Функции. Счетные множества. Операции.

Определим функцию как *специальное* отношение на множестве

*A*  *B* , на которое наложено

*дополнительное ограничение*. Отношение *f* на множестве

*A*  *B*

называется *функцией* из *A* в *B* и

обозначается

*f* : *A*  *B* , если для *каждого*

*a*  *A*

существует *единственный* элемент

*b*  *B*

такой, что

(*a*, *b*)  *f* . Это дополнительное свойство называется *однозначностью* или *функциональностью*. Запишем

его в символьном виде:

*a*  *A* (*a*,*b*)  *f* 

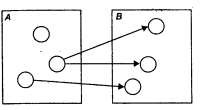
(*a*,*c*)  *f*  *b*  *c* . Если

*f* : *A*  *B*

– функция, и

(*a*.*b*)  *f* ,

то говорят, что *b*  *f* (*a*) . Рассмотрим отношение, заданное графически.



Данное отношение не является функцией, поскольку, во-первых, *не все элементы* множества *A* участвуют в отношении и, во-вторых, не выполняется свойство *однозначности*. Если для отношения не выполняется *хотя бы одно* из указанных свойств, отношение не является функцией.

Множество *A* называется *областью определения* функции и обозначается

*Df* . Множество

*E f*  {*b*  *B* | *a*  *A* \_ *b* 

*f* (*a*)}

называется *множеством значений* функции *f*. Множество *B* называется

*областью значений* функции.

Функция

*f* : *A*  *B*

называется также *отображением* множества *A во* множество *B*. Если речь идет о

функции как отображении, то элемент

*b*  *B*

называют *образом* элемента *a* , а элемент

*a*  *A* –

*прообразом* элемента *b* при отображении *f*. При этом множество *Ef*

называют *образом* множества

*Df* , а

множество *Df* – *прообразом* множества *Ef* .

Дадим несколько определений, связанных с функциями. Функция называется *инъективной* (*инъекцией*),

если из

*f* (*a*1 ) 

*f* (*a*2 )

следует

*a*1  *a*2 . Функция *f* называется *отображением «на»* или *сюръективной*

функцией (*сюръекцией*), если для *каждого b*  *B* существует некоторое *a*  *A* , такое что

*f* (*a*)  *b* . Функция, которая является *одновременно* и *инъективной* и *сюръективной* называется

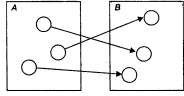
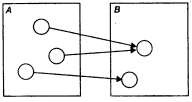
*биективной* (*биекцией*). Если

*f* : *A*  *B*

биективная функция, то говорят, что *f* осуществляет *взаимно*

*однозначное соответствие* между множествами *A* и *B.*

Рассмотрим геометрическую интерпретацию некоторых видов функций.



*Сюръекция, не инъекция*. *Биекция*.

Рассмотрим для примера функции, отображающие

1. *f*  *e x* , функция инъективна, но не сюръективна;

1

1. *f*  *x*3  *x* , сюръективна, но не инъективна;

2

1. *f*  *x* 2 , не инъективна, не сюръективна;

3

*R*  *R* , *fi* : *R*  *R* , *i*  1,4 :

1. *f* 4  3*x*  2 , биективна;

Рассмотрим еще пример.

Пусть заданы множества: *А = {1,2,3,4}* и *B = {5,6,7,8}*, Отношения из *A* в *B* заданы парами:

*R = {(1, 5), (2, 6), (2, 8), (3, 7)}*

*S = {(1,5),(2,6),(3,6),(4,8).*

*T = {1,7),(2,5),(3,8),(4,6)}*

Отношение *R* не является функцией, т. к. не выполняется свойство *однозначности*: элементу *2* из множества *A* соответствует два элемента из множества *B*, *6* и *8,* и не все элементы множества *A* участвует в отношении.

Отношение *S* является функцией:

*f* : *A*  *B* . Неинъективной, т. к. двум различным элементам *2* и *3* из

множества *A* соответствует один и тот же элемент *6* из множества *B* и не сюръективной, т. к. множество

значений *Ef* не совпадает с множеством *B*: *E f*  *B* .

Отношение *T* является функцией:

*f* : *A*  *B* и это биекция.

Отметим, что отобразив,

*f* : *A*  *E f*

мы всегда получим сюръективную функцию. Например,

*f*1 : *R*  *R*0 , где

*f*  *e x* является сюръективной функцией.

*Тождественной функцией* (*отображением*) множества *A на себя* называется функция *e A* : *A*  *A* , такая что *a*  *A* \_*e A* (*a*)  *a* .

1

Как и для отношения, определим для функции операцию *композиции*. Пусть

*f* : *A*  *B*

и *g* : *B*  *C* .

Тогда композиция двух функций *g* и *f* есть функция и она отображает множество *A* во множество *С*:

*g* ∘ *f* : *A*  *C* . Если *a*  *A* , то (*g* ∘ *f* )(*a*)  *g* *f* (*a*). Композиция функций как и композиция отношений

выполняется *справа налево*.

Композиция функций *ассоциативна*. И в общем случае *не коммутативна* :

1. *h* ∘ (*g* ∘ *f* )  (*h* ∘ *g*) ∘ *f* ;

2. *g* ∘ *f*  *f* ∘ *g* ;

Рассмотрим второе свойство на примере.

Пусть

*f* : *R*  *R*, *f* (*x*)  *x* 2 и

*g* : *R*  *R*, *g*(*x*)  2*x*  3 . Тогда:

(*g* ∘ *f* )(*x*)  *g*( *f* (*x*))  2*x* 2  3 ;

( *f* ∘ *g*)(*x*)  *f* (*g*(*x*))  (2*x*  3)2  4*x* 2  12 *x*  9 ;

Как видим, в *общем случае*,

*g* ∘ *f*

 *f* ∘ *g* .

В дальнейшем, значок композиции может быть опущен.

*Отметим также, что композиция биективных функций есть также биективная функция.*

Пусть задана функция

*f* : *A*  *B* и

*f* 1

– отношение, обратное *f* . Если отношение

*f* 1

является

функцией, то ее называют *обратной* функцией к *f* и она отображает множество *B* во множество *A*

*f* 1 : *B*  *A* .

*Функция f имеет обратную тогда и только тогда, когда f – биекция, причем биекцией.*

*f* 1

*также является*

Найдем обратную функцию для функции

*y*  3*x*  2 .

*f* 1  {( *x*, *y*) | ( *y*, *x*)  *f* }  {( *x*, *y*) | *x*  3*y*  2}  {( *x*, *y*) | *y*  (*x*  2) / 3} *или*

*f* 1  {( *y*, *x*) | (*x*, *y*)  *f* }  {( *y*, *x*) | *y*  3*x*  2}  {( *y*, *x*) | *x*  ( *y*  2) / 3}  {( *x*, *y*) | *y*  (*x*  2) / 3}

*f* 1 (*x*)  (*x*  2) / 3};

Если

*f* : *A*  *B*

– биекция, то справедливы следующие соотношения:

*f* 1 ( *f* (*a*))  *a*  *e*

*a*

*f* ( *f* 1(*b*))  *b*  *e*

*b*

*(1)*

Наша функция является биекцией, поэтому для нее выполняется соотношение *(1)* для композиций:

*f* 1 ( *f* (*x*))  ((3*x*  2)  2) / 3  *x*  *e* .

*x*

*f* ( *f* 1( *y*))  3( *у*  2) / 3  2  *y*  *e*

*y*

*Сужением функции*

образом:

*f* : *A*  *B*

на множество *M* называют функцию

*f M* , определяемую следующим

*f M*  {(*a*.*b*) | (*a*, *b*)  *f*

 *a*  *M* }.

Может случиться так, что сама функция, заданная на множестве *A* не имеет обратной функции, но сужение этой функции на некоторое подмножество множества *A*, на котором она биективна, уже имеет

обратную функцию. Например, обратной функции функция

*f*  *x* 2

на множестве *R* не имеет. Но сужения

этой функции на

*R*0 и

*R*0

уже имеют обратные функции:

*f*  : *R*0

 *R*0

\_ *f* 1:: *R*

 *R*0

, *f* 1 (*x*)  ;

*f*  : *R*0

*x*

 



0

 *R*0 \_ *f*

1

 0

:: *R*

 *R*0

, *f* 1 (*x*)  

Функции могут задаваться различными способами: *аналитическим (в виде формулы), табличным, графическим, с помощью рекурсивной процедуры*.

*x*



Рассмотрим *рекурсивный способ* задания функции. Рекурсивная процедура задает функцию, определенную на множестве *Z* 0 (*или* \_ *N* ) следующим образом:

1. задается значение *f(1)* или *f(0*);
2. значение *f(n+1)* определяется через суперпозицию *f(n)* и других, считающихся известными, функций;

Зададим, например, функцию

### *f* (0)  1;

*f* (*n*  1)  *a*  *f* (*n*);

*f* (*n*)  *an* , определенную на множестве

*Z* 0 , рекурсивно:

Как видим, указанная функция может быть задана как аналитически (с помощью формулы), так и рекурсивно.

Функция факториал

### *f* (0)  1;

*f* (*n*)  *n*! , определенная рекурсивно, имеет вид:

*f* (*n*  1)  (*n*  1)  *f* (*n*);

Последовательности чисел *Фибоначчи* и *Каталана* также могут быть заданы рекурсивно.

Большинство языков программирования позволяет реализовать рекурсивную процедуру (*функция вызывает саму себя*).

*Взаимно однозначным соответствием* между *двумя непустыми множествами A* и *B* называется такое

*правило* (*или закон*) *f*, по которому каждому элементу *a*  *A* ставится в соответствие единственный элемент

*f* (*a*)  *B*

и для любого элемента

*b*  *B* существует единственный элемент

*a*  *A* , такой что

*f* (*a*)  *b* ,

другими словами, *f* задает биекцию между *A* и *B*.

Множество *A* и *B* называются *равномощными* (обозначается *A*  *B* ), если между ними можно установить взаимно однозначное соответствие. Отношение равномощности множеств обладает свойством транзитивности:

*A*  *B*  *B*  *C*  *A*  *C* и, очевидно, является отношением *эквивалентности*.

Между *непустыми конечными* множествами *A* и *B* существует взаимно однозначное соответствие тогда и только тогда, когда | *A* || *B* | .

Вернемся к понятию мощности множеств.

Как мы уже говорили, *пустое множество есть конечное множество мощности 0*.

Если существует взаимно однозначное соответствие между множеством *А* и множеством *{1, 2, 3,..., n}*, то

*А* есть *конечное множество мощности n*.

За основу для сопоставления *бесконечных* множеств берется множество натуральных чисел *N*.

Множество *A* называется *счетно бесконечным*, если существует взаимно однозначное соответствие между множеством *A* и множеством натуральных чисел *N*.

Множество называется *счетным*, если оно *конечно* или *счетно бесконечно*. *Подмножество счетного множества является счетным.*

Множество целых чисел *Z* является *счетным*. Установим взаимно однозначное соответствие следующим образом:

*f* : *Z*  *N*

### *Z*. : ..

 *n*  4  3  2  1 0 1

### 2 3 4 ... *n*

         

### *N* : ...2*n*  1 9 7 5 3

1 2 4 6

8 ...2*n*

Взаимно однозначное соответствие осуществляется по правилу:

*f* (*z*) {

2*z*,

*z*  *N*

### 2 | *z* | 1,

*z*  *Z*0

Установив взаимно однозначное соответствие между множествами *Z* и *N*, мы *«занумеровали»* элементы множества *Z* в последовательности *0, 1,-1, 2, -2,..* Например, число *-5* будет иметь *11* порядковый номер в этой последовательности, т. е.  5  11 .

Счетным является множество *рациональных чисел Q*. Укажем, например, правило построения *взаимно*

*однозначного соответствия* между

*Q*0

и *N*. Обходя таблицу по направлениям стрелок, получим

следующую последовательность: 1/1, 1/2, 2/3, 2/1, 3/1, 3/2,…. Обход таблицы можно осуществлять и другими способами.

### 1  1

1 2

### 

2  2

### 1 3



### 3  3

1 2

### 4  4

1 3

### 

5 ...

### 1

1  1

### 3 4

 

### 2 2

### 5 7

 

###  3 3

4 5

### 

 4  4

### 5 7

1 

### 5

...

### ...

...

### ...

Если бесконечное множество *A* нельзя поставить во взаимно однозначное соответствие с множеством натуральных чисел *N*, то множество *A* называется *несчетным*.

Множество всех действительных чисел интервала *(0,1)* несчетно. Эта теорема доказана *Г. Кантором*. Мощность несчетных множеств называется *континуум*. Множество всех действительных чисел *R* также *несчетно*.

Если бесконечное множество можно привести во взаимно однозначное соответствие с *несчетным*

множеством, то такое бесконечное множество также является *несчетным*.

*Принцип Дирихле*

Пусть *f* : *A*  *B* – функция, причем *A* и *B* – конечные множества. Предположим, что *A* состоит из *n*

элементов:

*a*1, *a*2 ,...,*a n* . Принцип *Дирихле* гласит, что если *|A| > |B|*, то, по крайней мере, одно значение *f*

встретится более одного раза. Иначе, найдется пара элементов *ai*

 *a j* , для которых

*f* (*ai* ) 

*f* (*a j* ) .

Принцип *Дирихле* иногда называют *принципом клеток* и формулируют в легко запоминающей форме:

*Невозможно рассадить 10 кроликов в 9 клетках так, чтобы в каждой клетке сидел один кролик.*

Рассмотрим задачи на применение принципа *Дирихле*. *Задача 1*.

В сентябре *21* день был солнечным. Доказать, что в сентябре *два дня подряд* была солнечная погода;

*Решение*.

Подсчитаем количество *пасмурных* дней в сентябре: *30* - *21 = 9*. Они разбивают *все* дни сентября на промежутки из *солнечных* дней. Всего пасмурных дней *9*, они могут разбить все дни не более, чем на *10*, промежутков. Даже при *11* солнечных днях по принципу *Дирихле* хотя бы один промежуток состоял бы хотя бы из двух дней. Мы отражаем *дни в промежутки*, по количеству их больше чем промежутков, поэтому хотя бы два дня подряд была солнечная погода.

*Задача 2.*

В коробке лежат *2* пары *одинаковых* ботинок. Какое наименьшее число ботинок нужно вынуть не глядя, чтобы среди них обязательно оказалась одна пара?

*Решение*.

В таких задачах мы рассматриваем *худший* случай: за первые *два раза* мы вытащили *2 левых* или *2 правых* ботинка ( левый и правый – это лучший случай). Тогда, какой ботинок мы бы не вытащили в третий раз, мы обязательно получим пару одинаковых ботинок. Итак, ответ *3*.

Определим *операцию* как функцию *частного вида*.

*Операцией над множеством A* называется функция

*f* : *An*  *A* .

Поскольку мы определили операцию как функцию, то результат операции *однозначно* определен и т. к. *E f*  *A* операция *замкнута* на *A*. Определенная таким образом операция имеет *арность* (*порядок*) *n*. При *n = 1* операция называется *унарной*, например, операция отрицания. При *n = 2* операция называется

*бинарной*. Рассмотрим бинарную операцию.

*Бинарной операцией* на множестве *A* называется функция

*f* : *A*  *A*  *A* . *Таким образом, всякой*

*упорядоченной паре элементов множества A по некоторому закону ставится в соответствие вполне определенный элемент этого же множества.*

Как мы уже говорили, бинарная операция обладает свойством *замыкания*, а множество *A* свойством *замкнутости* относительно бинарной операции, поскольку значение операции на двух элементах *a* и *b* множества *A* также является элементом множества *A*.

Примерами бинарных операций на множестве натуральных чисел *N* и целых чисел *Z* является *сложение* и *умножение*. В то же время множество *N* не обладает свойством замкнутости относительно вычитания, *N* и *Z* относительно деления. Векторное умножение векторов пространства, умножение квадратных матриц порядка *n*, операции пересечения и объединения множеств являются бинарными операциями на соответствующих множествах.

Одним из способов задания бинарных операций является *таблица Кэли*. Слева и сверху в квадратной таблице записываются все элементы множества в одном и том же порядке. На пересечении строки, соответствующей элементу *a,* и столбца, соответствующего элементу *b*, записывается результат операции над *a* и *b*. Например, для операции конъюнкции таблица *Кэли* выглядит так:

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

В общем случае будем обозначать бинарную операцию значком \*.

*Определим свойства бинарных операций*.

* Бинарная операция называется *коммутативной*, если справедливо равенство:

*a* \* *b*  *b* \* *a* , *a*, *b*  *A* .

Коммутативными являются операции сложения и умножения на числовых множествах, операции сложения матриц одного порядка, операции объединения и пересечения множеств и т.д.

Некоммутативны операции: вычитания на *Z*, деления на *R*, разности множеств, векторное произведения векторов, произведение матриц порядка *n* при *n*  2 и т. д.

* Бинарная операция называется *ассоциативной*, если справедливо равенство:

(*a*  *b*) \* *c*  *a*  (*b*  *c*) , *a*, *b*, *c*  *A* .

Ассоциативными являются операции: сложение и умножение на *Z*, умножение матриц, объединение и пересечение множеств и т. д.

Не ассоциативными являются операции разности множеств, векторное умножение векторов пространства.

*Если операция, определенная на A, ассоциативна, то результат ее последовательного применения к n элементам множества не зависит от расстановки скобок.*

* Пусть дано множество *A,* на котором определенны две бинарные операции: ,  .

Тогда, если для всех выполняется: (*дистрибутивность слева*)

(дистрибутивность справа)

то говорят, что операция  *дистрибутивна* по отношению к операции  .

Операция *умножения* дистрибутивна относительно *сложения*, но не наоборот. Операции

дистрибутивны относительно друг друга.

##### , 

* Пусть операция  на множестве *A* и произвольный элемент *a*  *A* таковы, что *a*  *a*  *a* . В этом случае говорят, что элемент *a идемпотентен* по отношению к операции  . Любое множество идемпотентно по отношению к операциям объединения и пересечения.
* В ряде случаев, множество *A*, на котором определена бинарная операция, обладает *единичным*

*э*лементом, т.е. таким элементом *e*, что *a*  *e*  *e*  *a*  *a* для *a*  *A* . Единичный элемент, если он

существует, *единственен*. Например, на множествах *Z*, *R* относительно сложения единичным элементом является *0*, относительно умножения - *1*.

*Существуют 3 формы записи выражений: инфиксная*, *префиксная* и *постфиксная* (*суффиксная*).

Выражение находится в *инфиксной* (*обычной*) записи, если в нем знак бинарной операции записывается

*между* операндами, на которые она действует, например: *a + b*.

Выражение находится в *префиксной* записи, если в нем знак операции *непосредственно предшествует операндам*, на которые она действует, например, *+ ab*.

Выражение находится в *постфиксной* записи, если в нем знак операции *следует непосредственно за операндами*, на которые она действует, например, *ab +*.

*Оператором* называется функция, которая отображает функции в функции. Примеры операторов:

дифференцирования - *d* , интегрирования - и т. д.



*dx*

Булеан. Прямое произведение множеств.

Множество *всех* подмножеств множества *A* называется *булеаном* множества *A* и обозначается *P(A)* или

2 *A* .

Для *конечного* множества *A* мощность его булеана: | *P*( *A*) | 2 *A* .

