**3. Математическая логика**

1. Высказывание.
2. Таблицы истинности.
3. Булевы функции двух переменных.
4. Дизъюнктивная совершенная нормальная форма логической функции.
5. Конъюнктивная совершенная нормальная форма логической функции.
6. Законы булевой алгебры.
7. Минимальные дизъюнктивная и конъюнктивная формы функции.
8. Минимальная и тупиковая ДНФ.
9. Минимальная КНФ.
10. Релейно-контактные устройства.

**Математическая логика** – это набор формальных правил, записанных специальным математическим языком и позволяющих правильно организовать работу компьютеру. Основным объектом математической логики является высказывание.

**Высказывание** – это повествовательное предложение, о котором в данной ситуации можно вполне определенное сказать **истинно** оно или **ложно**.

Высказывания будем обозначать большими латинскими буквами *Р*, *Q*, логические переменные *х*, *y*, которые принимают значения 0 или 1 из множества 

Из нескольких высказываний (**простых**) можно образовывать новые высказывания (**сложные**). Следовательно, из логических переменных можно составлять новые конструкции, которые образуют формулы **алгебры логики** с помощью логических связок. На русском языке эти связки представлены союзами «**и**», «**или**», «**не**», словами «**если…то**», «**тогда только тогда**».

**Пример**. Простые высказывания: «Студенты на занятиях», «На улице идет снег», «Иванов – студент группы ФК-211».

Сложные высказывания: «На улице холодно и идет снег», «Если студент Иванов долго занимается вечером, то утром он может опоздать на занятия».

**Функцией** алгебры логики (логической функцией, булевой функцией *n* аргументов) называется *n*-арная логическая операция (*n*-мерный двоичный вектор) на множестве *E*, т. е. *f* (*x*1, *x*2,…, *xn*): 

Принято множество булевых функций обозначать так: 

Любая логическая функция является сложным высказыванием, любая логическая переменная является простым высказыванием.

**Пример**. Пусть имеется сложное высказывание «Если на улице непогода, то следует взять с собой зонтик и одеться потеплей». При введенных простых высказываниях *А* – «На улице непогода», *В* – «Взять зонтик», *С* – «Одеться потеплее» логическую функцию  можно записать так: «если *А*, то *В* и *С*».

Так как число различных значений аргументов конечно, то любую булеву функцию можно просто задать таблично, указав, в какой из своих «точек» она принимает значение 0, а в какой 1. Такое табличное значение булевой функции называют **таблицей истинности**. Если число аргументов ***n***, то количество строк в таблице истинности булевой функции *f* равно **2*n***. Число булевых функций от ***n*** переменных составляет ****

Для функции одного аргумента таблица истинности состоит из четырех

значений (табл. 4):

1. функция–константа «ложь»: 0(*х*) = 0 при любом *х*;
2. функция–константа «истина»: 1(*х*) = 1 при любом *х*;
3. функция–повторения *r*(*х*) = *x* при любом *х*;
4. функция–отрицания  при *х* = 1 и  при *х* = 0.

*Таблица 4*

|  |  |
| --- | --- |
| Значения переменных | Значения функций |
| *x* | «0» | «1» | *r*(*x*) |  |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |

Булевых функций двух переменных *x* и *y* всего 16  (табл. 5).

*Таблица 5*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *х* | *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 |
| Обозна-чения |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Прокомментируем некоторые.

1.  – **тождественный ноль**, при любых значениях переменных эта функция принимает значение «0» – **ложь**.
2.  – **тождественная единица**, при любых значениях переменных эта функция принимает значение «1» – **истина**.
3. **Дизъюнкция**. Функция  называется **дизъюнкцие**й, соответствует союзу «**или**», иногда называется **логическим сложением** (табл. 6).

Данная функция истина, когда истинно хотя бы одно высказывание.

1. **Конъюнкция**. Функция  называется **конъюнкцией**, соответствует союзу «**и**», иногда называется **логическим умножением** (табл. 7).

Данная функция истина, если истины оба высказывания.

1. **Импликация**. Функция  называется **импликацией**.

Переменная *х* называется посылкой импликации (причиной), а высказывание *y* называется заключением (выводом, следствием). Все высказывание ложно, если посылка истина, а вывод является ложным (табл. 8).

1. **Эквивалентность**. Функция  называется **эквивалент-ностью**. Иногда эту функцию называют **равнозначностью**. Функция истина, если истинные значения высказываний совпадают (табл. 9).
2. **Неравнозначность**. Функция  называется **неравнознач-ность**. Истинной эта функция является при несовпадении истинности высказываний (табл. 10).
3. **Стрелка Пирса**. Функция  называется **стрелкой Пирса**. Она является истинной, когда оба высказывания ложны (табл. 11). Ее можно посчитать по формуле: 
4. **Штрих Шеффера**. Функция  называется **штрихом** **Шеффера**. Она является ложной, если оба высказывания истины (табл. 12). Ее можно вычислить по формуле: 

*Таблица 6 Таблица 7 Таблица 8 Таблица 9 Таблица 10*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | *y* |  |  | *x* | *y* |  |  | *x* | *y* |  |  | *x* | *y* |  |  | *x* | *y* |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

 *Таблица 11 Таблица 12*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | *y* |  |  |  |  |  | *x* | *y* |  |
| 0 | 0 | 1 |  |  |  | 0 | 0 | 1 |
| 0 | 1 | 0 |  |  |  | 0 | 1 | 1 |
| 1 | 0 | 0 |  |  |  | 1 | 0 | 1 |
| 1 | 1 | 0 |  |  |  | 1 | 1 | 0 |

Две последних функций являются отрицанием друг друга: 

При составлении логических формул придерживаются правил.

1. Последовательность логических связок одного типа, записанная без скобок, вычисляется слева направо.
2. Принимается, что при отсутствии скобок операции выполняются в следующем порядке: отрицание, конъюнкция, дизъюнкция, импликация, эвивалентность.

**Пример**. Записать логическими формулами следующее высказывание: «Если допоздна работаешь за компьютером и при этом пьешь много кофе, то утром просыпаешься в дурном расположении духа или с головной болью».

Решение*.* Введем простые высказывания:

1. *х* – «Допоздна работаешь за компьютером»;
2. *у* – «Пьешь много кофе»;
3. *z* – «Плохое настроение»;
4. *u* – «Головная боль».

Тогда сложное высказывание, заданное в условии задачи, можно записать формулой: 

**Пример**. Если исследования фирмы показывают, что потребитель отдает предпочтение качеству и многообразию выбора товара, то фирме следует сделать упор на улучшение качества товара или увеличение многообразия новых форм.

Решение*.* Введем простые высказывания:

1. *х* – «Отдавать предпочтение качеству»;
2. *у* – «Отдавать предпочтение многообразию выбора товара»;
3. *z* – «Делать упор на улучшение качества товара»;
4. *u* – «Увеличивать многообразие форм».

Тогда сложное высказывание, заданное в условии задачи, можно записать формулой: 

Как видно, одна и та же логическая формула может описывать разные по конкретному содержанию явления. Верно и обратное, одно и то же явление может описываться разными по внешнему виду формулами. Формулы, имеющие одинаковые таблицы истинности, будем называть **равносильными** (**эквивалентными**).

 Проверить равносильность формул можно двумя способами.

1. способом эквивалентных преобразований, который обычно применяется в «непрерывной» математике. Применяя этот способ, над одной из формул производят эквивалентные преобразования, пока не получат вторую.
2. сравнением таблиц истинности формул, что обычно применяется в дискретной математике. Две формулы эквивалентны, если они совпадают при всех возможных значениях аргумента, т. е. имеют одинаковые таблицы истинности.

**Пример**. Доказать равносильность формул 

 Решение. Составим таблицы истинности формул (табл. 13).

*Таблица 13*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | *y* |  |  |  |  |  |  |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

Формулы эквивалентны.

Логические формулы трех и более переменных также могут быть заданы таблицами истинности или формулами, состоящими из логических переменных и знаков унарных и бинарных операций.

**Пример**. Составить таблицу истинности для трех переменных по формуле:  (табл. 14).

*Таблица 14*

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

Любую функцию булевой алгебры можно представить через три функции:  – отрицание,  – дизъюнкцию,  – конъюнкцию. Формулы, содержащие только эти операции, называются **булевыми**. Алгебра на множестве всех логических функций с операциями отрицания, дизъюнкции и конъюнкции называется **булевой алгеброй** логических функций. Минимально возможный набор функций, через которые можно выразить остальные, называется **функционально полной системой** или **базисом**. Набор функций **** называется **конъюнктивно-дизъюнктивным базисом.**

Для любой логической формулы можно построить таблицу истинности. Но и обратное действие возможно – переход от табличного задания к формуле булевой алгебры. При этом следует придерживаться следующих действий:

1. для каждого набора значений переменных  (любая переменная – первичный **терм**), на котором логическая функция  равна **1**, выписываются **конъюнкции всех переменных**;
2. над теми переменными, значения которых равны **0**, ставится **отрицание**;
3. все полученные конъюнкции соединяются знаками **дизъюнкции**.

Полученная таким образом формула называется **совершенной дизъюнктивной нормальной формой** (**СДНФ**). Для каждой функции такая форма единственна.

**Пример**. Пусть логическая функция задана таблицей истинности (табл. 15)

*Таблица 15*

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

Составим СДНФ:



По таблице истинности можно получить логическую формулу и иначе. Воспользуемся следующим алгоритмом:

1. для каждого набора значений переменных  на котором логи-ческая функция  равна **0**, выписываются **дизъюнкции всех переменных**;
2. над теми переменными, значения которых равны **1**, ставится **отрицание**;
3. все полученные конъюнкции соединяются знаками **конъюнкции**.

Полученная таким образом формула называется **совершенной конъюнктивной нормальной формой** (**СКНФ**). Для каждой функции такая форма единственна.

Для предыдущего примера (табл. 13) составим СКНФ.



При построении СДНФ требуется, чтобы функция была отлична от тождественного нуля. При построении СКНФ требуется, чтобы функция была отлична от тождественной единицы.

**Законы булевой алгебры** (табл. 16).

*Таблица 16*

|  |  |  |  |
| --- | --- | --- | --- |
| № | Дизъюнкция | № | Конъюнкция |
| 1. | Коммутативность  | 1. | Коммутативность  |
| 2. | Ассоциативность  | 2. | Ассоциативность  |
| 3. | Дистрибутивность относительно конъюнкции | 3. | Дистрибутивность относительно дизъюнкции  |
| 4. | Свойство нуля | 4. | Свойство нуля |
| 5. | Закон исключения третьего | 5. | Закон противоречия |
| 6. | Идемпотентность | 6. | Идемпотентность |
| 7. | Свойство единицы | 7. | Свойство единицы |
| 8. | Закон де Моргана | 8. | Закон де Моргана |
| 9. | Закон поглощения | 9. | Закон поглощения |
| 10. | Закон склеивания | 10. | Закон склеивания |
| 11. | Закон двойного отрицания  |

Довольно часто СДНФ и СКНФ, построенные по таблицам истинности, являются очень сложными. Возникает необходимость их минимизации. Число первичных символов (терм) уменьшается. Количество первичных символов, образующих логическую функцию, называется **сложностью** формы *L*(*f*).

Всякая функция реализуется своей **сокращенной ДНФ**. Это дизъюнкция всех значений функции равной **1**.Такая форма единственная. Алгоритм построения **сокращенной ДНФ** **с помощью СКНФ** следующий:

1. по таблице истинности строят СКНФ функции *f* ;

2. в СКНФ раскрывают скобки, удаляют дублирующие элементы () и элементы, имеющие переменные и ее отрицание ();

3. проводится поглощение () и удаление дублирующих элементов.

Для функции предыдущего примера (табл. 13) по СКНФ составим ДНФ.



Раскроем скобки и используем законы булевой алгебры (для простоты опустим знаки дизъюнкции и конъюнкции, заменив их знаками сложения и умножения):













Результатом упрощений является сокращенная ДНФ форма:



Если вынести общий множитель за скобки, можно привести к минимальной СКФ:



Если из дизъюнкций функции нельзя отбросить ни одного слагаемого, то говорят, что получена **тупиковая** ДНФ функции.

Тупиковая ДНФ функции *f* с минимальным числом переменных или их отрицаний является **минимальной ДНФ** функции.

Можно построить **сокращенную ДНФ из СДНФ**. При это используются следующие преобразования:

1. склеивание: 
2. поглощение: 
3. неполное поглощение: 
4. обобщенное поглощение: 

Один из методов получения сокращенной ДНФ известен как метод Блейка-Порецкого, который заключается в неполном попарном склеивании всех элементарных конъюнкций СДНФ между собой и затем в использованием правила поглощения.

**Пример**. Получить сокращенную ДНФ, используя

СДНФ

Для построения **минимальной конъюнктивной нормальной формы** функции нужно построить минимальную ДНФ отрицания функции и в ней произвести замену знаков: дизъюнкции на конъюнкцию и наоборот; каждую переменную заменить ее отрицанием. При этом следует помнить, что двойное отрицание дает саму переменную.

**Пример**. Построить минимальную конъюнктивную форму для функции (табл. 17).

*Таблица 17*

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

Построим СДНФ по таблице для функции 

СДНФ

Построим СКНФ для функции:

СКНФ

Раскроем скобки в этой форме и построим сокращенную ДНФ.

















Построена минимальная ДНФ: 

В формуле заменим знак дизъюнкции на знак конъюнкции и наоборот, и над каждой переменной поставим знак отрицания. Получим минимальную КНФ:



Уменьшить число переменных нельзя. Значит мы получили минимальную нормальную форму функции.

Технической реализацией логической функции является схема, например, **релейно-контактных устройств**. Контактная схема состоит из переключателей , соединяющих проводов и полюсов (вход в схему , выход из схемы). Каждый переключатель может находиться в двух положениях: включено (замкнуто – 1) и выключено (разомкнуто – 0). Эти положения можно описать булевой функций. Если функция равна 1, ток проходит через переключатель, при равенстве функции 0, ток не проходит через переключатель. Схемы таких устройств, реализующих булевы функции, можно представить так:

1. элемент «и» (рис. 13, а);
2. элемент «или» (рис. 13, б);
3. элемент «не» (рис. 13, в).



 а б в

Рис. 13 Элементы логического устройства:

а – элемент «и», б – элемент «или», в – элемент «не»

В общем случае элементы «и» и «или» могут иметь более, чем два входа.

Иногда такие схемы представляют так:

1.  (рис. 14, а);
2.  (рис. 14, б);
3.  (рис. 14, в);
4.  (рис. 14, г).



Рис. 14

Например, для СДНФконтактная схема показана на рис. 15.



Рис. 15

Минимизировать эту схему можно, используя минимальную нормальную форму (рис. 16).



Рис. 16

**Пример**. По схеме (рис. 17) восстановить логическую функцию.



Рис. 17

Ответ: 