3.4**. *Пусть – идеалы в . Тогда*

*.*

**Доказательство**. Пусть . Тогда для всех и всех . Если , то . Если для некоторого , то для всех . Значит . В обоих случаях .

Пусть теперь . Тогда или для всех , или для всех . Значит, для всех , т.е. .

3. Пересечение идеалов.

**Определение 3.4**.***Пересечение***  *двух идеалов идеалов – это множество полиномов, принадлежащих и , и .*

Справедливо следующее утверждение.

**Предложение 3.3**. *Пусть и – идеалы в . Тогда – тоже идеал.*

**Доказательство**. Прежде всего, , так как и . Далее, пусть . Тогда , так как . Аналогично, , откуда   
. Пусть теперь и - произвольный полином из *.* Тогда *,* так как *,* и *–* идеал.Аналогично, . Значит, .

**Теорема 3.5**. *Пусть и – идеалы из . Тогда*

Пустьи– идеалы в *.* Тогда рассматриваем идеал:

и находим его базис Грёбнера по отношению к *lex*-упорядочению, в котором . Тогда элементы этого базиса, не зависящие от , образуют базис Грёбнера идеала .

4. НОК и НОД полиномов.

**Определение 3.5**. *Полином называется* ***наименьшим общим кратным*** *полиномов , если*

1. *делит и делит ;*
2. *делит любой полином, который делится и на , и на .*

**Предложение 3.4**.

1. *Пересечение двух главных идеалов является главным идеалом.*
2. *Если , то , где*

Использование этого утверждения и алгоритма для вычисления пересечения двух идеалов дает нам *алгоритм для вычисления наименьшего общего кратного* двух полиномов. А именно, пусть даны полиномы . Мы находим пересечение . По предложению 3.4 пересечение является главным идеалом и его образующий элемент и есть наименьшее общее кратное полиномов .

**Предложение 3.5**. *Пусть* . *Тогда*

Это дает нам *алгоритм для вычисления наибольшего общего делителя* двух полиномов. А именно, мы вычисляем , использую алгоритм вычисления НОК, а затем делим произведение на , используя алгоритм деления.

ПРИЛОЖЕНИЕ А

Псевдокод

Псевдокоды используются в математике и информатике для описания алгоритмов.

Алгоритм – это набор инструкций для выполнения определённых численных или символьных вычислений. Алгоритм имеет *вход* или *входные данные,* т.е. информацию, которую он обрабатывает, и *выход* – результат его вычислений. На каждом шаге очередная операция полностью определена текущим состоянием алгоритма. Алгоритм прекращает работу после конечного числа шагов.

Большинство алгоритмов содержит следующие специальные структурные компоненты:

* Структуры повторения (циклы);
* Структуры условного перехода.

Эти структуры, а так же некоторые другие компоненты псевдокода будут описаны ниже.

1. Вход, выход, переменные, константы.

Вход и выход алгоритма указывается в двух строчках перед началом алгоритма. Входу и выходу присвоены имена в соответствии с правилами математических обозначений. Иногда указывается тип данных (если не указан, считается, что тип данных понятен из контекста). Переменные так-же имеют свои имена, их типы определяются самим процессом вычисления. Булевы константы *true* и *false* используются для обозначения истинности или ложности утверждений.

2. Оператор присваивания.

Оператор *присваивания* является наиболее часто встречаемым типом инструкций. Правило записи этого оператора таково:

Перед присваиванием вычисляется выражение справа, если переменная хранила значение, оно стирается и заменяется новым.

3. Операторы цикла.

В изложенных в работе алгоритмах используются 3 типа структур повторения.

Первая и наиболее часто встречаемая. «Действие» - это и есть последовательность инструкций, которая повторяется. «Условие» - утверждение, которое может быть истинным или ложным на каждом шаге алгоритма.

Вторая использованная в этой работе структура. Действие будет повторяться, пока условие ложно. Действие выполнится как минимум один раз.

Последняя структура повторения. Она означает «выполняй действие для каждого элемента ». – конечное множество объектов. Выполняется фиксированное количество раз (столько, сколько объектов в ).

4. Условный оператор.

В данной работе мы используем только один тип условного оператора, который записывается так:

Если условие истинно в тот момент, когда выполняется , то выполняется «действие1» (один раз). В противном случае, выполняется «действие2» (тоже один раз). В некоторых случаях *опускается и* «действие2». Тогда если условие ложно, ничего не выполняется.