Пусть дано множество *A* = *{1, 2, 3}*. Найдем его булеан.

*P*( *A*)  {,{1},{2},{3},{1,2},{1,3}, (2,3){1,2,3}} .

Как видим, элементами булеана являются множества и как мы уже отмечали, пустое множество  и само множество *A* являются подмножествами *A* и входят в булеан. Мощность булеана нашего множества *A*

равна: | *P*( *A*) | 2 *A*  23  8.

Над множествами определена еще одна операция – *декартово произведение* множеств. Дадим вначале определение *упорядоченной* пары.

*Упорядоченная пара на множествах A* и *B*, обозначаемая записью (*a, b*), определяется не только

самими элементами

*a*  *A*

и *b*  *B* , но и порядком, в котором они записаны. И в этом состоит ее

существенное отличие от неупорядоченной пары. В неупорядоченной паре порядок элементов не существенен, она берется в фигурные скобки и *{1,3} = {3,1}* (как мы знаем, это одно и то же множество).

Если *A* = *B*, то говорят об упорядоченной паре на множестве *A*.

Равенство упорядоченных пар определяется следующим образом:

(*a*,*b*)  (*c*, *d* )  *a*  *c*  *b*  *d* .

В общем случае, (*a*, *b*)  (*b*, *a*) .

Простейший и важнейший пример использования упорядоченных пар дает аналитическая геометрия.

Если на плоскости введена некоторая прямоугольная система координат, то каждая точка плоскости однозначно задается упорядоченной парой действительных чисел – координатами этой точки. Точка с координатами *(1, 3)* совсем не то же самое, что точка с координатами *(3,1)*:

(1,3)  (3,1) .

Обобщенным понятием упорядоченной пары является *упорядоченный n-набор*, или *кортеж*. Часто кортеж *называют вектором. Число n называется длиной кортежа (или размерностьюкортежа*), а

элемент

*ai* - *i*-*ой проекцией* или *i*-*ой компонентой кортежа*. В отличие от конечного множества {*a*1,...,*an* }

кортеж

(*a*1,..., *an* )

на множествах

*A*1 ,. , *An*

характеризуется не только входящими в него элементами

*a*1  *A*1...*an*  *An* , но и порядком, в котором они перечисляются, и *компоненты кортежа могут совпадать*. Как и для упорядоченных пар, роль порядка в кортеже фиксируется определением равенства

кортежей. Два кортежа (*a*1 ,..., *an* )

Пусть *A* и *B* – два множества.

и (*b*1,...,*bn* )

на множествах

*A*1 ,..., *An* равны, если: *ai*  *bi* ,

*i*  1, *n* .

*Прямым* (*декартовым*) *произведением* двух множеств *A* и *B* называется множество упорядоченных пар, в котором первый элемент каждой пары принадлежит множеству *A*, а второй принадлежит множеству *B*:

*A*  *B*  {(*a*,*b*) | *a*  *A*  *b*  *B*}.

Например, пусть заданы множества *A* = {1,2} и *B* = {3,4}. Тогда

*A*  *B*  {(1,3), (1,4), (2,3),(2,4)}.

В общем случае,

*A*  *B*  *B*  *A* .

Множество всех кортежей длины *n* на множествах

*A*1 ,..., *An*

называют декартовым (прямым)

произведением множеств

*A*1 ,..., *An*

и обозначают

*A*1 ... *An* .

*A*1 ...  *An*  {(*a*1 ,...,*an* ) | *a*1  *A*1  ...  *an*  *An* }.

Для нашего примера зададим еще множество *C* = {5,6}.

Тогда,

*A*  *B*  *C*  {(1,3,5), (1,3,6), (1,4,5), (1,4,6), (2,3,5),(2,3,6),(2,4,5),(2,4,6)}. Получили *8*

упорядоченных троек. Если все множества

*Ai* ,

*i*  1, *n* , равны между собой, то указанное декартово произведение называют *n*-

*ой декартовой степенью множества A* и обозначаются

*An* . Таким образом, *степенью множества A*

называется его прямое произведение самого на себя:

*An*  *A* ...  *A* .

*n*  *paз*

Если мощность множества *А* равна *n*, а мощность множества *B* равна *m*, то мощность прямого произведения: | *A*  *B* | *n*  *m* .

Соответственно, | *An* || *A* |*n* .

ТЕМА №2.***ОТНОШЕНИЯ.***

Понятие отношения используют для обозначения *связи* между объектами или понятиями.

Для строго математического описания любых связей между элементами двух множеств введем понятие бинарного отношения.

Пусть *A* и *B* – два множества. *Бинарным отношением R* из множества *A* во множество *B* (между множествами *A* и *B*) называется подмножество прямого произведения *A* и *B*:

*R*  *A*  *B* .

Для бинарных отношений обычно используется *инфиксная* форма записи:

*aRb* : (*a*,*b*)  *R*  *R*  *A*  *B* , означающая, что элементы *a* и *b* находится в отношении *R*. Если *A = B*, то говорят, что *R* есть бинарное отношение на множестве *A*.

*Таким образом, бинарное отношение R – это множество, элементами которого являются*

*упорядоченные пары, связанные данным отношением на множестве*

*A*  *B*

*или на множестве*

*A*2 ( *A*  *A*) *.*

По аналогии с бинарными отношениями введем понятие *n* - *местного* (*n*-*арного*) отношения: *n – местным*

*отношением R* на множествах

*A*1 , *A*2 ,..., *An* называется подмножество прямого произведения множеств:

*R*  *A*1  *A*2  ... *An* .

Наиболее изученным и употребляемым типом отношений являются *бинарные* отношения. Будем рассматривать бинарные отношения, слово «*бинарные*» в дальнейшем может быть опущено.

Рассмотрим примеры бинарных отношений:

1. Все множество *A*  *B*

есть отношение множеств *А* и *В,* называемое

*универсальным отношением*. Если мощность множества

| *A*  *B* | *m* , то существует 2*m*

различных

подмножеств множества

*A*  *B*

*A*  *B*

и, следовательно, существует 2*m*

различных отношений на множестве

1. Отношения

# , ,

, , ,

 , определенные на *соответствующих числовых*

*множествах*, являются примерами бинарных отношений. Например, зададим отношение на множестве целых чисел *Z* следующим образом: *R*  {(*a*, *b*)  *Z* 2 | *a*2  *b*2  25} и пара (*3,4*) принадлежит данному отношению *R*.

1. Отношения *«быть сыном»*, *«быть братом»*, *«быть соседом»* и т. д.,

определенные на множестве людей. Если *А* – множество людей, то бинарное отношение на множестве *А*.

*R*  {(*a*,*b*)  *A*2 | *a*

* *сын b*}есть

1. Пусть *А* – множество товаров в магазине, а *N* – множество натуральных чисел.

Тогда  – отношение множеств *А* и *N*.

1. Отношения параллельности и перпендикулярности прямых, отношения подобия и равенства треугольников и т. п. отношения, используемые в геометрии.

Отношение между *конечными* множествами может задаваться различными способами**:**

1. *с помощью подходящего предиката;*
2. *как матрица (или таблица);*
3. *как орграф;*
4. *как множество упорядоченных пар (перечислением);*

Рассмотрим пример.

Пусть на множестве *A = {2,3,4,5,6}*задано отношение *aR*1*b* : *a*  *b* . Зададим его различными способами.

1. С помощью предиката: *R*1  {(*a*,*b*)  *A* *A* | *a*  *b*} ;
2. Построим матрицу отношения. Перенумеруем элементы множества *A* в соответствии с порядком, в котором они записаны, тогда отношение можно представить матрицей *R,* элементы которой определяются условием:

1,

*R*[, *j* {

0,

*если если*

*ai Ra j*

(*ai Ra j* )

Другими словами, на пересечении *i*- *ой* строки и *j* - *го*столбца стоит *1*, если *i*–*ый* элемент множества *A*

находится в отношении *R* c *j*–*ым* элементом множества и *0* в противном случае. Таким образом, матрица отношений состоит из *0* и *1* и является *булевой матрицей*. При табличном представлении в первой строке и первом столбце указаны элементы множества *A* в порядке их следования. Заполнение ячеек таблицы аналогично матричному заполнению. Построим матрицу:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  0   | 0 | 0 | 0 | 0   |
|  1 | 0 | 0 | 0 | 0 |

### *R*   1 1 0 0 



0

1



 1 1 1 0 0



###  1 1 1 

1

0



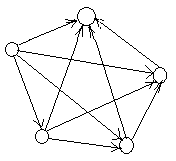
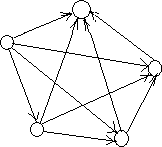
1. Перечислением пар:

*R*1  {(3,2), (4,2),(5,2),(6,2),(4,3),(5,3),(6,3),(5,4),(6,4),(6,5)}.

1. Построим орграф. Вершинам ориентированного графа соответствуют элементы множества *A,* дуга имеет направление от *a* к *b*, если элемент *a* находится в отношении с элементом *b*: *aRb*. Если *a* не находится в отношении с *b*, дуга отсутствует, если в отношении имеется пара (*a,a*) , то ей соответствует петля. На рисунке *1* приведен орграф нашего отношения *R*1 .

2

6



3

5 4

*Рис.1. Орграф отношения*

*R*1 *.*

Введем несколько понятий, связанных с отношениями. Пусть *R* есть отношение на *A*  *B* : *R*  *A*  *B* , (*a*, *b*)  *R*

и *a*  *A* , *b*  *B* .

*Область определения отношения:*

*D*(*R*)  {*a* | (*a*, *b*)  *R*}

– множество всех *первых элементов a* упорядоченных пар из *R*. Для нашего

отношения *R*1  *A* *A* , *A = {2,3,4,5,6}*:

*D*(*R*1)  {3,4,5,6}

*Множество значений отношения:*

*E*(*R*)  {*b* | (*a*, *b*)  *R*} – множество всех вторых элементов *b* упорядоченных пар из *R*.

*E*(*R*1)  {2,3,4,5} .

*Обратное отношение*:

Обратное отношение *R*1 на множестве *B*  *A* определяется следующим образом:

*R*1  {(*b*, *a*) | (*a*,*b*)  *R*}.

Таким образом, (*b*, *a*)  *R* 1 тогда и только тогда, когда (*a*, *b*)  *R* , другими словами элементы, но в *обратном порядке*.

Можно записать *R* 1 в другом виде:

*R*1  {(*a*,*b*) | (*b*, *a*)  *R*}.

Если задано отношение на двух множествах: *R*  *A*  *B* , то тогда *R* 1  *B*  *A* . Если мы рассматриваем отношение на множестве *A*: *R*  *A*  *A* , то *R* 1  *A* *A* . Для нашего отношения *R*1 обратным отношением будет:

*R* 1  {(*b*, *a*) | *a*,*b*  *A*  *a*  *b*}, или в иной записи:

1

*R* 1  {(*a*,*b*) | *a*,*b*  *A*  *b*  *a*}

1

*R* 1

связывает те же

Найдем обратное отношение для нашего отношения

# *R* 1  {(2,3),(2,4).(2,5),(2,6),(3,4),(3,5),(3,6),(4,5),(4,6),(5,6)}.

1

*R*1 .

Мы уже определяли *универсальное отношение*для множеств *А* и *В* как множество их прямого произведения:

*U*  *A* *B*  {(*a*,*b*) | *a*  *A*  *b*  *B*}.

Если отношение задано на множестве *A*, то *U A*  *A*  *A*  {(*a*, *b*) | *a*  *A*  *b*  *A*} .

Матрица универсального отношения состоит из одних единиц.

Определим *тождественное отношение*, называемое также *диагональным отношением* на множестве *A***:**

*I*  {(*a*, *a*) | *a*  *A*}.

Для нашего множества *A={2,3,4,5,6}*:

*I A*  {(2,2),(3,3),(4,4)(5,5),(6,6)}.

Операции над отношениями.

Поскольку *R –* это *множество* упорядоченных пар, то над отношениями можно выполнять операции объединения, пересечения, разности, дополнения.

Зададим на нашем множестве *A={2,3,4,5,6}* еще одно отношение *R*2 :

*R*2  {(*a*,*b*) | *a*,*b*  *A*  *a*  *b*}. Отношению *R*2 принадлежат пары:

*R*2  {(2,2), (3,3),(4,4),(5,5),(6,6)}.

Тогда:

*R*1  *R*2  {(*a*,*b*) | *a*,*b*  *A*  *a*  *b*} *R*1  *R*2 

*R*1 \ *R*2  *R*1

*R*2 \ *R*1  *R*2

Введем дополнение отношения:

*R*  {(*a*,*b*) | (*a*,*b*)  *R*  (*a*,*b*) *U*}. Здесь *U* – универсальное отношение. Тогда:

*R*1  {(*a*,*b*) | *a*,*b*  *A*  *a*  *b*} .

*R*2  {(*a*, *b*) | *a*, *b*  *A*  *a*  *b*}.

Для бинарных отношений определена *специальная* операция, называемая *композицией.*

Пусть *R* – отношение между множествами *A* и *B*,

*R*  *A*  *B*

и *S* – отношение между множествами *B* и *C*,

*S*  *B*  *C* . *Композицией отношений S* и *R* называется бинарное отношение между *A* и *C*, которое обозначается *S* ∘ *R* и определяется следующим образом:

*S* ∘ *R*  {(*a*,*c*) | *a*  *A*,*c* *C*   *b*  *B* (*a*,*b*)  *R*  (*b*,*c*)  *S*}. Расшифруем нашу запись:

*Композицией отношений S* и *R* называется множество упорядоченных пар (*a*, *c*) , таких что *a*  *A* и *c*  *C*

и существует такой элемент *b*  *B* , что упорядоченная пара (*a*,*b*)  *R*

и упорядоченная пара (*b*, *c*)  *S* .

Новое отношение устанавливает связь между элементами множеств *A* и *C* следующим образом: оно действует из *A* в *B* посредством *R*, а затем из *B* в *C* посредством *S*, используя элементы из множества *B* в качестве *посредников*.

*Для двух бинарных отношений R и S, заданных на множестве A, их композиция является бинарным отношением на том же множестве.*

Композицию

*R* ∘ *R*

бинарного отношения *R* на *некотором множестве* с самим собой называют

*квадратом бинарного отношения R* и обозначают

*n* раз называют *n*-*степенью* бинарного отношения.

*R* 2 , соответственно, композицию отношения *R* на себя

*Обратите внимание на запись определения S* ∘ *R , из записи следует, что отношения применяются*

***справа налево****: вначале применяется R, затем S, т.е. справедливо соотношение:*

(*S* ∘ *R*)(*a*)  *S* (*R*(*a*)) .

Данная терминология является международной.

Например, для отношений *R = {(3,1),(6,2)}* и *S* = *{(1,3),(2,2),(3,1)},* заданных на множестве *A* = *{1,2,3,4,5,6}*, их

композиции соответственно равны: *S* ∘ *R*  {(3,3), (6,2)}, *R* ∘ *S*  {(1,1)}.

Если два отношения, *R* и *S* заданы перечислением пар, их композицию следующему правилу:

*S* ∘ *R*

можно построить по

*В отношении R рассматриваем второй элемент каждой пары, ищем в отношении S равный ему первый элемент пары. Если находим совпадение, записываем новую пару, состоящую из первого элемента R и второго элемента S. Одинаковый элемент обеих пар выступает посредником в построении новой пары для нового отношения.*

Как видим из нашего примера, в общем случае:

*R* ∘ *S*  *S* ∘ *R* ,

т.е. операция композиции не обладает *свойством коммутативности*. Рассмотрим еще примеры.

Пусть *R* и *S* – бинарные отношения, заданные на множестве *натуральных чисел N:*

*R*  {(*a*,.*b*) | *a*,*b*  *N*  *b*  *a*2 }и *S*  {(*a*,*b* | *a*,*b*  *N*  *b*  *a* 1} . Тогда:

*S* ∘ *R*  {(*а*,*b*) | *a*,*b*  *N*  *b*  *a*2 1} и *R* ∘ *S*  {(*а*,*b*) | *a*,*b*  *N*  *b*  (*a*  1)2}.

Если отношения на числовых множествах заданы в *аналитическом* виде, можно также воспользоваться правилом:

Для нахождения композиции качестве первого элемента.

*S* ∘ *R*

выражаем из *R* второй элемент через первый и подставляем его в *S* в

Если отношения, *R* и *S* заданы матрицами, матрица их композиции определяется как *булево* (или

*логическое*) произведение соответствующих матриц.

Пусть отношение *R* задано матрицей *M*, а отношение *S* – матрицей *N* . Тогда матрица композиции отношений *S* ∘ *R* определяется как *M*  *N* . Здесь  – операция булевого произведения матриц: в обычной

операции произведения матриц, сложение заменяется логическим *«или»* (дизъюнкцией), а умножение – логическим *«и»* (конъюнкцией).

Пусть на множестве *B = {1, 2, 3}* заданы отношения *R* и *S*: *R*  {(*a*,*b*) | *a*,*b*  *B*  *a*  *b*  *четное число*},

*S*  {(*a*, *b*) | *a*, *b*  *B*  *a*  *b*} . Зададим отношения перечислением пар: *R = {(1,2), (2,1), (2,2), (3,2), (2,3)}* и *S*

*= {(2,1), (3,1) (3,2)}.* Найдем композицию отношений

*S* ∘ *R*  {(1,1),

# (2,1),

(3,1),

(2,2)}. Теперь зададим

наши отношения матрицами, матрица *M* для отношения *R* и матрица *N* для отношения *S*. Имеем:

###  0 1 0

 

*MR*  1 1 1



###  1 

0

0



 0 0 0

###  

*NS*  1 0 0



###  1 

1

0



Найдем композицию наших отношений через булево произведение матриц: *S* ∘ *R*  *M R*  *NS* . Первым

выполняется отношение *R*, поэтому в произведении матрица отношения *R* – результате получим матрицу *P.*

*M R* стоит также *первой*. В

### 1 0



*P*  *MR* *N S*  1 1



###  0

1

0

### 

0 **.**

### 

0



Поясним, каким образом мы получили элементы первой строки. Имеем:

*P*11  (0  0)  (11)  (0 1)  1, *P*12  (0  0)  (1 0)  (0 1)  0 ,

*P*13  (0  0)  (1 0)  (0  0)  0 .

Пусть на множестве людей *A* заданы отношения:

*R*  {(*a*, *b*) | *a*, *b*  *A*  *a*  *сестра b*} и *S*  {(*b*,*c*) | *b*,*c*  *A*  *b*  *мать c*}. Тогда:

*S* ∘ *R*  {(*а*,*c*) | *a*,*c*  *A*  *a*  *тетя c*} , и *S* ∘ *S*  {(*а*,*b*) | *a*,*b*  *A*  *a*  *бабушка b*} .

Для обратного отношения и операции композиции выполняются следующие свойства:

1. (*R*1)1  *R* .

2. (*R* ∘ *R* )1  *R* 1 ∘ *R* 1.

1

2

2

1

3. (*R*1 ∘ *R*2 ) ∘ *R*3  *R*1 ∘ (*R*2 ∘ *R*3 )

Третье свойство означает, что операция композиции *ассоциативна*

Свойства бинарных отношений.

Бинарные отношения обладают *специальными* свойствами. Рассмотрим их.

Бинарное отношение *R* на множестве *A* называют:

1. *Рефлексивным*, если пара (*а, а*) принадлежит *R* для ***всех*** элементов *а* из множества *А*. В

символьном виде это может быть записано: *a*  *A* : *aRa* .

Бинарные отношения равенства и подобия на множестве геометрических фигур рефлексивны: каждый треугольник равен самому себе и подобен самому себе. Рефлексивны все отношения равенства: равенство

чисел, равенство множеств, равенство векторов. Также рефлексивным является отношения: , ,  на

соответствующих множествах. На главной диагонали матрицы рефлексивного отношения стоят единицы. Для рефлексивного отношения выполняется: *I*  *R* .

1. *Антирефлексивным****,*** если оно не выполняется на паре (*a,a*) для любого элемента *a* из *A*. В

символьном виде: *a*  *A* : (*aRa*) . Напомним что  значок логического отрицания, т. е. элемент *а* не

находится в отношении с *а*. Отношения <, > на числовых множествах, отношение *«быть родителем»* на множестве людей являются антирефлексивыми. На главной диагонали матрицы антирефлексивного отношения стоят нули. Для

данного отношения

*R*  *I*  

1. *Симметричным*, если для всех *а* и *b*, принадлежащих *А*, из

(*a*, *b*)  *R* следует, что

(*b*, *a*)  *R* . В символьном виде: *a*, *b*  *A* : *aRb*  *bRa* . Все бинарные отношения в геометрии типа равенства и подобия симметричны, отношения  , || прямых на множестве прямых плоскости также симметричны. Матрица симметричного бинарного отношения на *A* симметрична относительно главной диагонали.Для симметричного отношения *R*  *R* 1 .

1. *Антисимметричным*, если для всех *а* и *b* из *А*, из принадлежности (*а*, *b*) и (*b*,а) отношению *R* следует, что *а* = *b*. В символьном виде: *a*, *b*  *A* : *aRb*  *bRa*  *a*  *b* . Отношения ,  , определенные на числовых множествах, «*а –* делитель *b»* на множестве *N*, отношение нестрогого включения множеств  на системе подмножеств некоторого множества*A*, являются примерами антисимметричных отношений.
2. *Транзитивным*, если для всех *а, b* и *c* из *А* из того, что (*a*, *b*)  *R* и (*b*, *c*)  *R* , следует, что

(*a*, *c*)  *R* . В символьном виде

*a*, *b*, *c*  *A* : *aRb*  *bRc*  *aRc* .

Отношения

,  ,<, >, *«а – делитель b»* на числовых множествах, отношение строгого  и нестрогого

включения множеств  на системе подмножеств некоторого множества *A* обладают свойством транзитивности.

Рассмотрим и охарактеризуем некоторые отношения.

1. Пусть на множестве *A = {1,2,3,4,5}* перечислением пар задано отношение *R = (1,1),(2,2),(3.3),(4,4),(5.5),(1,2),(2,1),(2,4),(4,2),(3,5),(5,3)}*.Отношение является рефлексивным, так как пары

вида (*а, а*) присутствуют для всех

*a*  *A* : *(1,1), (2,2), (3.3), (4,4), (5.5)*.

*R* – симметричное отношение, т. к. из присутствия пары (*а*, *b*) следует присутствие пары (*b*, *а*): *(1,2)* и *(2,1); (2,4)* и *(4,2);(3,5)* и *(5,3).*

*R* – не обладает свойством транзитивности: в отношении присутствуют пары *(4,2), (2,1),* а пары *(4,1)* в отношении нет.

Чтобы сказать, что отношение *не обладает* некоторым свойством, достаточно найти *хотя бы один* пример такого невыполнения.

*R* не является антисимметричным, поскольку

(1,2)  *R*

и (2,1)  *R* , но

2  1.

Отношение обладает указанным свойством, если это свойство выполняется для *всех* возможных случаев.

1. Рассмотрим отношение *S*, заданное на множества натуральных чисел *N*:,

*S*  {(*a*,*b*) | *a*,*b*  *N*  *a*  *делителл b*}.

*S –* рефлексивное отношение, поскольку *a* всегда делит сам себя и

(*a*, *a*)  *S*

*a*  *N*.

Отношение *S* не является симметричным, поскольку, например, *2* является делителем *8*, обратное утверждение не является верным. Отношение *S* является антисимметричным, поскольку из предположений: *a* делит *b* и *b* делит *a*, немедленно вытекает, что *a = b.*

*S* – транзитивное отношение. Пусть *a* делит *b,* это означает, что *b =ma* для некоторого

*m*  *N*.

И пусть *b*

делит *c*, что означает *c = nb* для некоторого следовательно, *a* делит *c.*

*n*  *N*.

Из этих двух посылок вытекает, что *c = (nm)a* и,

1. Рассмотрим отношение *T*, заданное на множества целых чисел

*Z: T*  {(*a*,*b*) | *a*,*b*  *Z*  *a*  *b*  *четное число*} .

Отношение не является рефлексивным, поскольку. не для всякого *a*  *Z* , *a* 2

(например, 52  25 ) является

четным числом.

Из этих же соображений, отношение *T* не является антирефлексивным.

* 1. – симметричное, поскольку операция умножения коммутативна и

*a* *b* 

*b*  *a* .

Отношение *T* не является транзитивным, поскольку, например,

# 3 4

* *четное число* и

# 4  5

* *четное число* , но

# 35 

## нечетное число .

Отношение *T* не является также антисиметричным.

Замыкание отношений.

Если отношение *R* на множестве *A* не обладает тем или иным свойством *P*, то его можно попытаться продолжить до отношения *R*\* , которое будет иметь нужное свойство. Под *«продолжением»* мы понимаем

присоединение некоторых упорядоченных пар к подмножеству *R*  *A*  *A*

так, что новое полученное

множество уже будет обладать требуемым свойством. Ясно, что исходное множество *R* будет подмножеством *R*\* . В том случае, если вновь построенное множество *R*\* будет *минимальным* среди всех

*расширений R* с выделенным свойством, то говорят, что *R*\*

свойства *P*.

является *замыканием* относительно данного

Таким образом, *рефлексивное замыкание R* есть *наименьшее* рефлексивное отношение на *A*, содержащее

*R* как подмножество. Очевидно, *R*  *I*

– есть рефлексивное замыкание на *R*.

*Симметричное замыкание* есть наименьшее симметричное отношение на *A*, содержащее *R* как подмножество, И *R*  *R* 1 есть симметричное замыкание на *R*.

*Транзитивное замыкание* есть наименьшее транзитивное отношение на *A*, содержащее *R* как

подмножество. Если *A* – конечное множество, содержащее *n* элементов, то отношение

*R*  *R* 2  *R*3 ...  *Rn*

Рассмотрим пример.

есть транзитивное замыкание на *R*.

Дано множество *A = {a,b,d,l}* на котором задано отношение *R* = *{(a,b),(b,d),(d,l)}*. Отношение не рефлексивно, не симметрично и не транзитивно.

Рефлексивным замыканием будет отношение:

*R*  {(*a*, *b*),(*b*, *d* ),(*d* , *l*),(*a*, *a*),(*b*, *b*),(*d* , *d* ),(*l*, *l*)}. Симметричным замыканием будет отношение:

*R*  {(*a*, *b*),(*b*, *d* ),(*d* , *l*),(*b*, *a*),(*d* , *b*),(*l*, *d* )}.

Транзитивным замыканием будетотношение:

*R*\*  {(*a*,*b*), (*b*, *d* ), (*d*,*l*),(*a*, *d* ),(*b*,*l*),(*a*,*l*)}.

Отношение эквивалентности.

*Рефлексивное, симметричное и транзитивное* отношение на множестве *A* называется *отношением эквивалентности*.

Отношениями эквивалентности, определенными на соответствующих множествах, являются: отношения равенства чисел и множеств, отношение равномощности множеств, отношение подобия треугольников, || прямых на множестве прямых плоскости, отношение *«иметь одинаковый возраст»*, *«быть на одном курсе»* на множестве всех людей и т. д.

Зададим отношение *R* на множестве *A = {2,3,4,5.6}*:

*R*  {(*a*, *b*) | *a*, *b*  *A* 

*a*  *b* 

## четное число}.

Это отношение *эквивалентности*.

1. Оно рефлексивно, поскольку *a +a = 2a* – всегда четное числодля  *a*  *A*
2. Симметрично, поскольку *a +b = b +a* и, если в отношении присутствует пара *(a,b)*, то, следовательно, присутствует и пара *(b,a)*.
3. И наконец, транзитивно, поскольку из того что (*a +b)* – четное число и (*b +c) –* четное число, следует, что и (*a +c )*также является четным числом. Как мы знаем, сумма двух чисел является четной, если оба числа четны, либо оба не четны.

Пусть *R* – эквивалентность на множестве *A* и

*a*  *A* . Множество всех элементов множества *A*

эквивалентных *a*, называется *классом эквивалентности для a* по отношению *R* и обозначается:

[*a*]  {*b*  *A* | *bRa*} или [*a*]  {*b*  *A* | (*b*, *a*)  *R*} .

Это разбиение множества на классы эквивалентности можно представить себе следующим образом. Пусть

*A* – это множество разноцветных шаров, на котором задано отношение *R*:

*R*  {(*a*,*b*) | *a*,*b*  *A*  *a и b имеют одинаковый цвет*}.

Поскольку *R* – отношение эквивалентности, то каждый класс эквивалентности будет состоять из шаров, имеющих одинаковый цвет, например, первый класс состоит из шаров красного цвета, второй – синего и т. д.

Построим классы эквивалентности для нашего отношения *R*:

*[2] = [4] = [6] = {2, 4, 6}.*

*[1] = [5] = {3,5}.*

Фактически, мы имеем *2* класса эквивалентности: *{четные числа}* и *{нечетные числа}*.

*Как мы видим, любой элемент класса эквивалентности порождает класс эквивалентности, т. е. если*

*b* [*a*], *то* [*a*]  [*b*] .

Отметим, что в силу свойства *рефлексивности*, класс эквивалентности не пуст, т. к.

*a* [*a*]

для любого

элемента

*a*  *A* . С другой стороны, никакой элемент не может принадлежать двум различным классам

эквивалентности.

Множество всех классов эквивалентности называется *фактор множеством множества A по эквивалентности R* и обозначается [ *A*]*R* .

Построим фактор множество для нашего отношения эквивалентности.

[ *A*]*R*  {{2,4.6}, (3,5}} .

Фактор множество, является *разбиением* множества *A* и является также подмножеством булеана:

* 1. ]  2 *A* .

*R*

Основная роль отношения эквивалентности состоит в том, что это отношение определяет некоторый *признак,* позволяющий разделить множество на *непересекающиеся классы*, называемые классами эквивалентности. Получается система классов, обладающая следующими свойствами:

1. она образует разбиение, т. е. классы попарно не пересекаются;
2. любые два элемента из одного класса эквивалентны;
3. любые два элемента из разных классов неэквивалентны; Приведем еще один важный пример отношения эквивалентности.

Рассмотрим множество целых чисел Z и зададим на нем *отношение сравнимости по модулю числа n*, где

*n* натуральное число и *n> 1*. Целое число *а* сравнимо с целым числом *b* по модулю *n*, что обозначается,

*a*  *b* (mod *n*) , если разность *(a - b)* делится на *n* без остатка или можно сказать по-другому, *a* и *b* имеют

*одинаковый остаток* при делении на *n*.

Отношение сравнимости по модулю числа *n* на множестве *Z* является отношением эквивалентности и разбивает множество *Z* на *n* классов эквивалентности. В каждом классе находятся числа, дающие одинаковый остаток от деления на *n*.

Например, пусть *n = 3*, отношению *R* сравнимости по модулю *3* будут принадлежать пары:

*R = {…, (-5, -2), (-4, -1),(-2, -2),(-1, -4), (0, 3), (3, 0), (3, 3), (3, 6)…}*.

Имеем *3* класса эквивалентности, к первому относятся числа, дающие остаток *0* при делении на *3* (кратные

* 1. , ко второму – остаток *1*, и к третьему – остаток *2*. Имеем:

[0] = {…-6, -3, 0, 3, 6,…};

[1] = {…-5, -2, 1, 4, 7,…};

[2] = {…-4, -1, 2, 5, 8,…};

Эти классы эквивалентности по отношению сравнимости по модулю *n* на множестве *Z* называются *классами вычетов по модулю n*. Классы вычетов вместе с операцией сложения образуют алгебраическую систему – *группу*.

Отношение порядка.

*Рефлексивное, антисимметричное, транзитивное отношение называется отношением частичного (нестрогого) порядка.*

Отношение частичного порядка принято обозначать знаком  , это обозначение является *условным*.

Если отношение *R* на множестве *A* является отношением частичного порядка, то

*частично-упорядоченным множеством или ЧУ-множеством*.

( *A*, )

называют

Рассматриваемое нами ранее отношение

*S*  {(*a*,*b*) | *a*,*b*  *N*  *a* 

*делителл b*}, определенное на

множестве натуральных чисел *N*, является отношением частичного порядка, а (*N* , ) – *ЧУ-множеством*.

Пусть *A = {1,2,3}*, его булеан

*P*( *A*)  {,{1},{2},{3},{1,2},{1,3}, (2,3){1,2,3}} . Определим отношение *S*

между элементами булеана (подмножествами *A*) как отношение нестрогого включения  : первый

элемент пары является подмножеством второго элемента. Таким образом, пара

({1},{1,2})  *S* , т. к.

{1}  {1,2} , но ({1.3},{3})  *S* , поскольку . Отношение *S* является отношением частичного

порядка, а (*P*( *A*),

) – *ЧУ*-*множеством*. Здесь знаком  обозначено  .

Два элемента *a* и *b ЧУ*-*множества*

( *A*,

) называются *сравнимыми* по отношению порядка *R*, если

выполняется *a*  *b* или *b*  *a* .

Множество *A* , на котором задано отношение *частичного порядка*, называется *полностью*

*упорядоченным* (*линейно упорядоченным, цепью*), если каждые два элемента *ЧУ-множества* ( *A*, )

сравнимы.

Рассмотрим примеры.

Элементы нет.

{1},{1,2} в *ЧУ–множестве*

(*P*( *A*), ) , где *A = {1,2,3}*, сравнимы между собой, а {3},

{1,

2} –

Пусть на множестве *B = {1,2,3,5,6,10,15,30}* задано отношение частичного порядка *T*:

Числа *3* и *15* сравнимы между собой, а *6* и *15* – нет.

*а*  *делитель b* .

На этом же множестве отношение *R:*

*a*  *b*

(обычное меньше или равно) задает линейный порядок,

полностью упорядочивает множество *B*. Отношения , , полностью упорядочивают множества *N*, *Z*, *Q*, *R*. Отношения частичного порядка на множестве удобно задавать *диаграммами Хассе*. Диаграмма *Хассе*

представляет собой *граф* (иногда *орграф*), в котором петли не указываются. Вершины графа изображают

элементы частично упорядоченного множества *A*, и если

*a*  *c* , для элементов *а* и *с* множества *A*, то

вершина *a* помещается *ниже* вершины *c* и соединяется с ней ребром, если не существует такое *b*

что *a*  *b*  *c* . Если же такое *b* существует и *a*  *b*  *c* , то линия от *a* к *c* не указывается.

 *a*, *c*

Заметим, что диаграммы *Хассе* для *ЧУ*-*множеств*

(*P*( *A*), )

и (*B*,

) , где  есть отношение

*а*  *делитель b* , совпадают. Это означает, что эти частично упорядоченные множества имеют одинаковую

*структуру*. Такие *ЧУ*-*множества* называются *изоморфными*. Ниже, на рисунках *2а*, *2б* приведены диаграммы *Хассе* наших двух*ЧУ*-*множеств*.

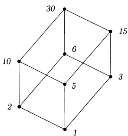
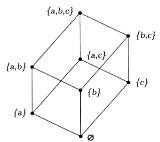


Рис 2а (*P*( *A*), ) Рис 2б (*B*, )

Принцип математической индукции.

Рассмотрим множество натуральных чисел *N*. *Инструментом* для доказательства *истинности* утверждений относительно всех натуральных чисел является принцип *математической индукции*. Сформулируем его.

Пусть *P(n)* есть такое утверждение, что:

* + 1. *P(1)* истинно, и
    2. *для каждого k*, если *P(k)* истинно, то *P(k+1*) истинно. Тогда *P(n*) истинно для любого натурального числа *n*.

В символической записи принцип математической индукции имеет вид:

(*P*(1)  ((*k*)*P*(*k* )  *P*(*k*  1))  (*n*)*P*(*n*) . Рассмотрим примеры.

*Пример 1.*

Докажем методом математической индукции, что

7 *n*  1

делится на *6* при любом натуральном

показателе *n*, т. е. имеет место равенство: 7 *n*  1  6*m* , где *m*  *N* .

1. При *n = 1* имеем: 71  1  7  1  6 . И, следовательно, *P(1)* – истинно.
2. Теперь покажем, что для

*k*  *N* , если *P(k)* истинно, то *P(k+1)* истинно. Предположим, что

7 *k*  1

делится на *6*, т. е.

7 *k*  1  6*m*

для некоторого натурального числа *m.* Запишем утверждение для

*(k+1)* и выполним *эквивалентные преобразования*. Имеем:

7(*k* !) 1  7(7*k* ) 1  7(7*k*  1)  7 1  7(7*k*  1)  6  7  6*m*  6  6(7*m*  1)  6*l* , где

*l*  *N* .

*l*  (7*m*  1) и

Следовательно, утверждение верно для

*делимости целых чисел*.

*n*  *N* . При доказательстве мы использовали *свойства*

Это утверждение мы можем доказать и через *сравнения*. Данное утверждение равносильно следующему:

7*n* 1  0(mod 6)

Имеем:

*n*  *N* .

7  1(mod 6) . Отсюда, возводя сравнение в степень:

7*n*  1*n*  1(mod 6) . И наконец, отнимая от обеих частей сравнения *1* :

### 7*n* 1  1  1  0(mod 6)

*Пример 2.*

*n*  *N* .

Докажем методом математической индукции, что *n*  *N*

### 1  3  5  ...  (2*n*  1)  *n*2 .

1. При *n = 1* имеем *1 = 1*, следовательно, *P(1)* истинно;

выполняется равенство:

1. Предположим что, *P(k)* истинно для некоторого *k*  *N* , т е. верно равенство:

### 1  3  5  ...  (2*k*  1)  *k* 2

*(1)*

и докажем, что из истинности этого утверждения следует истинность для (*k+1)*:

1  3  5  ...  (2*k*  1)  (2*k*  1)  (*k*  1)2 . (2)

Прибавим к обеим частям *(1)* слагаемое *(2k+1)* (отношение равенства при этом не изменяется). Имеем:

### 1  3  5  ...  (2*k*  1)  (2*k*  1)  *k* 2  2*k*  1  (*k*  1)2 .

Как видим, из предположения об истинности *(1)* следует истинность *(2)*, следовательно, равенство

1  3  5  ...  (2*n*  1)  *n*2 верно *n*  *N* .

Сформулируем принцип математической индукции в другой, *эквивалентной* форме:

Пусть *P(n)* – утверждение. Если:

* 1. *P(1)* истинно, и
  2. для произвольного *k* из истинности *P(m)* для *всех m < k* следует истинность *P(k)*, то *P(n*) истинно для всех натуральных чисел *n*.

И, наконец, сформулируем принцип индукции для целых чисел. Пусть *P(n)* – утверждение, обладающее свойствами:

1. *P(j)* истинно, и
2. для произвольного *k* 

*j* , если *P(k)* истинно, то *P(k+1)* истинно.

То *P(n)* истинно для каждого *n*  *j* .

Например, неравенство

*n*! 2*n* , справедливое для

*n*  *N*  *n*  4 доказывается, используя метод

математической индукции для целых чисел. Здесь *j=4*. Часто начинают доказательства с *j=0*, как например, в теореме о мощности булеана.

Докажем теорему о мощности булеана методом математической индукции. *Теорема.* Для конечного множества *A* мощность его булеана: | 2 *A* | 2|*A*|| Пусть задано множество *A*, мощности *k*: *|A |= k*. Индукция по |*A*|.

1. Если |*A* | *= k = 0,* то теорема верна.

*A*  

и 2 *A*  {}. Следовательно, | 2 || {} | 1  20  2|| . Для случая |*A* | = 0

1. Предположим, что теорема верна для множества мощности | *A* | *k* : | 2 *A* | 2|*A*|| .

Покажем, что из предположения справедливости теоремы для множества мощности меньше *k*, например, *k-1*, следует выполнение теоремы для множества мощности *k*.

Рассмотрим множество

*A*  {*a*1 , *a*2 ,..., *ak* } , *|A| = k*. Зафиксируем элемент *ak* .

Разобьем множество *всех* подмножеств *A* (его булеан) на *2* класса: *1* класс – семейство подмножеств, не

содержащие элемент

*ak* , и во *2-ой* класс входят подмножества, содержащие элемент

*ak* . Т.е.

1  {*Х*  *A* | *ak*  *X* } и 2  {*Х*  *A* | *ak*  *X* } .

Очевидно, 2 *A*  *A* U *A* и

1

2

*A* ∩ *A*

  .

По индукционному предположению

1

2

| *A*1

| 2*k*1 , поскольку

*A*1 - это булеан множества

*A* \ {*ak* } ,

состоящего из *(k-1)* элемента и | *A*2

| 2*k*1 , поскольку каждый элемент второго класса

2 получается из

соответствующего элемента первого класса 1

добавлением к нему

*ak* . Следовательно,

| 2 *A* || *A*  | *A*

##### | 2*k*1  2*k*1  2  2*k*1  2*k*

 2| *A*|

1 2

Следовательно, теорема верна *n*  *Z* 0 .

Эту теорему мы могли бы доказать, кодируя каждое подмножество множества *A* двоичным числом, где наличие *1* в некоторой позиции числа означало бы присутствие соответствующего элемента в подмножестве, *0*, соответственно, его отсутствие.

ТЕМА №3*.* ***БУЛЕВЫ ФУНКЦИИ****.*

Пусть задано множество *B = {0, 1}*. Тогда, отображение

*f* : *Bn*  *B*

множества *Bn*

на множество *B*

называется *булевой функцией n переменных* и ее можно записать в виде

*f* (*x*1 ,..., *xn* ) . Булеву функцию

называют также *переключательной функцией*. Переменные булевой функции, как и сама булева функция, могут принимать только *два значения*: *0* или *1*. Таким образом, булева функция определена на множестве всех упорядоченных наборов, состоящих из нулей и единиц, и сама принимает два возможных значения: *0* или *1*.

Каждый набор значений *n* переменных

(*a*1,..., *ai* ,..., *an* )

может быть представлен *n*-*разрядным двоичным*

*числом*. Количество двоичных *n*- разрядных чисел равно

2 *n* . Поэтому любая булева функция *n*

переменных может быть определена на

2 *n* наборах. И, таким образом, мы имеем набор

2 *n* значений

функции *n* переменных. Каждому такому набору значений булевой функции мы можем поставить в

соответствие

2 *n* **-** разрядное двоичное число. Количество

2 *n* - разрядных двоичных чисел равно

2*n*

и,

2

2*n*

2

следовательно, число различных булевых функций *n* переменных равно .

Например, булева функция двух переменных определена на *4* наборах: *(00, 01, 10, 11)*. Количество таких

функций – *16*.

Булева функция может быть задана *таблицей истинности*, где каждому набору значений переменных булевой функции ставится в соответствие значение функции. Каждому набору переменных можно поставить в соответствие номер, равный десятичной записи двоичного числа. Например, (*0, 0, 0*) – *нулевой набор*, (*0, 1, 1*) - третий набор и т.д. Набор *n* переменных, содержащий все единицы (*1, 1,…,1)* называют *единичным* набором. Наборы расположены сверху вниз по возрастанию двоичного числа. Рассмотрим булевы функции одной и двух переменных.

Имеем четыре булевых функций *одной переменной*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x* | *f* 0 | *f*1 | *f* 2 | *f* 3 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |

Функция

*f* 0 (*x*) тождественно равна *0*. Она называется *константой нуль* и обозначается

*f*0 (*x*)  0 .

Функция *f*1 (*x*)

тождественно равна значению переменной *x* и называется *тождественной функцией* и

обозначается

*f*1(*x*)  *x .*

Номер булевой функции соответствует десятичной записи двоичного числа, составленного из значений булевой функции. Двоичное число записывается слева направо, начиная со значения булевой функции на

нулевом наборе. В данном случае это 012  110 .

Функция

*f* 2 (*x*)

принимает значения, противоположенные значениям переменной *x*. Эта функция

называется *инверсией* или *отрицанием* и обозначается также использовать символ логического отрицания, .

*f* 2 (*x*)  *x* . Для обозначения инверсии будем

Функция

*f*3 (*x*) тождественно равна *1*. Она называется *константой единица* и обозначается

*f*3 (*x*)  1.

Константы нуль и единицу принято считать *нуль – местной* булевой функцией. Функций двух переменных – *16*.

Рассмотрим основные булевы функции.

Функция

*f*1(*x*, *y*)

- это известная нам *конъюнкция* или «*логическое и » (логическое умножени*е).

Обозначение: *x*  *y* , *x* & *y* , *x*  *y* , *xy*.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | *y* | *f* 0 | *f*1 | *f* 2 | *f* 3 | *f* 4 | *f* 5 | *f* 6 | *f* 7 | *f*8 | *f* 9 | *f*10 | *f*11 | *f*12 | *f*13 | *f*14 | *f*15 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

Конъюнкция равна единице тогда и только тогда, когда значения *всех* ее переменных *равны единице*. Для конъюнкции справедливы следующие соотношения:

*x*  1  *x* , *x*  0  0 , *x*  *x*  0 .

Функция

*f* 6 (*x*, *y*) называется *сложением по модулю два* (*исключающее или*). Значение функции равно *1*,

если число переменных, значение которых равно *1*, нечетно, и равно нулю, если число таких переменных четно. Обозначение: *x*  *y* .

Функция

*f* 7 (*x*, *y*)

– *дизъюнкция* или «*логическое или» (логическое сложение)*. Обозначение:

*x*  *y* ,

*x*  *y* .

Функция равна нулю тогда и только тогда, когда *все* ее переменные *равны нулю*. Для дизъюнкции справедливы следующие соотношения:

*x*  1  1, *x*  0  *x* , *x*  *x*  1 .

Функция

*f*8 (*x*, *y*)

называется *операцией Пирса* или *стрелкой Пирса*. Обозначение:

*x*  *y* . Ее можно

выразить через операции отрицания и дизъюнкции двух переменных: *x*  *y*  *x*  *y* .

Функция *f*9 (*x*, *y*) – это *эквивалентность* (*логическая равнозначность*). Обозначение: *x* ~ *y* .

Функция

*f*11 (*x*, *y*) – это *импликация* от *y* к *x*. Обозначение:

*y*  *x* .

Функция

Функция

*f*13 (*x*, *y*) – это *импликация* от *x* к *y*. Обозначение: *x*  *y* .

*f*14 (*x*, *y*) называется *операцией Шеффера* или *штрих Шеффера*.

Обозначение:

*x* | *y*  *x*  *y* .

1. | *y* . Ее можно выразить через операции отрицания и конъюнкции двух переменных:

Рассмотренные нами шестнадцать функций двух переменных, назовем их *элементарными*, позволяют строить новые булевы функции следующим образом:

* + путем перенумерации переменных;
  + путем подстановки в функцию новых функций вместо переменных.

Функцию, полученную из функций

*f*1 , *f* 2 ,..., *f k*

путем применения, возможно многократного, этих двух

правил, будем называть *суперпозицией функций f*1 , *f* 2 ,..., *f k* .

Например, функция

*f* (*x*, *y*, *z*)  (*x*  *y*)  *z*

получена суперпозицией элементарных функций: сложения по

модулю два и конъюнкции. Наша функция задана в *аналитическом виде*, составим для нее таблицу истинности.

|  |  |  |  |
| --- | --- | --- | --- |
| *x* | *y* | *z* | *f(x, y, z)* |
| *0* | *0* | *0* | *0* |
| *0* | *0* | *1* | *0* |
| *0* | *1* | *0* | *0* |
| *0* | *1* | *1* | *1* |
| *1* | *0* | *0* | *0* |
| *1* | *0* | *1* | *1* |
| *1* | *1* | *0* | *0* |
| *1* | *1* | *1* | *0* |

Для булевых функций устанавливается приоритет логических операций и они выполняются в таком порядке: отрицание , конъюнкция  , дизъюнкция  , импликация , эквивалентность . Если же нам

необходимо выделить первоочередность некоторой операции, *расставляем скобки*, как в нашем примере: сначала выполняется  , затем  .

Булевы функции могут задаваться различными способами.

*Аналитическим* способом, в виде логического выражения. Например,

*f* (*x*, *y*, *z*)  (*x*  *y*)  *z* .

* + - *Таблицей истинности*. Построим таблицу истинности для нашей функции.

|  |  |  |  |
| --- | --- | --- | --- |
| *x* | *y* | *z* | (*x*  *y*)  *z* |
| *0* | *0* | *0* | *0* |
| *0* | *0* | *1* | *0* |
| *0* | *1* | *0* | *0* |
| *0* | *1* | *1* | *1* |
| *1* | *0* | *0* | *0* |
| *1* | *0* | *1* | *1* |
| *1* | *1* | *0* | *0* |
| *1* | *1* | *1* | *0* |

* + - *Вектором значений* булевой функции. Функция имела бы вид:

### *f* (*x*, *y*, *z*)  (0,0,0,1,0,1,0,0)

* + - *Номером*, соответствующим в десятичной системе счисления двоичному числу, составленному из значений булевой функции на всех наборах. *Двоичное число записывается слева направо, начиная с нулевого набора.* В нашем случае оно имеет вид: 00010100 и ему соответствует номер 20, т. е.

*f* (*x*, *y*, *z*)  *f* 20 (*x*, *y*, *z*) .

* + - Можно перечислить *номера тех наборов, на которых функция равна единице:*

Напомним, что нумерация наборов начинается с нуля.

*f* (*x*, *y*, *z*)  (3,5) .

* + - *В виде ДНФ, КНФ, полинома Жегалкина*. Мы рассмотрим эти представления в дальнейшем.

Булева функция

*f* (*x*1 ,...*xi*1 , *xi* , *xi*1 ,..., *xn* )

*существенно зависит* от переменной

1. , если существует такой

набор значений (*a*1 ,..., *ai*1 , *ai*1 ,..., *an* ) , что выполняется соотношение:

*f* (*a*1 ,...*ai*1 ,0, *ai*1 ,..., *an* )  *f* (*a*1 ,...*ai*1 ,1, *ai*1 ,..., *an* )

В этом случае *xi* называют *существенной* переменной.

В противном случае говорят, что от переменной

*фиктивной* переменной.

*xi* функция зависит несущественно и

*xi* является ее

В функциях

*f* 3 ,

*f*12 из *16* элементарных булевых функций, фиктивной является переменная *y*.

В булевой функции фиктивной.

*f* (*x*, *y*, *z*) , заданной следующей таблицей истинности, переменная *z* является

|  |  |  |  |
| --- | --- | --- | --- |
| *x* | *y* | *z* | *f* (*x*, *y*, *z*) |
| *0* | *0* | *0* | *0* |
| *0* | *0* | *1* | *0* |
| *0* | *1* | *0* | *1* |
| *0* | *1* | *1* | *1* |
| *1* | *0* | *0* | *1* |
| *1* | *0* | *1* | *1* |
| *1* | *1* | *0* | *1* |
| *1* | *1* | *1* | *1* |

Пусть задана булева функция выполняется соотношение:

*f* ( *x*1 , *x*2

,..., *xn* ) .

Функция *f* \*

называется *двойственной* к функции *f* , если

*f* \* ( *x* , *x* ..., *x* )  *f* ( *x*1 , *x*2 ,..., *xn* ).

1 2 *n*

Можно записать и так:

*f* \* (*x* , *x*

1

2

..., *xn* ) 



*f* (*x*1 , *x* 2 ,..., *xn* ).

*Это означает, что функции f и значения*.

Очевидно, что:

\*

*на противоположенных наборах имеют противоположенные*

*f*

*f* \*\*  *f* .

Рассмотрим примеры двойственных функций.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *f* | 1 | 0 | *x*  *y* | *x*  *y* | *x* | *x* |
| *f* \* | 0 | 1 | *x*  *y* | *x*  *y* | *x* | *x* |

Как видим, операции *дизъюнкции* и *конъюнкции* являются *двойственными друг другу.*

Функция называется *самодвойственной*, если:

*f* \*  *f* .

Примеры самодвойственных функций: *x* , *x* ,

*f* ( *x*, *y*, *z*)  *x*  *y*  *z* .

Рассмотрим *основные логические эквивалентности* для булевых функций.*.*

1. *Коммутативность.*

*x*  *y*  *y*  *x* . *x*  *y*  *y*  *x* . *x*  *y*  *y*  *x* . *x* ~ *y*  *y* ~ *x* .

1. *Ассоциативность*.

(*x*  *y*} *z*  *x*  ( *y*  *z*) .

(*x*  *y*)  *z*  *x*  ( *y*  *z*) .

(*x*  *y*)  *z*  *x*  ( *y*  *z*) .

(*x* ~ *y*) ~ *z*  *y* ~ (*x* ~ *z*)

1. *Дистрибутивность*.

(*x*  *y*)  *z*  (*x*  *z*)  ( *y*  *z*) .

(*x*  *y*)  *z*  (*x*  *z*)  ( *y*  *z*) .

(*x*  *y*)  *z*  (*x*  *z*)  ( *y*  *z*) .

1. *Закон двойного отрицания.*

*x*  *x .*

1. *Законы де Моргана.*

*x*  *y*  *x*  *y .*

*x*  *y*  *x*  *y*

1. *Законы идемпотентности.*

*x*  *x*  *x*

*x*  *x*  *x*

1. *Законы поглощения.*

(*x*  *y*)  *x*  *x*

(*x*  *y*)  *x*  *x*

1. *Другие эквивалентности.*

*x*  *x*  1

*x*  *x*  0

*x*  1  1

*x*  0  0 *x*  1  *x x*  0  *x*

*x*  *y*  *x*  *y*.

*x* ~ *y*  *x*  *y*

*x* ~ *y*  (*x*  *y*)  ( *y*  *x*).

Имеют место следующие очевидные утверждения:

*x*1  *x*2    *xn*  1  *i* \_ *xi*  1 *.*

*x*1  *x*2  ...  *xn*  1  *i* \_ *xi*  1

СДНФ, СКНФ булевой функции. Полином Жегалкина.

Мы рассмотрели представление булевой функции через *таблицу истинности.* Решим *обратную* задачу: представим булеву функцию, заданную таблицей истинности, через элементарные функции.

Введем некоторые понятия, связанные с булевыми функциями.

Пусть имеется некоторый *конечный* набор переменных булевой функции:

*x*1 ,

*x*2 , ..., *xn*

Всякую

конъюнкцию переменных функции, взятых с отрицанием или без отрицания, при этом каждая переменная

встречается не более одного раза, называют *элементарной.* Например,

*U*  *x*1 *x*2 *x*3 *x*4

– элементарная

конъюнкция,

*f* (*x*1 , *x*2 , *x*3 , *x*4 )  *x*1 *x*2

*x*3 *x*4

* нет.

Число *r* переменных в элементарной конъюнкции *U* называется ее *рангом*. Наша конъюнкция имеет ранг: *r*

*= 4*. *Константа единица* считается элементарной конъюнкцией ранга *0*.

Дизъюнкцию элементарных конъюнкций называют *дизъюнктивной нормальной формой* булевой функции, сокращенно *ДНФ.*

Пусть булева функция задана через *ДНФ*:

*f* (*x*1,..., *xn* )  *U*1 *U* 2... *U s* , где *Ui* ,

*i*  1, *s*

* элементарные

конъюнкции. Число *s* называют *длиной ДНФ*.

*ДНФ* можно охарактеризовать также числом

*R*  *r*1  *r*2  ...  *rs* , которое называют *суммарным рангом*

этой ДНФ. Здесь *ri*

* *ранг i* – *ой* элементарной конъюнкции.

Например, для булевой функции

*f* (*x*1, *x*2 , *x*3 )  *x*1*x*2 *x*3  *x*2 *x*3 , ее суммарный ранг ее *ДНФ*, *R = 5*.

Пусть имеется элементарная конъюнкция

*U*  *x*1

 *x*2

### ...

*xn* . *Собственной частью* конъюнкции *U*

называют конъюнкцию, полученную из *U*, удалением из *U* некоторых переменных.

Например, если

*U*  *x*1 *x*2 *x*3

*x*4 , то собственными частями конъюнкции *U* являются, например,

конъюнкции:

*U*1  *x*1 *x*2

*x*3 , *U* 2  *x*3

*x*4 , *U* 3 *x*4 и т. д.

Аналогично вводится понятие и для *элементарной дизъюнкции*. Например,

*f* (*x*1 , *x*2 , *x*3 , *x*4 )  *x*1

 *x*2

 *x*3

 *x*4 – элементарная дизъюнкция ранга *4*.

*Конституентой единицы* называют булеву функцию *n* переменных, которая принимает значение, равное единице *на одном единственном наборе* переменных. Обозначается конституента единицы как

*Ki* (*x*1 ,..., *xn* ) , где *i* –номер набора, на котором значение принимает значение *1* на единственном наборе - *101* .

*Ki* равно единице. Например,

*K*5 (*x*1 , *x*2 , *x*3 )

Как мы знаем, конъюнкция равна единице, если значения *всех* ее переменных *равны единице*, поэтому конституенту единицы можно выразить через конъюнкцию *всех своих* переменных.

Для этого необходимо взять набор, на котором значение конституенты равно единице и из переменных набора и их отрицаний составить конъюнкцию. Причем, если значение переменной в наборе равно *1*, берем ее без отрицания, если *0* – с отрицанием. Например, *K*5 (*x*1 , *x*2 , *x*3 )  *x*1 *x*2 *x*3 .

Перейдем к ***СДНФ*.**

Дизъюнкция конституент единицы, равных единице на тех же наборах, что и заданная функция, называется совершенной дизъюнктивной нормальной формой булевой функции, сокращенно СДНФ.

***Л****юбую булеву функцию*

*f* (*x*1,..., *xn* ) *, кроме константы 0, можно представить в виде СДНФ и это*

*представление единственно.*

Как видим, любая булева функция может быть выражена через *«базис»* – *конъюнкцию*, д*изъюнкцию* и

*отрицание*.

Чтобы получить *совершенную дизъюнктивную нормальную форму* (***СДНФ*)**, надо взять все наборы, на которых значение функции равно *1* и записать для каждого из них конъюнкцию переменных и их отрицаний. Если в наборе значение переменной равно *0* – то переменную надо взять с отрицанием, если *1*

– без отрицания. Из получившихся конъюнкций надо построить дизъюнкцию.

**Рассмотрим пример, построим *СДНФ* функции**

*f*1(*x*, *y*, *z*) , заданной таблицей истинности**.**

|  |  |  |  |
| --- | --- | --- | --- |
| *x* | *y* | *z* | *f(x, y, z)* |
| *0* | *0* | *0* | *1* |
| *0* | *0* | *1* | *0* |
| *0* | *1* | *0* | *1* |

|  |  |  |  |
| --- | --- | --- | --- |
| *0* | *1* | *1* | *0* |
| *1* | *0* | *0* | *0* |
| *1* | *0* | *1* | *1* |
| *1* | *1* | *0* | *1* |
| *1* | *1* | *1* | *1* |

*СДНФ* нашей функции имеет вид:

*fСДНФ* (*x*.*y*, *z*)  *xyz*  *xyz*  *xyz*  *xy z*  *xyz* . Перейдем к *СКНФ*.

*Конституентой нуля* называют булеву функцию *n* переменных, которая принимает значение, равное

нулю, на *одном единственном наборе* переменных.

Будем обозначать конституенту нуля как

*Mi* (*x*1 ,..., *xn* ) , где *i* – номер набора, на котором значение *Mi*

равно нулю. Например, значение *M* 5 (*x*1 , *x*2 , *x*3 )

равно *0* на единственном наборе – *101* .

Как мы знаем, дизъюнкция переменных равна нулю, если значения *всех* переменных *равны нулю*, поэтому конституенту нуля можно выразить через дизъюнкцию *всех* своих переменных следующим образом. Для

этого необходимо взять набор, на котором значение конституенты *Mi*

равно нулю и из переменных

набора и их отрицаний составить дизъюнкцию. Причем, если значение переменной в наборе равно *0*, берем ее без отрицания, если – *1*, с отрицанием. Например, *M* 5 (*x*1, *x*2 , *x*3 )  *x*1  *x*2 *x*3 .

Конъюнкция конституент нуля, которые равны нулю на тех же наборах, что и заданная функция, называется

*совершенной конъюнктивной нормальной формой, сокращенно* ***СКНФ***.

***Л****юбую булеву функцию*

*f* (*x*1 ,..., *xn* ) *, кроме константы 1, можно представить в виде СКНФ и это*

*представление единственно.*

*СКНФ* строится по следующему правилу:

Чтобы получить *совершенную конъюнктивную нормальную форму*, надо взять все наборы, на которых значение функции равно *0* и записать для каждого из них дизъюнкцию переменных и их отрицаний. Если в наборе значение переменной равно *0*, то переменную надо взять без отрицания, если *1* – с отрицанием. Из получившихся дизъюнкций надо построить конъюнкцию.

Построим *СКНФ* функции

*f*1(*x*, *y*, *z*) из предыдущего примера.

Имеем:

*fСКНФ* (*x*, *y*, *z*)  (*x*  *y*  *z*)(*x*  *y*  *z*)(*x*  *y*  *z*).

Перейдем к *полиному Жегалкина*.

Справедлива следующая *теорема Жегалкина*.

Любая булева функция может быть представлена в виде полинома, т. е. записана в форме:

*f* (*x*1 ,..., *xn* )  *a*0  *a*1 *x*1 *a* 2 *x*2  ...  *an xn*  *an*1 *x*1 *x*2  ...  *aN x*1 ...*xn*

**где** *a*0 ,

*a*1 ,

### ...,

*aN* – коэффициенты, равные нулю или единице.

Слагаемое **отсутствует, если** коэффициент при нем равен нулю. Как мы видим, полином *Жегалкина* строится с помощью операций: *сложение по модулю два, конъюнкции и константы 1*. Полином ***Жегалкина*** для каждой булевой функции *единственен*.

Максимальное число сомножителей в элементарных конъюнкциях, входящих в полином, называется

*степенью полинома*.

**Полином Жегалкина можно построить различными способами: используя *таблицу истинности* (через**

***СДНФ*), *методом неопределенных коэффициентов*, *методом треугольника*. Построим полином, используя таблицу истинности булевой функции.**

**Возьмем наборы, на которых значение функции равно единице. Из переменных каждого такого набора составим конъюнкцию. Если значение переменной в наборе равно *0*, заменим ее соотношением** *x*  *x* 1***,*** если *1*, оставим без изменения. Между полученными **таким** образом выражениями возьмем

сложение по модулю два. Раскроим скобки, применив дистрибутивный закон: (*x*  *y*)*z*  *xz*  *yz*

Это может быть, например, выражение вида:

*xyz*  (*x* 1) *уz*  *xyz*  *yz* .

###### И приведем подобные члены с учетом соотношения:

*x*  *x* ... *x* {*x*,

*если n*

*нечетно*

*n раз*

0, *если n четно*

Построим *полином Жегалкина* для функции истинности.

*f*1(*x*, *y*, *z*)

из предыдущего примера, заданной таблицей

Имеем:

*f* (*x*, *y*, *z*)  (1 *x*)(1 *y*)(1 *z*)  (1 *x*) *y*(1 *z*)1 *x*(1 *y*)*z*  *xy*(1 *z*)  *xyz* 

 1 *y*  *x*  *xy*  *z*  *yz*  *xz*  *xyz*  *y*  *yz*  *xy*  *xyz*  *xz*  *xyz* 

 *xy*  *xyz*  *xyz*  1 *x*  *z*  *xy*  *xyz*.

Получили полином *третьей* степени, поскольку максимальное число переменных в элементарных конъюнкциях в нашем случае, равно *3 – xyz*.

Можно строить полином, используя *СДНФ* функции.

Классы булевых функций. Полные системы булевых функций.

**Рассмотрим *пять специальных* классов (множеств) булевых функций, называемых также *классами Поста*. Эти классы обладают свойством *функциональной замкнутости*: любая булева функция, полученная с помощью операций суперпозиций из функций данного класса, принадлежит этому же классу.**

**Перечислим эти классы.**

**Класс (множество) булевых функций, *сохраняющих нуль (константу нуль)*. Обозначается как** *T*0 **. Булева**

###### функция называется функцией, сохраняющей нуль, если на нулевом наборе переменных значение функция равна *0*, т. е. *f(00…0) = 0.*

**Класс булевых функций, *сохраняющих единицу (константу единицу).* Обозначается как**

*T*1 . **Булева**

**функция называется функцией, *сохраняющей единицу*, если на единичном наборе переменных значение функция равно *1*, т. е. *f(11…1) = 1.***

###### Класс *линейных* булевых функций. Обозначается через *L*. Булева функция называется *линейной*, если она может быть представлена полиномом степени не *выше первой*, т. е. представлена в виде:

*f* (*x*1 ,..., *xn* )  *a*0  *a*1 *x*1  *a*2 *x*2 ,...,*an xn* ,

где *a*0 ,

*a*1 ,

### *a*2 , ...,

*an* – коэффициенты, равные *0* или *1*.

**Класс *монотонных* булевых функций. Обозначается через *M*.**

**Булева функция называется *монотонной*, если для любых двух *сравнимых наборов a* и *b* выполняется условие:**

(*a*1 ,

### *a*2 , ...,

*an* )  (*b*1 ,

### *b*2 , ...,

*bn* )  *f* (*a*1 ,

### *a*2 , ...,

*an* )  *f* (*b*1 ,

### *b*2 , ...,

*bn* ) .

Рассмотрим множество двоичных наборов функции *n* переменных.

На этом множестве введем *отношение сравнимости двоичных наборов* следующим образом. Говорят, что

двоичный набор

*a*  (*a*1 ,

### *a*2 , ...,

*an* )

*не больше* двоичного набора

*b*  (*b*1 ,

### *b*2 , ...,

*bn* ) , обозначается

*a*  *b* , если для *каждой пары* (*ai* ,

*bi* ),

*i*  1, *n*

справедливо соотношение: *ai*  *bi* .

Отношение сравнимости является отношением *частичного порядка*.

Для функции двух переменных следующие наборы сравнимы: *11 > 10*, *11 > 01*, *10 > 00* и т.д. Наборы *01* и *10*

несравнимы, поскольку

*a*1  *b*1 , но

*a*2  *b*2 . Таким образом, условие *монотонности* для функции двух

переменных мы можем записать в виде:

*f* (1,1) 

*f* (1,0) 

*f* (0,0)  *f* (1,1) 

*f* (0,1) 

*f* (0,0) .

Для функции трех переменных наборы *111* и *110, 101* и *001* являются сравнимыми. Наборы *110* и *001*, *101* и

*011* – несравнимы. Например, для последней пары наборов имеем:

*a*1  *b*1 ,

*a*2  *b*2

и *a*3 

*b*3 , а

отношение *ai*  *bi*

должно выполняться по *всем переменным* наборов.

Множество двоичных наборов функции *n* переменных вместе с заданным на нем отношением сравнимости образует *ЧУ*- *множество*. Диаграмма *Хассе* для него при *n = 3* приведена ниже на *рисунке 1*.

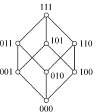


Рис. 1.

**Для того, чтобы убедиться в *немонотонности* заданной булевой функции, достаточно найти *хотя бы***

***одну пару* сравнимых наборов *a* и *b*, таких, что** *a*  *b*

**и *f(a ) > f(b)*.**

**Чтобы сказать, что заданная булева функция монотонна, следует убедиться, что на *всех сравнимых***

***наборах* выполняется условие монотонности** *a*  *b*  *f* (*a*)  *f* (*b*) **.**

**Класс *самодвойственных* булевых функций. Обозначается через *S*.**

**Булева функция называется *самодвойственной*, если на *каждой паре противоположенных наборов*, она принимает *противоположенные значения*, т. е. если выполняется условие:**

*f* (*x*1, ...,

*xn* ) 

*f* (*x*1, ...,

*xn* ) .

Или, что то же самое:

*f* (*x*1, ...,

*xn* ) 

*f* (*x*1, ...,

*xn* ) .

Два набора переменных называются *противоположенными*, если все значения переменных одного набора противоположны значениям переменных другого набора. Чтобы получить противоположенный набор, инвертируем все значения переменных исходного набора.

Например, пары наборов *101* и *010*, *011* и *100* – противоположенные.

Заданная функция не является самодвойственной, если *найдется хотя бы одна* пара противоположенных

наборов, такая что выполняется условие

*f* (*a*) 

*f* (*a*) . Соответственно, функция самодвойственная, если

на *всех парах* противоположенных наборов выполняется условие

*f* (*a*) 

*f* (*a*) .

Зададим булеву функцию

*f* (*x*, *y*, *z*)  *x* ~ ( *y*  *z*) таблицей истинности и проверим ее на принадлежность

пяти классам *Поста*. Для нашей функции сначала выполняется импликация между значением *x* и результатом импликации.

*y*  *z* , затем эквивалентность

|  |  |  |  |
| --- | --- | --- | --- |
| *x* | *y* | *z* | *f(x, y, z)* |
| *0* | *0* | *0* | *0* |
| *0* | *0* | *1* | *0* |
| *0* | *1* | *0* | *1* |

|  |  |  |  |
| --- | --- | --- | --- |
| *0* | *1* | *1* | *0* |
| *1* | *0* | *0* | *1* |
| *1* | *0* | *1* | *1* |
| *1* | *1* | *0* | *0* |
| *1* | *1* | *1* | *1* |

1. Функция **сохраняет нуль, *f(0,0,0) = 0*.**
2. Функция **сохраняет единицу, *f(1,1,1) = 1*.**
3. Построим для нашей функции полином Жегалкина:

*f* (*x*, *y*, *z*)  (*x* 1) *y*(*z* 1)  *x*( *y* 1)(*z* 1)  *x*( *y* 1)*z*  *xyz*  *xyz*  *yz*  *xy*  *y*  *xyz* 

 *xz*  *xy*  *x*  *xyz*  *xz*  *xyz*  *x*  *y*  *yz*.

полином второй степени, функция не является линейной.

Получили

1. Она не является монотонной, поскольку *f(0,1,1)* < *f(0,1,0)* (набор больше, значение функции меньше).
2. Функция не самодвойственная, поскольку *f(0,0,1) = f(1,1,0)* (на противоположенных наборах значения функции равны).

Рассмотрим вопрос *полноты* системы булевых функций.

Система булевых функций

### { *f*1, ...,

*f m*}

называется *полной* (*функционально полной*), если любая булева

функция может быть выражена через функции системы составления сложных функций).

*f*1 , ..., *f m*

с помощью суперпозиций (т. е.

Мы уже сталкивались с полными системами при построении *СДНФ*, *СКНФ*, *полинома Жегалкина* для булевых функций.

*СДНФ*, *СКНФ* позволяют выразить любую булеву функцию через функции отрицания, конъюнкции и

дизъюнкции. Имеем, система { ,,

} – полная система булевых функций.

*Полином Жегалкина* позволяет выразить любую булеву функцию через функции сложение по модулю два,

конъюнкцию и константу *1*. Таким образом, система {,

, 1} – также является полной системой.

Рассмотрим критерий построения полной системы, который дает *теорема Поста*.

*Теорема Поста*.

Для того, чтобы система булевых функций { *f*1, ...,

*f m*} была полной, необходимо и достаточно, чтобы для

каждого из классов *T*0 , *T*1 , *L*, *M*, и *S* нашлась функция *fi* из системы, не принадлежащая этому классу.

Построим еще полные системы булевых функций. Для этого составим *таблицу Поста*, где в строках укажем наиболее важные элементарные функции, а в столбцах – классы. В клетках таблицы будем ставить знаки *« + »* или *« - »* в зависимости от того, принадлежит ли рассматриваемая функция данному функционально замкнутому классу или нет.

В силу *теоремы Поста* для полноты рассматриваемой системы необходимо и достаточно, чтобы для системы в каждом столбце был *хотя бы один минус*.

#### Таблица Поста.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Функции* | *Классы* | | | | |
| *T*0 | *T*1 | *L* | *M* | *S* |
| *Константа 0* | *+* | *-* | *+* | *+* | *-* |
| *Конъюнкция* | *+* | *+* | *-* | *+* | *-* |
| *Сложение по модулю два* | *+* | *-* | *+* | *-* | *-* |
| *Дизъюнкция* | *+* | *+* | *-* | *+* | *-* |
| *Стрелка Пирса* | *-* | *-* | *-* | *-* | *-* |
| *Эквивалентность* | *-* | *+* | *+* | *-* | *-* |
| *Импликация* от *x* к *y* | *-* | *+* | *-* | *-* | *-* |
| *Штрих Шеффера* | *-* | *-* | *-* | *-* | *-* |
| *Константа 1* | *-* | *+* | *+* | *+* | *-* |
| *Инверсия x* | *-* | *-* | *+* | *-* | *+* |

Найдем из таблицы полные системы булевых функций.

1. Одна функция, *стрелка Пирса*, *f* (*x*, *y*)  *x*  *y*

является полной системой.

1. *Штрих Шеффера*,

*f* (*x*, *y*)  *x* | *y* также пример полной системы.

1. Импликация от *x* к *y*,

*f* (*x*, *y*)  *x*  *y* и инверсия *x*,

*f* (*x*)  *x* – полная система.

1. Конъюнкция,
2. Дизъюнкция

*f* (*x*, *y*)  *x*  *y*

*f* (*x*, *y*)  *x*  *y*

и инверсия *x*. и инверсия *x.*

Как видим из таблицы, система функций

### { ,  , }

является *избыточной*, нам бы хватило системы

### { ,

} или

### { ,

}. Полная система булевых функций называется *базисом*, если при удалении любой

функции из этой системы, она перестает быть полной. Очевидно, полные системы

{,

, 1}, { ,

},

### { , }

являются базисами.

Полная система

{ ,  , } получила наибольшее практическое применение, и логическая часть *ПК*

реализована, в основном, через функции этой системы. Эта система получила название *основной функционально полной системы* булевых функций, сокращенно *ОФПС*.

Минимизация булевых функций.

В общем случае существует несколько способов записи одной и той же булевой функции. Сложность записи булевой функции можно оценивать *числом элементарных операций*, используемых в такой записи и образующих функционально полную систему булевых функций.

Задачу поиска наиболее простой записи булевой функции называют задачей минимизации. Такая задача возникает в ряде приложений. В частности, при проектировании устройств автоматики или вычислительных устройств их работа может быть описана некоторой булевой функцией или системой таких функций. Каждой элементарной функции в устройствах соответствует некоторый физический элемент (ячейка), реализующий эту функцию. И, следовательно, представлению булевой функции в минимальной форме соответствует более простое устройство автоматики или вычислительной техники по сравнению с устройством, реализующим не минимальную булеву функцию.

Наиболее распространённой формой представления булевой функции является *дизъюнктивная нормальная форма.* Поэтому задачу упрощения булевых функций обычно формулируют в *классе ДНФ*.

*ДНФ* булевой функции

*f* (*x*1,..., *xn* )

называют *минимальной*, если ей соответствует *наименьший*

*суммарный ранг R* по сравнению с другими *ДНФ* этой же функции. Минимальную *ДНФ* булевой функции будем обозначать как *МДНФ*.

Поскольку методы минимизации булевых функций, разработанные для *ДНФ*, достаточно легко переносятся на конъюнктивные нормальные формы (*КНФ*), ограничимся рассмотрением методов минимизации в классе *ДНФ*.

В основном эти методы в явной или в неявной форме основаны на выполнении *трех* операций. Рассмотрим эти операции.

Две элементарных конъюнкции одинакового ранга *r* называются *соседними,* если они являются функциями одних и тех же переменных и отличаются только знаком отрицания одной из переменных.

Например, следующие элементарные конъюнкции являются соседними:

* 1.  *x x x x* и *U*  *x x x x* .

1 1 2 3 4 2 1 2 3 4

Первая операция, называемая *операцией склеивания*, для элементарных конъюнкций осуществляется по следующему правилу: *дизъюнкцию двух соседних конъюнкций некоторого ранга r можно заменить одной элементарной конъюнкций ранга r-1, являющейся общей частью исходных конъюнкций.*

*xA*  *xA*  (*x*  *x*)  *A*  1 *A*  *A* ,

где *A* – некоторая элементарная конъюнкция. В этом случае говорят, что конъюнкции *xA* и *xA*

*склеиваются по переменной x.* Например:

*x*1 *x*2

*x*3 *x*4  *x x*2

*x*3 *x*4

1

 *x*2

*x*3 *x*4 .

Здесь

*A*  *x*2 *x*3 *x*4

и две заданные конъюнкции были склеены по переменной

*x*1 *,* в одной из которых эта

переменная без отрицания, а в другой – с отрицанием.

