Лекция 2

Основы алгебры логики

Способы представления чисел в ЭВМ

Целые числа (integer) – представляют собой подмножество бесконечного множества целых чисел, ограниченное максимальным и минимальным значениями.

Различают беззнаковые и знаковые целые числа. Знак числа обычно кодируется старшим битом (1 – отрицательное).

В памяти целое число представляется в виде цепочки битов фиксированного (кратного 8) размера. Диапазон чисел определяется количеством байтов **m**, отводимых под одно число **A**:

- беззнаковое целое: $\mathbf{A_{min} = 0}$, $\mathbf{A_{max} = 2^{8m} - 1}$
- знаковое целое: $\mathbf{A_{min} = -2^{8m-1}}$, $\mathbf{A_{max} = 2^{8m-1} - 1}$

Знаковые числа могут быть представлены в:

➢ прямом коде; ➢ обратном коде; ➢ дополнительном коде

Прямой код – старший разряд (бит) $\mathbf{n}$ объявляется знаковым (0 – если $\mathbf{A} \geq \mathbf{0}$, иначе – 1), в остальных (цифровых) $\mathbf{n-1}$ разрядах записывается двоичное представление модуля числа $\mathbf{A}$

$$[\mathbf{A}]_{\text{пр}} = \begin{cases} \mathbf{A}, & \mathbf{A} \geq 0 \\ 2^{\mathbf{n}} + |\mathbf{A}|, & \mathbf{A} < 0 \end{cases}$$

Обратный код – положительные числа записываются в прямом коде, а отрицательные образуются простой инверсией всех $\mathbf{n-1}$ бит модуля числа $\mathbf{A}$

$$[\mathbf{A}]_{\text{обр}} = \begin{cases} \mathbf{A}, & \mathbf{A} \geq 0 \\ 2^{\mathbf{n}} + \overline{|\mathbf{A}|}, & \mathbf{A} < 0 \end{cases}$$

Дополнительный код – получается из обратного добавлением единицы $2^{\mathbf{n}} + |\mathbf{A}| + 1 = 2^{\mathbf{n}} - |\mathbf{A}|$ является дополнением модуля числа $\mathbf{A}$ до нуля $2^{\mathbf{n}} - |\mathbf{A}| + |\mathbf{A}| = 0$, поскольку $2^{\mathbf{n}} \equiv 0$

$$[\mathbf{A}]_{\text{доп}} = \begin{cases} \mathbf{A}, & \mathbf{A} \geq 0 \\ 2^{\mathbf{n}} - |\mathbf{A}|, & \mathbf{A} < 0 \end{cases}$$

Арифметические операции в двоичной системе счисления

Основные правила

- ***Сложение***

$$0 + 0 = 0,\ 0 + 1 = 1,\ 1 + 0 = 1,\ 1 + 1 = 2_{10} = 10_2$$

(единица идет в старший разряд)

- ***Вычитание***

$$0 - 0 = 0,\ 1 - 0 = 1,\ 1 - 1 = 0,\ 0 - 1 = 10 = 2_{10} - 1 = 1$$

(единицу забираем у старшего разряда)

- ***Умножение***

$$0 \times 0 = 0,\ 0 \times 1 = 0,\ 1 \times 0 = 0,\ 1 \times 1 = 1$$

- ***Деление***

0 : 0 = не определено, 1 : 0 = не определено, 0 : 1 = 0, 1 : 1 = 1

$$+\begin{array}{r}0110110\\0010101\\\hline 1001011\end{array}\qquad -\begin{array}{r}110110\\010101\\\hline 100001\end{array}\qquad \times\begin{array}{r}110110\\10\\\hline 1101100\end{array}\qquad :\begin{array}{r}110110\\10\\\hline 011011\end{array}$$

Представления целых чисел в различных кодах

Десятичное представление	Двоичное представление (8 бит)		
	прямой	обратный	дополнительный
127	0111 1111	0111 1111	0111 1111
1	0000 0001	0000 0001	0000 0001
0	0000 0000	0000 0000	0000 0000
-0	1000 0000	1111 1111	---
-1	1000 0001	1111 1110	1111 1111
-127	1111 1111	1000 0000	1000 0001
-128	---	---	1000 0000

Выполнение арифметических операций в прямом коде затруднено, поскольку требует особой обработки знакового разряда и «отрицательного нуля», что усложняет реализацию процессора.

