ТЕМА №3*.* ***БУЛЕВЫ ФУНКЦИИ****.*

Пусть задано множество *B = {0, 1}*. Тогда, отображение **** множества  на множество  называется *булевой функцией* *n* *переменных* и ее можно записать в виде ****. 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 **** может быть представлен *n*- *разрядным* *двоичным числом*. Количество двоичных *n*- разрядных чисел равно ****. Поэтому любая булева функция *n* переменных может быть определена на **** наборах. И, таким образом, мы имеем набор **** значений функции *n* переменных. Каждому такому набору значений булевой функции мы можем поставить в соответствие **-** разрядное двоичное число. Количество ****- разрядных двоичных чисел равно **** и, 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* |  |  |  |  |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |

Функция **** тождественноравна *0*. Она называется *константой нуль* и обозначается ****.

Функция**** тождественно равназначениюпеременной *x* и называется *тождественной функцией* и обозначается *****.*

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 ****.

Функция **** принимает значения, противоположенные значениям переменной *x*. Эта функция называется *инверсией* или *отрицанием* и обозначается ****. Для обозначения инверсии будем также использовать символ логического отрицания, .

Функция **** тождественно равна *1*. Она называется *константой единица* и обозначается ****.

Константы нуль и единицу принято считать *нуль – местной* булевой функцией.

Функций двух переменных – *16*.

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

Рассмотрим основные булевы функции.

Функция **** - это известная нам *конъюнкция* или «*логическое и » (логическое умножени*е). Обозначение: , ,, *xy*.

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

,

,

.

Функция **** 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: .

Функция **** - *дизъюнкция* или «*логическое или» (логическое сложение)*. Обозначение: , . Функция равна нулю тогда и только тогда, когда *все* ее переменные *равны нулю*. Для дизъюнкции справедливы следующие соотношения:

,

,

.

Функция **** называется *операцией Пирса* или *стрелкой Пирса*. Обозначение: . Ее можно выразить через операции отрицания и дизъюнкции двух переменных: .

Функция**** – это *эквивалентность* (*логическая равнозначность*). Обозначение: .

Функция **** - это *импликация* от *y* к *x*. Обозначение: .

Функция **** - это *импликация* от *x* к *y*. Обозначение: .

Функция **** называется *операцией Шеффера* или *штрих Шеффера*.

Обозначение: . 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 **** путем применения, возможно многократного, этих двух правил, будем называть *суперпозицией функций*****.

Например, функция  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,  .

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

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *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: 

- *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.

.

- Можно перечислить *номера тех наборов, на которых функция равна единице:* . Напомним, что нумерация наборов начинается с нуля.

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

Булева функция **** *существенно* *зависит* от переменной ****, если существует такой набор значений ****, чтовыполняется соотношение:

****

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

В противном случае говорят, что от переменной **** функция зависит несущественно и **** является ее *фиктивной* переменной.

В функциях **, ** из 16 элементарных булевых функций, фиктивной является переменная *y*.

В булевой функции , заданной следующей таблицей истинности, переменная *z* является фиктивной.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *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 ****. Функция**** называется *двойственной* к функции ****, если выполняется соотношение:

****

Можно записать и так:

****

Это означает, что функции**** и на противоположенных наборах имеют противоположенные значения.

Очевидно, что:

****

Рассмотрим примеры двойственных функций.

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

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

Функция называется *самодвойственной*, если:

****

Примеры самодвойственных функций: , , .

Рассмотрим о*сновные логические эквивалентности* для булевых функций.*.*

1. *Коммутативность.*

.

.

.

.

*2. Ассоциативность*.

.

.

.  
 

3*. Дистрибутивность*.

.

.

.

*4. Закон двойного отрицания.*

*.*

*5. Законы де Моргана.*

*.*

**

*6. Законы идемпотентности.*

**

**

*7. Законы поглощения.*

**

**

*8. Другие эквивалентности.*

**

**

**

**

**

**

**

**

**

Имеют место следующие очевидные утверждения:

*.*

**

***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:  Всякую конъюнкцию переменных функции, взятых с отрицанием или без отрицания, при этом каждая переменная встречается не более одного раза, называют *элементарной.* Например,  - элементарная конъюнкция,  - 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*, сокращенно *ДНФ.*

Пусть булева функция задана через *ДНФ*: ****, где  - элементарные конъюнкции. Число *s* называют *длиной ДНФ*.

*ДНФ* можно охарактеризовать также числом , которое называют *суммарным рангом* этой ДНФ. Здесь  - *ранг i* – *ой* элементарной конъюнкции.

Например, для булевой функции ****, ее суммарный ранг ее *ДНФ*, *R = 5*.

Пусть имеется элементарная конъюнкция . *Собственной частью* конъюнкции *U* называют конъюнкцию, полученную из *U*, удалением из *U* некоторых переменных.

Например, если , то собственными частями конъюнкции *U* являются, например, конъюнкции: , ,  и т. д.

Аналогично вводится понятие и для *элементарной дизъюнкции*. Например,  - 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 , где *i* - номер набора, на котором значение  равно единице. Например,  принимает значение *1* на единственном наборе - *101* .

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, .

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* *******, 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 **, 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:

****.

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 , где *i* - номер набора, на котором значение  равно нулю. Например, значение  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  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, .

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* *******, 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 **** из предыдущего примера.

Имеем:

****

Перейдем к *полиному Жегалкина*.

Справедлива следующая теорема Жегалкина.

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

****

**где** ** -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 *,***если *1*, оставим без изменения. Между полученными **таким** образом выражениями возьмем сложение по модулю два. Раскроим скобки, применив дистрибутивный закон: 

Это может быть, например, выражение вида:

.