Вторая из операций называется *операцией поглощения,* и она осуществляется по правилу:

*Дизъюнкцию двух элементарных конъюнкций разных рангов, из которых одна является собственной частью другой, можно заменить конъюнкцией, имеющим меньший ранг.*

*A*  *A*  *B*  *A* 1 *A*  *B*  *A*  (1 *B*)  *A* 1  *A* ,

где *А* и *В* – некоторые элементарные конъюнкции. Например,

*x*1 *x*2 *x*3 *x*4  *x*2 *x*4  *x*2 *x*4 (*x*1 *x*3 1)  *x*2 *x*4 .

Третья операция – *операция неполного склеивания*:

*xA*  *xA*  *xA*  *xA*  *A* .

Введем некоторые понятия.

Булева функция (*x*1 ,..., *xn* ) называется *импликантой* функции

*f* (*x*1 ,..., *xn* ) , если функция  равна нулю

на *всех тех же наборах*, на которых равна нулю и функция *f*. Если  есть импликанта функции *f*, то этот

факт записывают следующим образом:   *f* .

Рассмотрим импликанты функции

*f*6 (*x*, *y*) .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *x* | *y* | *f*6 | *f*4 | *f*2 | *f*0 | *f*3 |
| *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *0* | *1* | *1* | *1* | *0* | *0* | *0* |
| *1* | *0* | *1* | *0* | *1* | *0* | *1* |
| *1* | *1* | *0* | *0* | *0* | *0* | *1* |

Как видим из таблицы, функции

*f*4 ,

*f*2 ,

*f*0 , являются импликантами функции

*f*6 , функция

*f*3 – нет.

Единицы функции *f* покрываются как значнниями *0*, так и значениями *1* функции  .

Термин *«импликанта»* возник ввиду того, что импликация наборам тождественно равна единице.

  *f* значений функций  и *f* по всем

Конституента единицы в *СДНФ* функции *f* всегда является её импликантой. Для функции

*f*2 являются конституентами елиницы из ее *СДНФ*.

*f*6 функции

*f*4 и

Из определения импликанты  следует, что если на некотором наборе значение  равно единице, то и значение функции *f* на этом же наборе также равно единице и в этом случае говорят, что импликанта  *покрывает единицу функции f.*

Чем меньше переменных содержит импликанта функции *f*, представленная в виде элементарной конъюкции, по сравнению с конституентой единицы из ее *СДНФ*, тем больше единиц функции она покрывает.

Элементарная конъюнкция *U* *x*1*x*2 ...*xk* называется *простой импликантой* булевой функции *f* (*x*1 ,..., *xn* ) ,

если *U* является импликантой функции *f* и никакая собственная часть *U* не является импликантой *f* .

Рассмотрим пример. Зададим функция *f(x, y, z)* таблицей истинност. (таблица ниже). *СДНФ* функции имеет вид:

*fСДНФ* (*x*.*y*, *z*)  *xyz*  *xyz*  *xyz*  *xyz*  *xyz* .

Укажем импликанты и простые импликанты нашей функции.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | *y* | *z* | *f(x, y, z)* | *xyz* | *xz* | *xyz* | *xy* | *y* |
| *0* | *0* | *0* | *1* | *0* | *0* | *0* | *0* | *1* |
| *0* | *0* | *1* | *1* | *0* | *1* | *0* | *0* | *1* |
| *0* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *0* | *1* | *1* | *1* | *1* | *1* | *0* | *0* | *0* |
| *1* | *0* | *0* | *1* | *0* | *0* | *1* | *1* | *1* |
| *1* | *0* | *1* | *1* | *0* | *0* | *0* | *1* | *1* |
| *1* | *1* | *0* | *0* | *0* | *0* | *0* | *0* | *0* |
| *1* | *1* | *1* | *0* | *0* | *0* | *0* | *0* | *0* |

Как видим из таблицы, конституента единицы

*K*3 (*x*, *y*, *z*)  *xyz*

является импликантой нашей функции, ее

собственная часть

*U*1  *xz*

– простой импликантой, поскольку ни

*U* 2  *x* , ни

*U* 3  *z*

не являются

импликантами. Аналогично имеем,

*K*4 (*x*, *y*, *z*)  *xyz* , *U* 4  *x y*

– импликанты нашей функции, *U* 5  *y* –

простая импликанта; *U*5

серым цветом.

покрывает *4* единицы нашей функции. Простые импликанты в таблице выделены

Справедливы следующие теоремы.

*Теорема 1.*

*Всякая булева функция импликант.*

*f* (*x*1,..., *xn* )

*может быть представлена как дизъюнкция всех своих простых*

Дизъюнкция всех простых импликант функции *f* называется *сокращённой ДНФ* этой функции. *Сокращённая ДНФ* булевой функции *единственна*.

*Теорема 2*. *(теорема Квайна)*.

*Если в совершенной дизъюнктивной нормальной форме булевой функции выполнить* ***все*** *операции неполного склеивания, а затем* ***все*** *операции поглощения, то в результате будет получена сокращенная дизъюнкти*вная *нормальная форма этой функции, или дизъюнкция всех ее простых импликант*.

Как следует из теоремы, чтобы получить *все* простые импликанты, мы проводим операции неполного склеивания. Это связано с тем, что одна и та же элементарная конъюнкция дизъюнктивной формы может склеиваться с несколькими конъюнкциями. При каждом таком склеивании образуются *различные импликанты* и, поэтому мы оставляем в выражении каждую элементарную конъюнкцию для использования ее при других склеиваниях.

От сокращенной *ДНФ* переходим к минимальной дизъюнктивной нормальной форме – *МДНФ*. Рассмотрим один из методов минимизации булевых функций

Метод Квайна.

Этот метод удобен для нахождения *МДНФ* функции от любого числа переменных.

*Алгоритм метода Квайна****.***

1. Записываем *СДНФ* булевой функции

*f* (*x*1,..., *xn* ) .

1. Проводим все возможные операции неполного склеивания конституент единицы *СДНФ* булевой функции, в результате операций получим элементарные конъюнкции ранга *(n – 1)*. Проводим, если это возможно, операции поглощения между конъюнкциями различных рангов. Затем вновь проводим операции неполного склеивания и поглощения до тех пор, пока это возможно. Элементарная конъюнкция, которая не может больше участвовать в склеивании, является *простой импликантой* булевой функции.

Операцию поглощения можно выполнить после всех операций неполного склеивания.

1. Дизъюнкция простых импликант приводит к *сокращённой ДНФ* булевой функции:

*f* *x*1 , *x*2 ,..., *xn*    *k* ,

*k*

где *k*

− простая импликанта.

1. Строим *таблицу покрытия* (*импликантную матрицу*). *Таблица покрытия* − двумерная таблица, каждой строке которой взаимно однозначно соответствует простая импликанта, а каждому столбцу – конституента единицы из заданной СДНФ. На пересечении *i-ой* строки и *j-го* столбца находится *1* (можно ставить

*«крестик»*), если простая импликанта является *собственной частью* соответствующей конституенты единицы, т. е. каждая переменная, входящая в простую импликанту, входит и в конституенту.

1. Выбираем *существенные импликанты*, т. е. такое *минимальное количество* простых импликант, чтобы они *совместно* своими единицами покрывали все колонки импликантной матрицы.
2. *Дизъюнкция существенных импликант приводит к МДНФ.*

Рассмотрим пример.

Найдем *МДНФ* булевой функции, заданной *СДНФ*:

*fСДНФ* *x*1 , *x*2 , *x*3   *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3 .

1. Проведем операции *неполного склеивания*, их лучше выполнять последовательно, слева на право.

1) *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*2 *x*3 .

2) *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*3 .

3) *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 .

4) *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*2 *x*3 .

У нас *все* конституенты единицы участвовали в склеивании, поэтому операций поглощения нет.

Элементарные конъюнкции больше не склеиваются между собой, следовательно, мы получили *простые импликанты.*

1. Из простых импликант строим сокращенную *ДНФ* функции:

*f* *x*1 , *x*2 , *x*3   *x*2 *x*3  *x*1 *x*3  *x*1 *x*2  *x*2 *x*3 .

1. Строим таблицу покрытия.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Простые импликанты* | *Конституенты единицы* | | | | |
| *x*1 *x*2 *x*3 | *x*1 *x*2 *x* | *x*1 *x*2 *x*3 | *x*1 *x*2 *x*3 | *x*1 *x*2 *x* |
| *x*2 *x*3 | *1* | *1* | *0* | *0* | *0* |
| *x*1 *x*3 | *0* | *1* | *1* | *0* | *0* |
| *x*1 *x*2 | *0* | *0* | *1* | *0* | *1* |
| *x*2 *x*3 | *0* | *0* | *0* | *1* | *1* |

Выбираем *существенные* импликанты.

Первая импликанта входит в *МДНФ* обязательно, она единственная, кто покрывает единицу функции на наборе *000* (единственная единица в первой колонке), затем можно выбрать *3-ю* и *4-ю* простые импликанты. Получили *МДНФ*:

*fМДНФ* *x*1 , *x*2 , *x*3   *x*2 *x*3  *x*1 *x*2  *x*2 *x*3

1 .

У нашей функции *две МДНФ*: к первой импликанте можно выбрать *2-ю* и *4-ю*. Получим:

*fМДНФ* *x*1 , *x*2 , *x*3   *x*1 *x*3  *x*2 *x*3  *x*2 *x*3

2 .

Как видим, у булевой функции может быть *несколько МДНФ*. Ранг обеих *МДНФ*, *R = 6*.

В тоже время, построив *СКНФ* заданной функции и проведя минимизацию методом *Квайна* в базисе *КНФ*, мы получим единственную *МКНФ* (минимальную конъюнктивную нормальную форму) меньшего ранга, *R = 5*.

*fСКНФ* *x*1 , *x*2 , *x*3   (*x*1  *x*2  *x*3 )(*x*1  *x*2  *x*3 )(*x*1  *x*2  *x*3 )

*fМКНФ* *x*1 , *x*2 , *x*3   (*x*2  *x*3 )(*x*1  *x*2  *x*3 ) . Рассмотрим еще пример.

*fСДНФ* *x*, *y*, *z*  *xyz*  *xyz*  *xyz*  *xyz*  *xyz* . Проведем операции неполного склеивания.

1. *xyz*  *xyz*  *xy* .

1. *xyz*  *xyz*  *yz* .
2. *xyz*  *xyz*  *xz* .

1. *xyz*  *x yz*  *yz* .
2. *xyz*  *xyz*  *xy* .

Все конституенты единицы участвовали в склеивании, у нас нет операции поглощения. Импликанта *xz* больше не участвует в склеивании, это простая импликанта. Оставшиеся импликанты могут склеиваться между собой, продолжим склеивание.

1. *xy*  *xy*  *y* .

1. *yz*  *yz*  *y* .

Операции поглощения у нас опять нет, получили *2* простые импликанты. Сокращенная ДНФ функции имеет вид:

*f* *x*, *y*, *z*  *xz*  *y* .

Построим таблицу покрытия.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Простые импликанты* | *Конституенты единицы* | | | | |
| *xyz* | *xyz* | *xyz* | *xyz* | *xyz* |
| *xz* | 0 | 1 | 1 | 0 | 0 |
| *y* | 1 | 1 | 0 | 1 | 1 |

Обратите внимание, что простая импликанта *y* состоит только из отрицания одной переменной и покрывает *4* единицы нашей функции.

Как видим, обе простые импликанты входят *МДНФ*:

*fМДНФ* *x*, *y*, *z*  *xz*  *y*

Построение полинома Жегалкина методом треугольника.

Рассмотрим пример из лекции. Мы строили полином *Жегалкина* для функции, заданной таблицей истинности *методом зквивалентных преобразований*, наш полином имел вид: .

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

Построим его теперь *методом треугольника*.

* + Выпишем все значения нашей булевой функции в ряд, записывая *сверху вниз* и

*слева-направо*. Имеем :

1 0 1 0 0 1 1 1

* + Теперь возьмем между *каждыми 2* значениями булевой функции операцию *сложение по модулю 2*. Мы получим 7 значений, каждое из них будем записывать под серединой *2* предыдущих значений. Получаем *2* строки:

1 0 1 0 0 1 1 1

1 1 1 0 1 0 0

* + Далее складывем *по модулю 2* каждые *2* значения второй строки, получаем третью строку, состоящую из *6* значений. Продолжая процесс получим *8* строк, в

последней из которой будет стоять только одно значение. Для нашей функции *треугольник* будет иметь вид:

1 1

1 0 1 0 0 1

1 1 1 0 1 0

0 0 1 1 1

0 1 0 0

1 1 0 1

0 1 1

1 0

1

0

0

1

* Рассмотрим *крайнюю левую диагональ* нашего треугольника ( она выделена *бирюзовым* цветом) и будем сравнивать ее наборами переменных в таблице истинности. Имеем *8* наборов переменных и *8* значений на *бирюзовой* диагонали, таким образом, каждому набору соответствует свое значение. ( рассматриваем *сверху вниз*).
* Если для для некоторого набора соответствующее ему значение на диагонали равно *единице*, значит из переменных этого набора составляем коньюнкцию, причем, если значение переменной в наборе равно *1*, эта переменная есть в элементарной коньюнкции, если значение переменной в наборе равно *0*, переменная в коньюнкции отсутствует. Составляем коньюнкции для всех таких наборов и берем между ними *сложение по модудю 2*.
* Ддя *нулевого* набора (*000*) значение *1* на диагонали означает присутствие в полиноме константы *1*, значение *0* – отсутствие константы *1* (*0* просто не пишем).

Теперь построим для нашей функции *полином Жегалкина*.

Набору *000* соответствует *1*, значит присутствует константа *1*, следующий такой набор *001*, ему будет соответствовать коньюнкция – *z* (только значение *z* в наборн равно *1*), следующий набор, которому соответствует *1– 100,* он дает нам *x*, набор *110* дает нам *xy* и, наконец, набор *111* дает нам коньюнкцию *xyz*.

Теперь, чтобы получить полином, возьмем между всеми полученными таким образом коньюнкциями,

*сложение по модулю 2*.

В итоге получим:

.

Метод треугольника, иногда называемый методом треугольника *Паскаля* из-за схожести принципов построения, удобно реализовывать *программно.*

*Полином Жегалкина* можно строить и *методом неопределенных коэффициентов,* по существу, в методе треугольника и заложен метод неопределенных коэффициентов.

Метод диаграмм Карно-Вейча.

Диаграмма *Карно* выглядит как прямоугольная таблица с количеством клеток, зависящим от числа переменных булевой функции, для которой она строится.

Общее правило построения диаграмм *Карно* состоит в следующем. Число *n* переменных булевой функции делится на два. Если *n* – нечётное, то строкам диаграммы ставятся в соответствие наборы значений *m = (n - 1)/2 первых* булевых переменных, а столбцам – наборы значений оставшихся (*n – m*) булевых переменных. Если же *n* – чётное, то строкам и столбцам диаграммы соответствуют наборы *m = n/2* переменных.

Применяются также диаграммы, где число переменных по строкам и столбцам меняются местами, и диаграмма при нечетном числе *n* вытянута в длину.

При построении прямоугольной (или квадратной) таблицы с *m* строками и (*n – m*) столбцами, расположение наборов переменных булевой функции по строкам и столбцам должны удовлетворять следующему условию – соседние наборы должны отличаться не более чем в одной позиции (разряде). Например, для двух и трёх булевых переменных соответственно имеем:

*00 01 11 10*

*000 001 011 010 110 111 101 100.*

Порядок наборов представляет собой генерацию кодов *Грея,* в которой *2* соседних кода отличаются ровно на *1* позицию.

Таким образом, каждой клетке таблицы однозначно соответствует набор из *n* переменных булевой функции, первые *m* значений переменных набора соответствуют строке расположения клетки таблицы, последующие (*n - m*) – столбцу.

*Прямоугольная таблица заполняется значениями булевой функции по всем наборам.*

Поскольку две соседние клетки отличаются значением только одной переменной (например, *x* и *x* ), то в случае расположения единиц в двух соседних клетках и минимизации в базисе *ДНФ*, мы склеиваем конституенты единицы по этой переменной (в нашем случае по *x*). В случае минимизации в базисе *КНФ* при расположении нулей в двух соседних клетках, склеиваем конституенты нуля по соответствующей переменной.

Будем минимизировать в базисе *ДНФ*.

В общем случае на диаграмме нужно выделить одну или группу размером

2*k* ,

*k*  0, *n*

рядом стоящих

единиц, таким образом, чтобы перекрыть одним выделением как можно больше единиц.

При этом необходимо учитывать, что верхние и нижние ряды клеток рассматриваются как соседние. Аналогично следует рассматривать как соседние, левые и правые столбцы клеток. Выделенные группы должны образовывать или одну клетку, или линию, или прямоугольник, или квадрат.

Такими группами должны быть выделены *все единицы* на диаграмме.

По выделенным группам составляем конъюнкции. Переменные, которые изменяются в выделенной группе, в конъюнкцию не входят, мы склеиваем конъюнкции по этим переменным. Составляем конъюнкцию из переменных, значение которых *не изменяется* в выделенной группе наборов. Из полученных конъюнкций составляем дизъюнкцию и получаем *ДНФ* булевой функции.

Если выделенные группы содержат только максимально возможное количество единиц и количество групп минимально возможно, то полученная *ДНФ* – минимальная *ДНФ*, *МДНФ*.

Рассмотрим пример, проведем минимизацию функции, заданной *СДНФ*:

*fСДНФ* *x*, *y*, *z*  *xyz*  *xyz*  *xyz*  *xyz*  *xyz* .

Строим диаграмму Карно и записываем в клетки таблицы значения функции на всех наборах.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x\yz* | *00* | *01* | *11* | *10* |
| *0* | *1* | *1* | *1* | *0* |
| *1* | *1* | *1* | *0* | *0* |

Первая группа, выделенная светло-серым цветом, состоит из четырех клеток, переменные *x* и *z* в этой группе меняют свои значения, поэтому записываем только значение переменной, которое не меняется в

выделенной группе – *y* .

Фактически, в этой группе мы склеиваем конституенты единицы, например, по строкам по переменной *z*. Затем полученные две импликанты склеиваем по *x*. В результате склеивания получим простую импликанту

– *y* .

Во второй группе, состоящей из двух клеток, выделенных темно-серым цветом, не меняется значение

переменной *x*, ее значение в обоих наборах – *x* и значение переменной *z*. Получаем конъюнкцию – Получаем такую же *МДНФ* как и при минимизации методом *Квайна*:

*fМДНФ* *x*, *y*, *z*  *xz*  *y* .

Минимизируем нашу функцию в базисе *КНФ*.

*xz* .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x\yz* | *00* | *01* | *11* | *10* |
| *0* | *1* | *1* | *1* | *0* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *1* | *1* | *1* | *0* | *0* |

Поскольку это базис *КНФ*, то если переменная в наборе равна *0*, мы берем ее без отрицания, если *1* – с отрицанием, между переменными составляем дизъюнкцию. В результате получим:

.

Из *МДНФ* и *МКНФ* выбираем минимальную функцию с наименьшим рангом, в данном случае это *МДНФ*. Рассмотрим еще пример.

Пусть булева функция задана *МКНФ*:

*fСКНФ* *x*1 , *x*2 , *x*3   (*x*1  *x*2  *x*3 )(*x*1  *x*2  *x*3 )(*x*1  *x*2  *x*3 ) . Построим диаграмму *Карно* для нашей функции.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x*1 *\ x*2 *x*3 | *00* | *01* | *11* | *10* |
| *0* | *1* | *0* | *1* | *0* |
| *1* | *1* | *0* | *1* | *1* |

У нас склеивается *первая* и *третья дизъюнкции* по переменной , поскольку это базис *КНФ*, то если переменная в наборе равна *0*, мы берем ее без отрицания, если *1* – с отрицанием, между переменными

составляем дизъюнкци. В результате имеем: . Bторая дизъюнкция не склеивается ни с чем и ее мы записываем полностью – .