Обратный код позволяет сделать операции сложения и вычитания одинаковыми для знаковых и беззнаковых чисел, но также имеет два кода числа нуль.

Дополнительный код наиболее удобен, т.к. устраняет двойственность нуля обратного кода, сохраняя все его достоинства.

Вещественные числа (float) – имеют две формы представления в ЭВМ:

- ***с фиксированной запятой*** (точкой) (fixed-point number) – формат представления вещественного числа $\mathbf{A}$ в памяти ЭВМ в виде целого числа $\mathbf{A'}$ разрядностью $\mathbf{n}$ бит

$$\mathbf{A = A' z},$$

  где вес младшего разряда $\mathbf{z}$ обычно выбирают равным $\mathbf{z = 2^{-f}}$, что соответствует $\mathbf{f}$ битам на дробную часть и $\mathbf{i = n - f}$ – на целую, и обозначают как «i,f»

- ***с плавающей запятой*** (floating-point number) – форма представления вещественного числа $\mathbf{A}$ в виде мантиссы $\mathbf{m}$ и показателя степени $\mathbf{k}$ основания системы счисления $\mathbf{q}$

$$\mathbf{A = m\, q^k}, \text{ где } \mathbf{q^{-1} \le |m| < 1}$$

Предложена в 1937 г. К. Цузе для увеличения диапазона представляемых вещественных чисел и повышения точности

Определение алгебры логики

Основоположники логики

Аристотель (384-322 до н.э.)

Основоположник формальной логики (понятие, суждение, умозаключение)

Джордж Буль (1815-1864)

Создатель новой области науки – Математической логики (Булевой алгебры или Алгебры высказываний)

Алгебра – наука об общих операциях, аналогичных сложению и умножению, которые могут выполняться над математическими объектами – числами, множествами и т.п.

Клод Шеннон (1916-2001)

Автор исследований, позволивших применить алгебру логики в вычислительной технике

Определения алгебры логики

Алгебра логики определяет правила записи, вычисления значений, упрощения и преобразования **высказываний**

Высказывание – это предложение на любом языке, содержание которого можно однозначно определить как **истинное** или **ложное**

В алгебре логики высказывания обозначаются буквами и называются **логическими переменными**, которые могут принимать **логические значения 0** и **1**

Если высказывание истинно, то значение логической переменной обозначают единицей (**A = 1**), если ложно – нулем (**B = 0**)

Высказывания подразделяются на:

- ***простые*** – никакая их часть сама не является высказыванием
- ***сложные*** (составные) – строятся из простых высказываний с помощью **логических функций**

Логическая функция (булева функция, функция алгебры логики – **ФАЛ**) – функция от **n** логических переменных, принимающая одно из логических значений **0** или **1**. Неотрицательное целое число **n** называют **арностью** функции. В случае **n = 0** функция превращается в **булеву константу**

Булева функция арности **n** полностью определяется заданием своих значений на своей области определения, т.е. на всех **булевых векторах** (наборах входных значений) длины **n**. Число таких векторов $\mathbf{2^n}$, а число возможных **n**–арных булевых функций $2^{2^{\mathbf{n}}}$

Булева функция задаётся конечным набором значений, что позволяет представить её в виде **таблицы истинности**

x_1	x_2	…	x_{n-1}	x_n	$f(x_1, x_2,…,x_n)$
0	0	…	0	0	0
…	…	…	…	…	…
1	1	…	1	1	0

Если значение логической функции не зависит от одной из переменных, такая переменная называется **фиктивной**, в противном случае переменная называется **существенной**

Свойство суперпозиции – результат вычисления булевой функции может быть использован в качестве одного из аргументов другой функции. Результат **суперпозиции** можно рассматривать как новую булеву функцию со своей таблицей истинности

Свойство тождественности – две булевы функции **f** и **g** тождественны друг другу, если на любых одинаковых наборах существенных аргументов принимают равные значения

$$\mathbf{f}\left(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_\mathbf{n}\right)=\mathbf{g}\left(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_\mathbf{n}\right)$$

Свойство двойственности – две функции **f** и **g** называются двойственными, если

$$\mathbf{f}\left(\overline{\mathbf{x}_1},\overline{\mathbf{x}_2},\ldots,\overline{\mathbf{x}_\mathbf{n}}\right)=\overline{\mathbf{g}\left(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_\mathbf{n}\right)}$$

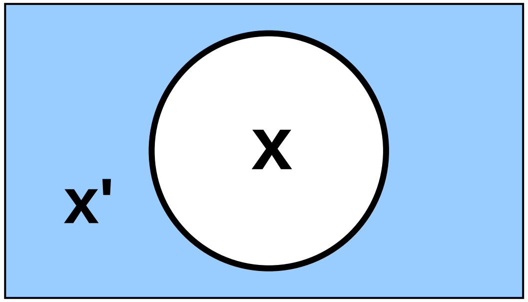
Если в булевом тождестве заменить каждую функцию на двойственную ей, снова получится верное тождество

Базовые функции алгебры логики

Инверсия (логическое отрицание) – функция одного аргумента, которая возвращает противоположное ему значение

$$f(x) = \overline{x} = \neg x = x' = NOT(x) = НЕ(x)$$

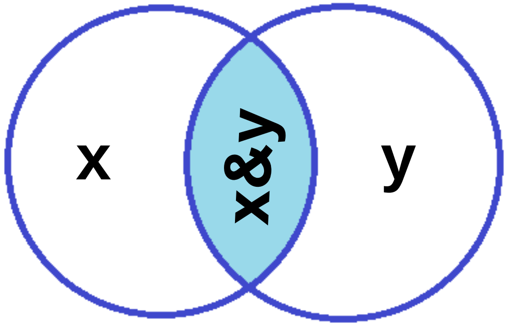
x	x'
0	1
1	0



Конъюнкция (логическое умножение) – логическая функция двух аргументов, которая возвращает истинное значение тогда и только тогда, когда оба исходных высказывания истинны

$$f(x,y) = x \wedge y = x \cdot y = x \,\&\, y = AND(x,y) = И(x,y)$$

x	y	x & y
0	0	0
0	1	0
1	0	0
1	1	1

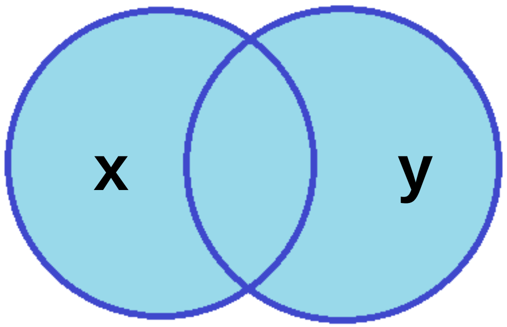


Базовые функции алгебры логики

Дизъюнкция (логическое сложение) – логическая функция двух аргументов, которая возвращает ложное значение тогда и только тогда, когда оба исходных высказывания ложны

$$\mathbf{f}\left(\mathbf{x},\mathbf{y}\right)=\mathbf{x}\vee\mathbf{y}=\mathbf{x}+\mathbf{y}=\mathbf{OR}\left(\mathbf{x},\mathbf{y}\right)=\mathbf{ИЛИ}\left(\mathbf{x},\mathbf{y}\right)$$

x	y	x v y
0	0	0
0	1	1
1	0	1
1	1	1



x v y

При суперпозиции логических функций порядок их применения определяется в соответствии со следующим **приоритетом**: вначале операции инверсии, затем конъюнкции и в последнюю очередь – дизъюнкции. Приоритет можно изменить применением скобок – выражение в скобках рассчитывается в первую очередь