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

****

Построим полином Жегалкина для функции **** из предыдущего примера, заданной таблицей истинности.

Имеем:

****

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** ****. 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** . **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:**



где  - 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:**

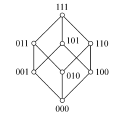
****.

Рассмотрим множество двоичных наборов функции *n* переменных. На этом множестве введем *отношение сравнимости двоичных наборов* следующим образом. Говорят, что двоичный набор **** не больше двоичного набора ****, обозначается , если для *каждой пары* **** справедливо соотношение: ****. 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 , но . Таким образом, условие монотонности для функции двух переменных мы можем записать в виде: ****.

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: ,  и , а отношение **** 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  и *f(a ) > f(b)*.**

**Чтобы сказать, что заданная булева функция монотонна, следует убедиться, что на *всех сравнимых наборах* выполняется условие монотонности .**

**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:**

****.

Или, что то же самое: ****.

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 ****. Соответственно, функция самодвойственная, если на *всех парах* противоположенных наборов выполняется условие ****.

Зададим булеву функцию  таблицей истинности и проверим ее на принадлежность пяти классам *Поста*. Для нашей функции сначала выполняется импликация , затем эквивалентность между значением *x* и результатом импликации.

|  |  |  |  |
| --- | --- | --- | --- |
| *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:   
   Получили полином второй степени, функция не является линейной.
4. Она не является монотонной, поскольку *f(0,1,1)* < *f(0,1,0)* (набор больше, значение функции меньше).
5. 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 **** называется *полной* (*функционально полной*), если любая булева функция может быть выражена через функции системы  с помощью суперпозиций (т. е. составления сложных функций).

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  - также является полной системой.

Рассмотрим критерий построения полной системы, который дает теорема Поста.

*Теорема Поста*.

Для того, чтобы система булевых функций **** была полной, необходимо и достаточно, чтобы для каждого из классов ** ,  , *L*, *M*, и *S* нашлась функция  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* | | | | |
|  |  | *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, является полной системой.
2. Штрих Шеффера,  также пример полной системы.
3. Импликация от *x* к *y*,  и инверсия *x*,  - полная система.
4. Конъюнкция,  и инверсия *x*.
5. Дизъюнкция  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 , ,  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 **** 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:

 и .

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.*

,

где *A* – некоторая элементарная конъюнкция. В этом случае говорят, что конъюнкции  и  *склеиваются по переменной* *x.* Например:

.

Здесь  и две заданные конъюнкции были склеены по переменной *,* 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.*

,

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

Например,

.

Третья операция – *операция неполного склеивания*:

.

Введем некоторые понятия.

Булева функция **** называется *импликантой* функции ****, 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: .

Рассмотрим импликанты функции .

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *x* | *y* |  |  |  |  |  |
| *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 , , , являются импликантами функции , функция  - нет. Термин *«импликанта»* возник ввиду того, что импликация значений функций  и  по всем наборам тождественно равна единице.

Конституента единицы в *СДНФ* функции *f* всегда является её импликантой. Для функции  функции  и  являются конституентами елиницы из ее *СДНФ*.

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* по сравнению с конституентой единицы из ее *СДНФ*, тем больше единиц функции она покрывает.

Элементарная конъюнкция  называется *простой импликантой* булевой функции ****, 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:

****.

Укажем импликанты и простые импликанты нашей функции.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | *y* | *z* | *f(x, y, z)* |  |  |  |  |  |
| *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  является импликантой нашей функции, ее собственная часть  - простой импликантой, поскольку ни , ни  не являются импликантами. Аналогично имеем, ,  - импликанты нашей функции,  - простая импликанта;  покрывает 4 единицы нашей функции. Простые импликанты в таблице выделены серым цветом.

Справедливы следующие теоремы.

*Теорема 1.*

*Всякая булева функция* **** *может быть представлена как дизъюнкция всех своих простых импликант.*

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 ****.
2. 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.
3. Дизъюнкция простых импликант приводит к *сокращённой* ДНФ булевой функции: , где  − простая импликанта.
4. 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.
5. Выбираем *существенные импликанты*, т.е. такое *минимальное количество* простых импликант, чтобы они *совместно* своими единицами покрывали все колонки импликантной матрицы.
6. *Дизъюнкция существенных импликант приводит к МДНФ.*

Рассмотрим пример. Найдем *МДНФ* булевой функции, заданной *СДНФ*:

.

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

.  
.  
.  
.  
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:  
   .
2. Строим таблицу покрытия.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Простые*  *импликанты* | *Конституенты единицы* | | | | |
|  |  |  |  |  |
|  | *1* | *1* | *0* | *0* | *0* |
|  | *0* | *1* | *1* | *0* | *0* |
|  | *0* | *0* | *1* | *0* | *1* |
|  | *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*:

.

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

.

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*.



 .

Рассмотрим еще пример.

.

Проведем операции неполного склеивания.

1. .
2. .
3. .
4. .
5. .  
   Все конституенты единицы участвовали в склеивании, у нас нет операции поглощения. Импликанта  больше не участвует в склеивании, это простая импликанта. Оставшиеся импликанты могут склеиваться между собой, продолжим склеивание.
6. .
7. .

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

.

Построим таблицу покрытия.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Простые*  *импликанты* | *Конституенты единицы* | | | | |
|  |  |  |  |  |
|  | 0 | 1 | 1 | 0 | 0 |
|  | 1 | 1 | 0 | 1 | 1 |

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

Как видим, обе простые импликанты входят *МДНФ*:

