Лекция № 4

Т. Ф. Хирьянов

Алгебра логики

Составные высказывания

В логике Аристотеля существуют два состояния: *истина* и *ложь*, которые удобно обозначать соответственно как 1 и 0. Изначально рассматривалось содержание высказываний. Однако впоследствии утвердился подход, в котором существуют атомарные (простые) высказывания, не разбивающиеся на отдельные. Их истинность зависит от сторонних знаний, и в конкретном рассмотрении атомарное выражение принимается истинным либо ложным аксиоматически. Так например, выражение

$$1 + 1 = 1$$

можно считать ложным с точки зрения привычной арифметики либо истинным, если рассматривать алгебру, где это возможно.

Предметом же рассмотрения алгебры логики являются *составные высказывания*. Для их образования из простых высказываний используются *логические операции*. Они выполняются в определенном порядке, соблюдая приоритетность, которая убывает следующим образом.

1. *не* A (логическое отрицание).
   Эта операция имеет самый большой приоритет и применяется к одному высказыванию. На письме отрицание A (не A) записывается одним из двух равноправных способов:

   $$\neg A \qquad \bar{A}$$

2. A *или* B (логическое умножение)
   Такая операция применяется к двум выражениям. Ее запись также имеет альтернативу.

   $$A \wedge B \qquad A \cdot B$$

3. A *или* B (логическое сложение)
   Записывается аналогично.

   $$A \vee B \qquad A + B$$

4. *импликация*
   Эта операция может быть выражена с помощью предшествующих трех, однако из-за частой встречаемости обозначается специальным образом.

   $$A \to B \qquad A \Rightarrow B$$

5. *эквиваленция*
   Является прямой и обратной импликацией.

   $A \leftrightarrow B \qquad A \Leftrightarrow B$

6. *либо* (исключающее или)
   Обозначается так.

   $A \oplus B$

По своей сути эти операторы являются логическими функциями и, как любые функции, отображают множество, называемое областью определения, на множество, являющееся областью значений. Так как область определения в данном случае состоит из двух элементов – правды (1) и лжи (0) – то можно сопоставить каждому элементу соответствующее значение функции (тоже 1 или 0). Получается что для одного переменного существует $2 * 2 = 4$ функций, а для двух аргументов – $(2 * 2) * 2 = 8$ функций. Удобно записать это в виде таблицы. Для введенных ранее операций они будут выглядеть следующим образом.

A	not A
0	1
1	0

A	B	$A \cdot B$	$A + B$	$A \oplus B$	$A \Rightarrow B$	$A \Leftrightarrow B$
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	0	0
1	1	1	1	0	1	1

Так $A \cdot B$ возвращает истину только когда оба аргумента истинны, а $A + B$ наоборот возвращает ложь только когда оба аргумента ложны. Исключающее или $A \oplus B$ похоже на обычное или, однако в отличие от него возвращает ложь в случае истинности обоих аргументов; значение данной операции равно правде только в том случае, если один из аргументов является правдой, а другой обязательно ложью. Эквиваленция верна, когда оба аргумента равны друг другу.

Особого пояснения заслуживает импликация (следствие). Очевидно, что если из истинного утверждения следует истинна, то рассуждения верны, т.е. истинны ($1 \Rightarrow 1 = 1$). Также истинно рассуждение, если из ложного утверждения получается ложное. Например, $2 + 2 = 5$ ложно, откуда следует, что и $4 + 4 = 10$ тоже ложно. Может показаться противоречивым, но из ложного суждения может следовать истинное. Так, $2 + 2 = 5$ ложно. Утверждение $5 = 2 + 2$ также ложно и является справедливым следствием первого утверждения (так как$0 \Rightarrow 0 = 1$). Так как эти утверждения ложны одновременно, то их логическое произведение возвращает ложь$(0 \cdot 0 = 0)$. Однако из этих двух утверждений вместе следует, что $2+2 = 2+2$, а это истинное выражение, т.е. $0 \Rightarrow 1 = 1$. Импликация возвращает ложь только в случае, когда из верного утверждения следует ложное. Действительно, если из утверждения $2 + 2 = 4$ получается, что, например, $3 + 3 = 5$, то рассуждения, очевидно, не верны ($1 \Rightarrow 0 = 0$). Анализируя полученные значения для импликации, можно заметить, что из ложного утверждения может следовать как правда, так и ложь. Именно поэтому в системе аксиом не должно быть противоречивых утверждений, так как тогда следствия из них не всегда будут верны.

Нормальная дизъюнктивная форма

Из указанных функций некоторые выражаются через другие. Из-за этого возникает вопрос, какого количества из этих операций будет достаточно для описания произвольной логической функции любого количества аргументов. Теория показывает, что трех из них (не, и, или) достаточно для этого, нужно только привести функцию к *нормальной дизъюнктивной форме*, т.е. к логической сумме специально сконструированных произведений, равных истине только в одном из случаев, когда функция также истинна. Значит, таких произведений будет столько, сколько раз функция принимает значение 1 в ее таблице истинности.

A	B	C	f(A,B,C)	$\bar{A} \cdot \bar{B} \cdot C$	$\bar{A} \cdot B \cdot C$	$A \cdot B \cdot \bar{C}$
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	0
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	1
1	1	1	0	0	0	0

Тогда нормальная дизъюнктивная форма для приведенной функции будет равна следующей величине.

$$f(A, B, C) = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot B \cdot \bar{C}$$

Из нее видно, что логического отрицания, умножения и сложения достаточно, чтобы представить логическую функцию в виде такой формы.

Законы алгебры логики

Очевидные соотношения:

- $A + 1 = 1$
- $A + 0 = A$
- $A \cdot 1 = A$
- $A \cdot 0 = 0$

Простые законы:

- $\cdot =$ $A + A = A$(законы повторения)
- $A \cdot \bar{A} = 0$ $A + \bar{A} = 1$ (законы «третьего лишнего»)
- $A \cdot (A + B) = A$ $A + A \cdot B = A$ (закон поглощения)
  Действительно, подставив вместо A 1, а затем 0, получаем с помощью очевидных соотношений верные утверждения.

Свойства математических операций.

- $A \cdot B = B \cdot A$ $A + B = B + A$ (коммутативность)
- $A \cdot (B \cdot C) = (A \cdot B) \cdot C = A \cdot B \cdot C$ $A + (B + C) = (A + B) + C = A + B + C$ (ассоциативность)
- $A \cdot (B + C) = A \cdot B + A \cdot C$ $A + B \cdot C = (A + B) \cdot (A + C)$ (дистрибутивность)

Законы отрицания.

- $\bar{\bar{A}} = A$
- $A \bar{+} B = \bar{A} \cdot \bar{B}$
- $A \bar{\cdot} B = \bar{A} + \bar{B}$

Представления операторов.

- $A \Rightarrow B = \bar{A} + B$
- $A \Leftrightarrow B = A \cdot B + \bar{A} \cdot \bar{B}$
- $AB = \bar{A} \cdot B + A \cdot B$

Определение четверти координатной плоскости, которой принадлежит точка

```
x = int(input())
y = int(input())
if y>0:
    if x>0:
        print('1')
    else:
        print('2')
else:
    if x>0:
        print('4')
    else:
        print('3')
```

Каскадная условная конструкция

```
x = int(input())
y = int(input())
if x<0:
    print('negative')
elif x==0:
    print('null')
elif x<1:
    print('0<x<1')
elif x == 1:
    print('1')
else: # x > 1
    print('x > 1')
```

Поиск максимального значения

```
x = int(input())
while x!=0:
    if x > tmp_max: # tmp_max = max(x, tmp_max)
        tmp_max = x
    x = int(input())

x = int(input())
while x!=0:
    if x > m1:
        m1, m2, m3 = x, m1, m2
    elif x > m2:
        m2, m3 = x, m2
    elif x > m3:
        m3 = x
    x int(input())

x = int(input())
```

```
m1 = m2 = m3 = None
while x!=0:
    if m1 == None or x > m1:
        m1, m2, m3 = x, m1, m2
    elif m2 == None or x > m2:
        m2, m3 = x, m2
    elif m3 == None or x > m3:
        m3 = x
    x  int(input())
```

Тест простоты числа

```
def isPrime(x):
    "x is a prime"
    delitel = 2
    while delitel**2 < x:
        if x % delitel == 0:
            return False
        delitel += 1
    return True
```

Разложение числа на множители

```
x = int(input())
delitel = 2
while x > 1:
    if x % delitel == 0:
        print (delitel)
        x //= delitel
    else:
        delitel += 1
```