Получаем *МКНФ*:

*fМКНФ* *x*1 , *x*2 , *x*3   (*x*2  *x*3 )(*x*1  *x*2  *x*3 ) . Рассмотрим еше пример.

Пусть булева функция задана *СДНФ*:

*fСДНФ* *x*1 , *x*2 , *x*3   *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3 . Можно выделить группы так

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x*1 *\ x*2 *x*3 | *00* | *01* | *11* | *10* |
| *0* | *1* | *0* | *1* | *0* |
| *1* | *1* | *0* | *1* | *1* |

В этом случае *МДНФ* имеет вид:

*fМДНФ* *x*1 , *x*2 , *x*3   *x*2 *x*3  *x*1 *x*2  *x*2 *x*3 . Или так:

1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *x*1 *\ x*2 *x*3 | *00* | |  | *01* | *11* | *10* |  |
| *0* | *1* | |  | *0* | *1* | *0* |  |
| *1* | *1* |  |  | *0* | *1 1* |  |  |

*fМДНФ* *x*1 , *x*2 , *x*3   *x*1 *x*3  *x*2 *x*3  *x*2 *x*3 .

2

Рассмотрим минимизацию функции четырех переменных.

*fСДНФ* *x*, *y*, *z*,*u*  *xyzu*  *xyzu*  *xyzu*  *xyzu*  *xyzu*  *xyzu*  *xy zu*  *xyzu*  *xyzu*  *xyzu*

Составим диаграмму *Карно* и выделим группы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *xy\zu* | *00* | *01* | *11* | *10* |
| *00* | *1* | *1* |  | *1* |
| *01* |  | *1* | *1* |  |
| *11* |  | *1* | *1* |  |
| *10* | *1* | *1* |  | *1* |

Рассмотрим группу фиолетового цвета, указанные ячейки образуют квадрат, если учитывать, что верхние и нижние ряды клеток рассматриваются как соседние и аналогично следует рассматривать соседними левые и правые столбцы клеток.

В квадрате не изменяется значение переменной *y*, соответственно получаем, – *y* и значение переменной

*u* – *u* . В результате получаем простую импликанту – *yu* . Квадрат, выделенный красным цветом, нам дает

– *yu* . И, наконец, зеленый прямоугольник нам дает – *zu* . В итоге имеем единственную *МДНФ*:

*fМДНФ* *x*, *y*, *z*, *u*  *yu*  *yu*  *zu*

***Минимизация булевых функций(Б.Ф)***

В общем случае существует несколько способов записи одной и той же булевой функции. Например, пусть булева функция задана следующей таблицей истинности:



|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | f( , ) |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

*Совершенная ДНФ* заданной функции имеет вид:

f( ) =



,



С другой стороны, совершенная КНФ имеет вид:

f( ) =



,

Сложность записи булевой функции можно оценивать *числом элементарных операций*, используемых в такой записи и образующих функционально полную систему булевых функций. Так для записи вышеуказанной булевой функции в СДНФ было использовано 29 элементарных булевых операций, с другой стороны, для записи этой же функции в виде совершённой КНФ использовано только две операции дизъюнкции.

Задачу поиска наиболее простой записи булевой функции называют *задачей минимизации*. Такая задача возникает в ряде приложений. В частности, при проектировании устройств автоматики или вычислительных устройств их работа может быть описана некоторой булевой функцией или системой таких функций. Каждой элементарной функции в устройствах соответствует некоторый физический элемент(ячейка) , реализующий эту функцию. Следовательно, представлению булевой функции в минимальной форме соответствует более простое устройство автоматики или вычислительной техники, чем устройство, реализующее не минимальную булеву функцию. Наиболее распространённой формой представления булевой функции является дизъюнктивная нормальная форма. Поэтому задачу упрощения булевых функций обычно формулируют в классе ДНФ. Всякую конъюнкцию переменных функции, взятых с отрицанием или без отрицания, при этом каждая переменная встречается не более одного раза, называют *элементарной.* Дизъюнкцию элементарных конъюнкций называют *дизъюнктивной нормальной формой*

*(ДНФ).* Пусть имеется ДНФ булевой функции f(: f( 

где  (i =1,2,…s) - элементарная конъюнкция. Число s называют длиной ДНФ.

ДНФ булевой функции f( называют кратчайшей , если она содержит наименьшее число элементарных конъюнкций по сравнению с другими ДНФ этой же функции.

Рассмотрим некоторую элементарную конъюнкцию: U = 

Число *r* переменных в конъюнкции U называют ее *рангом.* ДНФ, представленную соотношением (1) можно охарактеризовать также числом

R =  ,

которое называют суммарным рангом этой ДНФ.

ДНФ булевой функции f() называют *минимальной*, если ей соответствует наименьший суммарный ранг R по сравнению с другими ДНФ этой же функции.

Так как методы минимизации булевых функций, разработанные для дизъюнктивных нормальных форм, достаточно легко переносимы на конъюнктивные нормальные формы, то обычно ограничиваются рассмотрением методов минимизации в классе ДНФ.

В основном эти методы в явной или в неявной форме основаны на выполнении трех следующих операций. Первая из них называется операцией склеивания

*x A * *A* = (x ) *A* = *A*

так как *x * =*1* и *1 А* = *А* , где *А* есть некоторая элементарная конъюнкция.

В этом случае говорят, что конъюнкции *x A * *A* склеиваются по переменной *x.*

Например,



Здесь *А* =  и две заданные конъюнкции были склеены по переменной *x1,* в одной из которых эта переменная без отрицания, а в другой – с отрицанием.

Вторая из операций называется *операцией поглощения*

*А АВ = А,*

где *А* и *В* – некоторые элементарные конъюнкции.

Так как *1 * *В* = *1* и *1 * *А* = *А*, то нетрудно убедится также в справедливости соотношения:

*А АВ =А(1 * *В)=А.*

Например,

=

Здесь *А* = , *В* = .

Третья операция – *операция обобщенного склеивания* :

Ax



В самом деле, имеем Ax Ax(1 (1



Булева функция  () называется *импликантой* функции

*f*( ) , если функция  равна нулю на тех же наборах, на которых равна нулю функция *f*. Если есть импликанта функции *f* то этот факт записывают так:

.

В том случае, когда импликанта  равна единице на наборе, на котором единице равна и функция *f,* то говорят, что импликанта  *покрывает* единицу функции *f*.

Пусть имеется элементарная конъюнкция *U* = . *Собственной частью* конъюнкции *U* называют конъюнкцию, полученную из *U*,

удалением из *U* некоторых переменных.

Например, если *U*= , то собственными частями функции *U* являются конъюнкции , , .

Элементарная конъюнкция *U* =  называется *простой импликантой* булевой функции

*f*( ), если *U* является импликантой функции *f* и никакая собственная часть *U* не является импликантой *f* .

Конституента единицы в совершенной ДНФ функции *f* всегда является её импликантой. Чем меньше переменных содержит импликанта функции *f* по сравнению с конституентой единицы, тем больше единиц функции она покрывает.

**Теорема.** *Всякая булева функция f*() *может быть представлена как дизъюнкция всех простых импликант.*

Дизъюнкция всех простых импликант функции *f* называется сокращённой ДНФ этой функции.

**Методы минимизация булевых функций (БФ) Метод Квайна.**

Этот метод удобен для нахождения минимальной ДНФ функции от любого числа переменных.

Выражение  − называется элементарной конъюнкцией, *r* -ранг элементарной конъюнкции.

Идея метода Квайна (алгоритм)

1. Выписываем все элементарные конъюнкции из СДНФ функции. Проводим все возможные операции склеивания между элементарными конъюнкциями, согласно правилу *xy*  *xy*  *x*
2. Между полученными элементарными конъюнкциями проводим все возможные операции склеивания и поглощения до тех пор, пока это возможно. Элементарная конъюнкция которая не может больше участвовать в склеивании, является *простой импликантой* БФ.
3. Дизъюнкция простых импликантов приводит к *сокращённой* ДНФ.

*f* *x*1 , *x*2 ,..., *xn*   *k* , где *k*

*k*

− набор простых импликантов

1. Выбираем существенные импликанты. Для этого необходимо построить таблицу покрытия. *Таблица покрытия* − двумерная таблица, каждой строке которой взаимно однозначно соответствует простой импликант, а каждому столбцу − элементарная конъюнкция из заданной СДНФ. На пересечении *i* -ой строки и *j* -го столбца находится 1, если простой импликант входит (покрывает) в элементарную

конъюнкцию. Простой импликант покрывает элементарную конъюнкцию, если он содержится в данной элементарной конъюнкции. (Простой импликант покрывает элементарную конъюнкцию, если каждая переменная, входящая в простой импликант, входит и в элементарную конъюнкцию).

Выбираем минимальное количество простых импликантов (существенных), таким образом, чтобы были покрыты все элементарные конъюнкции.

1. Дизъюнкция существенных импликантов приводит к минимальной ДНФ.

**Пример:** Найти минимальную ДНФ (МДНФ) функции, заданной СДНФ:

*f* *x*1 , *x*2 , *x*3   *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3

1. 1)*x*1*x*2 *x*3  *x*1*x*2 *x*3  *x*2 *x*3 

2)*x*1*x*2 *x*3  *x*1*x*2 *x*3  *x*1*x*3 

3)*x x x*  *x x x*  *x x* 

1 2 3

1 2 3

1 2  простые импликанты

4)*x*1*x*2 *x*3  *x*1*x*2 *x*3  *x*2 *x*3 

2. *f* *x*1 , *x*2 , *x*3   *x*2 *x*3  *x*1 *x*3  *x*1 *x*2  *x*2 *x*3

− сокращённая ДНФ

3. Строим таблицу покрытия :

ростые

элементарные конъюнкции

пликанты

*x*1 *x*2 *x*3

*x*1 *x*2 *x*3 *x*1 *x*2 *x*3 *x*1 *x*2 *x*3 *x*1 *x*2 *x*3

*x*2 *x*3 *x*1 *x*3 *x*1 *x*2

*x*2 *x*3

1

1

0

0

0

0

1

1

0

0

0

0

1

0

1

0

0

0

1

1

4.

или

*f* *x*1 , *x*2 , *x*3   *x*2 *x*3  *x*1 *x*2  *x*2 *x*3 *f* *x*1 , *x*2 , *x*3   *x*1 *x*3  *x*2 *x*3  *x*2 *x*3

− минимальная ДНФ

**Вывод**: БФ может иметь несколько минимальных форм.

###### Метод Квайна − Мак-Класки

Каждой элементарной конъюнкции ставим в соответствие булев вектор. (x с отрицанием – 0, без отрицания – 1).

1. Выписываем все элементарные конъюнкции из СДНФ функции в формализованном виде в столбец, располагая их в порядке возрастания числа единиц в векторах и разбивая на классы по числу единиц.
2. Между элементарными конъюнкциями проводим все возможные склеивания. В результате склеивания получают конъюнкцию, в которой отсутствует одна из переменных. Такую конъюнкцию представляют также двоичным набором, где на место каждой отсутствующей переменной ставят прочерк или крестик. Результат записываем в новый столбец справа, а элементарные конъюнкции, участвовавшие в склеивании, помечаем звездочкой. **Склеивать можно только элементарные конъюнкции из соседних классов.**
3. После всех склеиваний полученные конъюнкции также разбиваются на классы по числу оставшихся в них единиц.
4. На следующем этапе склеиванию подлежат конъюнкции из нового столбца, которые расположены в соседних классах и, кроме того имеют символ отсутствия переменной в одно и той же позиции. Элементарные конъюнкции, участвовавшие в склеивании, помечаем звездочкой. После всех склеиваний полученные конъюнкции также разбиваются на классы по числу оставшихся в них единиц
5. Для полученного столбца применяем шаг 4 пока не будут найдены все склеиваемые конъюнкции .
6. Все элементарные конъюнкции, которые остались непомеченными звездочкой, являются простыми импликантами.
7. Выбираем существенные импликанты. Для этого строим таблицу покрытия.
8. Определяем минимальную ДНФ (дизъюнкция существенных импликантов)

**Пример1:** Найти минимальную ДНФ (МДНФ) функции, заданной СДНФ:

*f* *x*1 , *x*2 , *x*3   *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3  *x*1 *x*2 *x*3

|  |  |  |
| --- | --- | --- |
| 000\* | \_00 |  |
| 100\* | 1\_0 |  |

|  |  |  |
| --- | --- | --- |
| 110\*  011\* | 11\_  \_11 |  |
| 111\* |  |  |
|  |  |  |

*f* *x*1 , *x*2 , *x*3   *x*2 *x*3  *x*1 *x*3  *x*1 *x*2  *x*2 *x*3

− сокращённая ДНФ

Строим таблицу покрытия :

00

00

10

11

+

+

+

+

\_11

+

+

11\_

+

1\_0

+

\_00

11

Элементарные конъюнкции

ростые

пликанты

или

*f* *x*1 , *x*2 , *x*3   *x*2 *x*3  *x*1 *x*2  *x*2 *x*3 *f* *x*1 , *x*2 , *x*3   *x*1 *x*3  *x*2 *x*3  *x*2 *x*3

− минимальная ДНФ

**Вывод**: БФ может иметь несколько минимальных форм.

**Пример2:** Найти минимальную ДНФ (МДНФ) функции, заданной СДНФ:

*f( )=*



|  |  |  |
| --- | --- | --- |
| 001\*  000 | 0\_01\*  00\_1\* | 0\_ \_1 |
| 011\*  101\* | 01\_1\*  \_101\*  0\_11\* | \_1\_1 |
| 111\*  101\* | \_111\* 11\_1\* |  |

|  |  |  |
| --- | --- | --- |
| 111\* |  |  |

*f( )= –* сокращённая ДНФ

Строим таблицу покрытия :

00

01

11

10

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Простые  мпликанты | Элементарные конъюнкции | | | | | | |
| 001 | 000 | 011 | 101 | 111 | 101 | 111 |
| 1000 |  | + |  |  |  |  |  |
| 0\_ \_1 | + |  | + | + | + |  |  |
| \_1\_1 |  |  |  | + | + | + | + |

*f( )= * *--* минимальная ДНФ.

###### Минимизация с помощью диаграммы Карно (Вейча).

Диаграмма Карно выглядит как прямоугольная таблица с количеством клеток, зависящим от числа переменных булевой функции, для которой она строится.

Общее правило построения диаграмм Карно состоит в следующем. Число n переменных булевой функции делится на два. Если n нечётное, то строкам диаграммы ставятся в соответствие наборы значений m=n/2+1 первых булевых переменных, а столбцам – наборы значений оставшихся n-m булевых переменных. Если n – чётное, то строкам и столбцам диаграммы соответствуют наборы m=n/2 переменных.

При построении прямоугольной таблицы с m строками и n-m столбцами, расположение наборов переменных функции по строкам и столбцам должны удовлетворять следующему условию – соседние наборы должны отличаться не более чем в одной позиции (разряде). Например, для двух и трёх булевых переменных соответственно имеем:

00 01 11 10

000 001 011 010 110 111 101 100

Прямоугольная таблица заполняется значением булевой функции для значений переменных соответствующих столбцу и строке расположения клетки таблицы.

Например: для булевой функции

*f( )=*



построим диаграмму Карно.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 00 | 0 | 1 | 1 | 0 |
| 01 | 0 | **1** | **1** | 0 |
| 11 | 0 | **1** | 1 | 0 |
| 10 | 1 | 0 | 0 | 0 |



Далее на диаграмме нужно выделить одну или группу () рядом стоящих единиц, таким образом

, чтобы перекрыть одним выделением как можно больше единиц. При этом необходимо учитывать , что верхние и нижние ряды клеток рассматриваются как соседние. Аналогично следует рассматривать

соседними левые и правые столбцы клеток. Выделенные группы должны образовывать или одну клетку, или линию, или прямоугольник, или квадрат. И должны быть выделены все единицы на диаграмме.

По выделенным группам составляем конъюнкции (при этом изменяемые переменные в группе в конъюнкции не перечисляются) и дизъюнктивно их объединяем.

Таким образом получаем ДНФ:

*f= .*

Если выделенные группы содержат только максимально возможное количество единиц и количество групп минимально возможно, то полученная ДНФ – минимальна.

Ещё один пример:

Для

*f( )=*



найдем минимальную ДНФ с помощью диаграммы Карно.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |

*f=*

Обратите внимание, что группа выделенная зеленым цветом образует квадрат, если учитывать, что верхние и нижние ряды клеток рассматриваются как соседние и аналогично следует рассматривать соседними левые и правые столбцы клеток.

Лекция. Сравнения. Кольцо классов вычетов.

Рассмотрим множество *Z* и рассмотрим на этом множестве *операцию деления с остатком.*

Справедлива следующая *теорема 1*. *Теорема 1*.

Для любых целых чисел *a* и *b*, *b*  0 , существуют *единственные* целые числа *q* и *r*, 0  *r* | *b* | , такие, что *a = bq + r*.

При этом – *r* называют *остатком*, а *q* − *частным* (*неполным частным* при *r*  0 ) от деления *a* на

*b*. Если остаток *r = 0*, то говорят, что *b* делит *a нацело*, и обозначают *b*|*a*. Еще раз подчеркнем, что остаток у нас – *целое неотрицательное число*.

Поскольку при изучении сравнений рассматривается деление с остатком целого числа на *целое положительное число* (*натуральное*), перейдем к таким случаям.

Деление *целых положительных чисел* удобно выполнять *столбиком*, этот способ позволяет получить и неполное частное (или просто частное) и остаток.

Чтобы получить неполное частное от деления *целого отрицательного* числа *a* на *целое положительное* число *b* нужно взять число, противоположное неполному частному от деления *модулей исходных чисел* и *вычесть из него еди*ницу, после чего остаток *r* вычислить по формуле *r*  *a*  *bq* .

Рассмотрим *примеры*.

Найдем остаток от деления числа *19* на *4*. Имеем: частное, *q* = *4*, остаток, *r*  19  4  4  3 .

Теперь найдем остаток от деления *-19* на *4*. Делим модули исходных чисел, получаем *4*, меняем знак и отнимаем единицу, получаем *q* = -5. Теперь найдем остаток, *r*  19  4  (5)  1.

Перейдем к сравнениям. Справедлива следующая *теорема 2*. *Теорема 2*.

Пусть *n* – *натуральное число*, *n* > *1*.

Для любых целых чисел *a* и *b* следующие условия *равносильны*:

* 1. *a* и *b* имеют одинаковые остатки от деления на *n*;
  2. *(a − b)* делится на *n* без остатка, то есть *a – b = nq* для подходящего целого *q*;
  3. *a = b + nq* для некоторого целого *q*.

Доказательство проводится по схеме: 1  2  3  1. Покажем, например, что 1  2 .

Из условия *1* имеем:

*a*  *nq*1  *r*

и *b*  *nq*2  *r* , 0  *r*  *n* , *q*1,

*q*2  *Z* , тогда

*a*  *b*  *n*(*q*1  *q*2 )  *nq*3 , *q*3  *Z* , что и означает

делимость *(a − b)* на *n*.