Основные законы алгебры логики

- ***Аксиомы***

x = 0, если **x ≠ 1**; **x = 1**, если **x ≠ 0**

$$\bar{0}=1;\ \bar{1}=0;\ \mathbf{x}\cdot 0=0;\ \mathbf{x}\vee 1=1$$

- ***Законы коммутативности***

$$\mathbf{x\cdot y=y\cdot x;\quad x\vee y=y\vee x}$$

- ***Законы ассоциативности***

$$\mathbf{(x\cdot y)\cdot z=x\cdot(y\cdot z);\quad (x\vee y)\vee z=x\vee(y\vee z)}$$

- ***Законы поглощения (нуля и единицы)***

$$\mathbf{x}\vee 0=\mathbf{x};\quad 1\cdot\mathbf{x}=\mathbf{x}$$

- ***Законы дистрибутивности***

$$\mathbf{x\cdot(y\vee z)=x\cdot y\vee x\cdot z;\quad x\vee y\cdot z=(x\vee y)\cdot(x\vee z)}$$

- ***Закон противоречия***

$$\mathbf{x}\cdot\bar{\mathbf{x}}=0$$

Основные законы алгебры логики

- ***Закон исключения третьего***

$$\mathbf{x} \vee \overline{\mathbf{x}} = 1$$

- ***Законы идемпотентности***

$$\mathbf{x} \cdot \mathbf{x} = \mathbf{x} \vee \mathbf{x} = \mathbf{x}$$

- ***Закон двойного отрицания***

$$\overline{\overline{\mathbf{x}}} = \mathbf{x};$$

- ***Законы де Моргана***

$$\overline{\mathbf{x} \cdot \mathbf{y}} = \overline{\mathbf{y}} \vee \overline{\mathbf{x}}; \qquad \overline{\mathbf{y}} \cdot \overline{\mathbf{x}} = \overline{\mathbf{x} \vee \mathbf{y}}$$

- ***Законы поглощения***

$$\mathbf{x} \vee \left(\mathbf{x} \cdot \mathbf{y}\right) = \mathbf{x}; \qquad \mathbf{x} \cdot \left(\mathbf{x} \vee \mathbf{y}\right) = \mathbf{x}$$

Любой из перечисленных законов доказывается при помощи таблицы истинности

Канонические формы логических формул

Каждую булеву функцию можно записать бесконечным множеством представляющих ее логических формул. Одна из основных задач алгебры логики – нахождение наиболее простых канонических форм формул, представляющих булевы функции

Если логическая функция выражена через дизъюнкцию, конъюнкцию и отрицание переменных, то такая форма представления называется **нормальной**

Нормальные формы, в которых функции записываются единственным образом, называются **совершенными**

Особую роль в алгебре логики играют классы дизъюнктивных и конъюнктивных совершенных нормальных форм, основанных на понятиях **элементарных дизъюнкции** и **конъюнкции**

Э***лементарной (диз-)конъюнкцией*** называется формула, в которой (диз-)конъюнкция осуществляется над одной или несколькими переменными, взятыми с отрицанием или без

Формула называется **дизъюнктивной нормальной формой (ДНФ)**, если она является дизъюнкцией неповторяющихся элементарных конъюнкций

Формула **A** от **k** переменных называется **совершенной дизъюнктивной нормальной формой (СДНФ)**, если:

1) **A** является ДНФ, в которой каждая элементарная конъюнкция есть конъюнкция **k** переменных, причем на **i**-м месте этой конъюнкции стоит либо переменная $\mathbf{x_i}$, либо ее отрицание

2) Все элементарные конъюнкции в такой ДНФ попарно различны

Аналогичным образом вводятся определения **конъюнктивной нормальной формы (КНФ)** и **совершенной конъюнктивной нормальной формы (СКНФ)** только операции конъюнкции дизъюнкции в ней меняются местами