Целые числа *а* и *b* называются *сравнимыми по модулю n*, если они удовлетворяют одному из

условий *теоремы 2*, что обозначается

*a*  *b*(mod *n*)

или в эквивалентной форме

*a*  *b*  0(mod

*n*) .

Может быть использовано и краткое обозначение: *a*  *b*(*n*) .

Данное соотношение между целыми числами называют *сравнением по модулю n*.

Например,

###  8  3(mod

5)  7(mod

### 5)  12(mod

5)  2(mod

5) .

Как мы знаем, остаток *r* при делении целого числа *a* на число *n*  *N*

*всегда не отрицателен* и находится

в диапазоне 0  *r*  *n* . Все указанные числа *-8, -3, 7, 12* дают при делении на *5* один и тот же остаток, *r =*

*2*. Сравнение означает, что  8  2(mod 5) , и  8  2  5  2 .

Очевидно что, *n*  0(mod

1. .

Рассмотрим *основные свойства сравнений*.

* 1. Пусть

*a*  *b*(mod *n*) . Тогда

(*a*  *c*)  (*b*  *c*)(mod

*n*) для любого

*c*  *Z* . Это соотношение

означает, что к обеим частям сравнения можно *добавить*, или *вычесть* из обеих частей, *одно и то же*

*число*. Например: 3  11(mod

4) отсюда следует, что

### 6  14(mod

4) и 1  9(mod

4) .

* 1. Сравнения можно почленно *складывать* или *вычитать*. Если

*a*  *b*(mod *n*)

и *с*  *d* (mod *n*) ,

то (*a*  *c*)  (*b*  *d* )(mod

*n*) . Например:

### 7  3(mod 4)

и 10  6(mod 4)

отсюда следует, что

### 17  9(mod

4)  1(mod

4) .

* 1. Сравнения можно почленно *перемножать*. Если

*a*  *b*(mod *n*) и

*с*  *d* (mod *n*) , то

*ac*  *bd* (mod *n*) . Перемножим сравнения из предыдущего свойства, имеем 70  18(mod

### 4)  2(mod

4) .

* 1. Сравнения можно почленно возводить в любую *натуральную степень*, *m*, если

*am*  *bm* (mod *n*) , Например,

*a*  *b*(mod *n*) , то

*m*  *N* .

, отсюда следует что .

*Обратное заключение не верно !!!*

Например,

### 5  5(mod

##### 52  72 (mod

6) .

##### 6)  1mod

6 , но *5* не сравнимо с *7* по модулю *6*:

### 7(mod

6)  1(mod

6) и

* 1. Если в сравнении

*a*  *b*(mod *n*)

числа *a*, *b* и *n* имеют *общий множитель d*, то на него

сравнение можно *сократить*:

*a* / *d*  *b* / *d* (mod

*n* / *d* ) . Например,

### 6  10(mod

4) , отсюда следует, что

### 3  5(mod

2) .

* 1. Операция *деления* для сравнений *отсутствует!!!* Если, *a*1*d*  *b*1*d* (mod

*n*) , то это *не всегда* означает, что

*a*1  *b*1 (mod

*n*) . К примеру,

### 3 2  5  2(mod

4) , но *3* не сравнимо с *5* по модулю *4* . Однако свойство

*сократимости сравнений* можно сохранить в часто встречающемся случае, когда *d* и *n взаимно просты*.

Если *a*1*d*  *b*1*d* (mod

*n*) , и *НОД(d,n) = 1*, то сравнение можно сократить на общий множитель, взаимно

простой с модулем *n*: из сравнения

*a*1*d*  *b*1*d* (mod

*n*) , тогда следует сравнимость

*a*1 и *b*1

по модулю *n*,

*a*1  *b*1(mod *n*)

Например, 10  22(mod

3),

###  5  11(mod

3) . В нашем случае *НОД(2,3) = 1*.

* 1. Если *a*  *b*(mod *m*  *n*) , то *a*  *b*(mod *m*) и

*a*  *b*(mod *n*) .

Например 14  38  2(mod

12) , отсюда 14  38  2(mod 3)

### и 14  38  2(mod

4) .

С помощью сравнений можно доказывать тождества.

*Пример 1*. Требуется доказать, что 2*n* 1 кратно *3* тогда и только тогда, когда *n* четно.

Поскольку

### 2  1(mod

3) , то возведя сравнение в степень *n* получим,

2*n*  (1)*n* (mod

3) . Теперь

отнимем *1* от обеих частей сравнения, получим:

2*n* 1  ((1)*n* 1)(mod

3) . А это означает, что

##### 2*n* 1  0(mod

3) , если *n* четно.

Рассмотрим еще примеры использования сравнений.

*Пример 2*. Требуется найти с каким числом сравнимо

146 2 по модулю *21*. Имеем: 147  0(mod

21) .

Отсюда

### 147 1  146  1(mod

21) . Возведем обе части этого сравнения в квадрат, получим:

##### 146 2  (1)2 (mod

1.  1(mod

21) .

*Пример 3*. Требуется найти последнюю цифру числа

131028 .

Запишем требования задачи через сравнение: 131028  *x*(mod10) . Необходимо найти *x*. Имеем:

13  3(mod10) ;

### 134  34  1(mod10) ;

..

Получили, что последняя цифра нашего числа – *1*.

Свойство сравнимости целых чисел по данному модулю *n* определяет бинарное отношение на множестве *Z*.

*R*mod *n*

Два целых числа находятся в отношении сравнимости

*R*mod *n* , тогда и только тогда, когда они

сравнимы друг с другом по модулю *n*. Отношение сравнимости является *отношением эквивалентности*

и для него выполняются свойства:

* 1. рефлексивно, поскольку *a*  *a*(mod *n*) для каждого *a*  *Z* ;
  2. симметрично, поскольку, если *a*  *b*(mod*n* ) , то *b*  *a*(mod*n* ) для целых чисел *а* и *b*;
  3. транзитивно, поскольку, если

*a*  *b*(mod*n* ) и

*b*  *c*(mod*n* )

, то *a*  *c*(mod*n* ) .

Множество *Z* разбивается отношением сравнимости по модулю *n* на *классы эквивалентности*. Множество всех классов эквивалентности по модулю *n* называется *множеством классов вычетов по модулю n*.

В каждом классе вычетов по модулю *n* находятся сравнимые между собой по модулю *n* элементы, дающие при делении на *n* одинаковый остаток.

Например, пусть *n = 3*.

Имеем *3* класса эквивалентности по модулю *3*, так что множество классов вычетов содержит три элемента: *{[0], [1], [2]}*. В первом классе эквивалентности, *[0]* находятся все числа кратные *3,* и число *0*, как представитель этого класса. Во втором классе, *[1]* – все числа, дающие остаток *1* при делении на *3*, и число *1* как представитель класса. И наконец, в третьем классе, *[2]* – все числа, дающие остаток *2* при делении на *3*, и число *2* как представитель этого класса.

Как мы видим, для произвольного числа *n*  *N* , *n > 1*, множество классов вычетов по модулю *n*, состоит

точно из *n* различных классов вычетов: *[0]*, *[1], [2],…, [n-1]*.

Для каждого 0  *r*  *n* класс вычетов *[r]* состоит точно из тех целых чисел *а*, для которых *a*  *r*(mod*n* ) . На множестве классов вычетов можно определить *операции сложения* и *умножения*.

Если *[а]* – класс вычетов по модулю *n*, содержащий элемент *а*, и *[b]* — класс вычетов по модулю *n*, содержащий *b*, то сложение и умножение определяется соотношениями:

[*a*] [*b*]  [*a*  *b*]

[*a*] [*b*]  [*a*  *b*]

Слева операции  ,  выполняются между классами эквивалентности, Справа – между целыми числами, в результате получаем число, являющееся *представителем* нового класса эквивалентности.

Например, для *n = 4*, имеем *4* класса эквивалентности: *{[0], [1], [2],[3]}*. Находим:

[3] [2]  [3  2]  [5]  [1], поскольку справедливо, 5  1(mod 4) .

[3] [2]  [3 2]  [2]  [2], поскольку 6  2(mod

4) .

Составим таблицы сложения и умножения (*таблицы Кэли*) для классов вычетов по модулю *4*.

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

Классы вычетов умножения по *вычетов*, если *n поле Галуа*. абелевую группу.

с определенными выше модулю составного числа *n*

– простое число, то получаемая Классы вычетов с операцией

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

операциями сложения и образуют *кольцо классов* алгебраическая система – сложения образуют

Лекция. Сравнения. Кольцо классов вычетов.

Рассмотрим множество *Z* и рассмотрим на этом множестве *операцию деления с остатком.*

Справедлива следующая *теорема 1*. *Теорема 1*.

Для любых целых чисел *a* и *b*, *b*  0 , существуют *единственные* целые числа *q* и *r*, 0  *r* | *b* | ,

такие, что *a = bq + r*.

При этом – *r* называют *остатком*, а *q* − *частным* (*неполным частным* при *r*  0 ) от деления *a* на

*b*. Если остаток *r = 0*, то говорят, что *b* делит *a нацело*, и обозначают *b*|*a*. Еще раз подчеркнем, что остаток у нас – *целое неотрицательное число*.

Поскольку при изучении сравнений рассматривается деление с остатком целого числа на *целое положительное число* (*натуральное*), перейдем к таким случаям.

Деление *целых положительных чисел* удобно выполнять *столбиком*, этот способ позволяет получить и неполное частное (или просто частное) и остаток.

Чтобы получить неполное частное от деления *целого отрицательного* числа *a* на *целое положительное* число *b* нужно взять число, противоположное неполному частному от деления *модулей исходных чисел* и *вычесть из него еди*ницу, после чего остаток *r* вычислить по формуле *r*  *a*  *bq* .

Рассмотрим *примеры*.

Найдем остаток от деления числа *19* на *4*. Имеем: частное, *q* = *4*, остаток, *r*  19  4  4  3 .

Теперь найдем остаток от деления *-19* на *4*. Делим модули исходных чисел, получаем *4*, меняем знак и отнимаем единицу, получаем *q* = -5. Теперь найдем остаток, *r*  19  4  (5)  1.

Перейдем к сравнениям. Справедлива следующая *теорема 2*. *Теорема 2*.

Пусть *n* – *натуральное число*, *n* > *1*.

Для любых целых чисел *a* и *b* следующие условия *равносильны*:

1. *a* и *b* имеют одинаковые остатки от деления на *n*;
2. *(a − b)* делится на *n* без остатка, то есть *a – b = nq* для подходящего целого *q*;
3. *a = b + nq* для некоторого целого *q*. Доказательство проводится по схеме: 1  2  3  1. Покажем, например, что 1  2 .

Из условия *1* имеем:

*a*  *nq*1  *r*

и *b*  *nq*2  *r* , 0  *r*  *n* , *q*1,

*q*2  *Z* , тогда

*a*  *b*  *n*(*q*1  *q*2 )  *nq*3 , *q*3  *Z* , что и означает

делимость *(a − b)* на *n*.

Целые числа *а* и *b* называются *сравнимыми по модулю n*, если они удовлетворяют одному из

условий *теоремы 2*, что обозначается

*a*  *b*(mod *n*)

или в эквивалентной форме

*a*  *b*  0(mod

*n*) .

Может быть использовано и краткое обозначение: *a*  *b*(*n*) .

Данное соотношение между целыми числами называют *сравнением по модулю n*.

Например,

###  8  3(mod

5)  7(mod

### 5)  12(mod

5)  2(mod

5) .

Как мы знаем, остаток *r* при делении целого числа *a* на число *n*  *N*

*всегда не отрицателен* и находится

в диапазоне 0  *r*  *n* . Все указанные числа *-8, -3, 7, 12* дают при делении на *5* один и тот же остаток, *r =*

*2*. Сравнение означает, что  8  2(mod 5) , и  8  2  5  2 .

Очевидно что, *n*  0(mod

*n*) .

Рассмотрим *основные свойства сравнений*.

1. Пусть

*a*  *b*(mod *n*) . Тогда

(*a*  *c*)  (*b*  *c*)(mod

*n*) для любого

*c*  *Z* . Это соотношение

означает, что к обеим частям сравнения можно *добавить*, или *вычесть* из обеих частей, *одно и то же*

*число*. Например: 3  11(mod

4) отсюда следует, что

### 6  14(mod

4) и 1  9(mod

4) .

1. Сравнения можно почленно *складывать* или *вычитать*. Если

*a*  *b*(mod *n*)

и *с*  *d* (mod *n*) ,

то (*a*  *c*)  (*b*  *d* )(mod

*n*) . Например:

### 7  3(mod 4)

и 10  6(mod 4)

отсюда следует, что

### 17  9(mod

4)  1(mod

4) .

1. Сравнения можно почленно *перемножать*. Если

*a*  *b*(mod *n*) и

*с*  *d* (mod *n*) , то

*ac*  *bd* (mod *n*) . Перемножим сравнения из предыдущего свойства, имеем 70  18(mod

### 4)  2(mod

4) .

1. Сравнения можно почленно возводить в любую *натуральную степень*, *m*, если

*am*  *bm* (mod *n*) , Например,

*a*  *b*(mod *n*) , то

*m*  *N* .

, отсюда следует что .

*Обратное заключение не верно !!!*

Например,

### 5  5(mod

##### 52  72 (mod

6) .

##### 6)  1mod

6 , но *5* не сравнимо с *7* по модулю *6*:

### 7(mod

6)  1(mod

6) и

1. Если в сравнении

*a*  *b*(mod *n*)

числа *a*, *b* и *n* имеют *общий множитель d*, то на него

сравнение можно *сократить*:

*a* / *d*  *b* / *d* (mod

*n* / *d* ) . Например,

### 6  10(mod

4) , отсюда следует, что

### 3  5(mod

2) .

1. Операция *деления* для сравнений *отсутствует!!!* Если, *a*1*d*  *b*1*d* (mod *n*) , то это *не всегда* означает, что

*a*1  *b*1 (mod

*n*) . К примеру,

### 3 2  5  2(mod

4) , но *3* не сравнимо с *5* по модулю *4* . Однако свойство

*сократимости сравнений* можно сохранить в часто встречающемся случае, когда *d* и *n взаимно просты*.

Если *a*1*d*  *b*1*d* (mod

*n*) , и *НОД(d,n) = 1*, то сравнение можно сократить на общий множитель, взаимно

простой с модулем *n*: из сравнения

*a*1*d*  *b*1*d* (mod

*n*) , тогда следует сравнимость

*a*1 и *b*1

по модулю *n*,

*a*1  *b*1(mod *n*)

Например, 10  22(mod

3),

###  5  11(mod

3) . В нашем случае *НОД(2,3) = 1*.

1. Если *a*  *b*(mod

*m*  *n*) , то *a*  *b*(mod *m*) и

*a*  *b*(mod *n*) .

Например 14  38  2(mod

12) , отсюда 14  38  2(mod 3)

### и 14  38  2(mod

4) .

С помощью сравнений можно доказывать тождества.

*Пример 1*. Требуется доказать, что 2*n* 1 кратно *3* тогда и только тогда, когда *n* четно.

Поскольку

### 2  1(mod

3) , то возведя сравнение в степень *n* получим,

2*n*  (1)*n* (mod

3) . Теперь

отнимем *1* от обеих частей сравнения, получим:

2*n* 1  ((1)*n* 1)(mod

3) . А это означает, что

##### 2*n* 1  0(mod

3) , если *n* четно.

Рассмотрим еще примеры использования сравнений.

*Пример 2*. Требуется найти с каким числом сравнимо

146 2 по модулю *21*. Имеем: 147  0(mod

21) .

Отсюда

### 147 1  146  1(mod

21) . Возведем обе части этого сравнения в квадрат, получим:

##### 146 2  (1)2 (mod

21)  1(mod

21) .

*Пример 3*. Требуется найти последнюю цифру числа

131028 .

Запишем требования задачи через сравнение: 131028  *x*(mod10) . Необходимо найти *x*. Имеем:

13  3(mod10) ;

### 134  34  1(mod10) ;

..

Получили, что последняя цифра нашего числа – *1*.

Свойство сравнимости целых чисел по данному модулю *n* определяет бинарное отношение на множестве *Z*.

*R*mod *n*

Два целых числа находятся в отношении сравнимости

*R*mod *n* , тогда и только тогда, когда они

сравнимы друг с другом по модулю *n*. Отношение сравнимости является *отношением эквивалентности*

и для него выполняются свойства:

1. рефлексивно, поскольку *a*  *a*(mod *n*) для каждого *a*  *Z* ;
2. симметрично, поскольку, если *a*  *b*(mod*n* ) , то *b*  *a*(mod*n* ) для целых чисел *а* и *b*;
3. транзитивно, поскольку, если

*a*  *b*(mod*n* ) и

*b*  *c*(mod*n* )

, то *a*  *c*(mod*n* ) .

Множество *Z* разбивается отношением сравнимости по модулю *n* на *классы эквивалентности*. Множество всех классов эквивалентности по модулю *n* называется *множеством классов вычетов по модулю n*.

В каждом классе вычетов по модулю *n* находятся сравнимые между собой по модулю *n* элементы, дающие при делении на *n* одинаковый остаток.

Например, пусть *n = 3*.

Имеем *3* класса эквивалентности по модулю *3*, так что множество классов вычетов содержит три элемента: *{[0], [1], [2]}*. В первом классе эквивалентности, *[0]* находятся все числа кратные *3,* и число *0*, как представитель этого класса. Во втором классе, *[1]* – все числа, дающие остаток *1* при делении на *3*, и число *1* как представитель класса. И наконец, в третьем классе, *[2]* – все числа, дающие остаток *2* при делении на *3*, и число *2* как представитель этого класса.

Как мы видим, для произвольного числа *n*  *N* , *n > 1*, множество классов вычетов по модулю *n*, состоит

точно из *n* различных классов вычетов: *[0]*, *[1], [2],…, [n-1]*.

Для каждого 0  *r*  *n* класс вычетов *[r]* состоит точно из тех целых чисел *а*, для которых *a*  *r*(mod*n* ) . На множестве классов вычетов можно определить *операции сложения* и *умножения*.

Если *[а]* – класс вычетов по модулю *n*, содержащий элемент *а*, и *[b]* — класс вычетов по модулю *n*, содержащий *b*, то сложение и умножение определяется соотношениями:

[*a*] [*b*]  [*a*  *b*]

[*a*] [*b*]  [*a*  *b*]

Слева операции  ,  выполняются между классами эквивалентности, Справа – между целыми числами, в результате получаем число, являющееся *представителем* нового класса эквивалентности.

Например, для *n = 4*, имеем *4* класса эквивалентности: *{[0], [1], [2],[3]}*. Находим:

[3] [2]  [3  2]  [5]  [1], поскольку справедливо, 5  1(mod 4) .

[3] [2]  [3 2]  [2]  [2], поскольку 6  2(mod

4) .

Составим таблицы сложения и умножения (*таблицы Кэли*) для классов вычетов по модулю *4*.

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

Классы вычетов умножения по *вычетов*, если *n поле Галуа*. абелевую группу.

с определенными выше модулю составного числа *n*

– простое число, то получаемая Классы вычетов с операцией

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

операциями сложения и образуют *кольцо классов* алгебраическая система – сложения образуют