Математические основы информатики

Логика и бинарная арифметика.

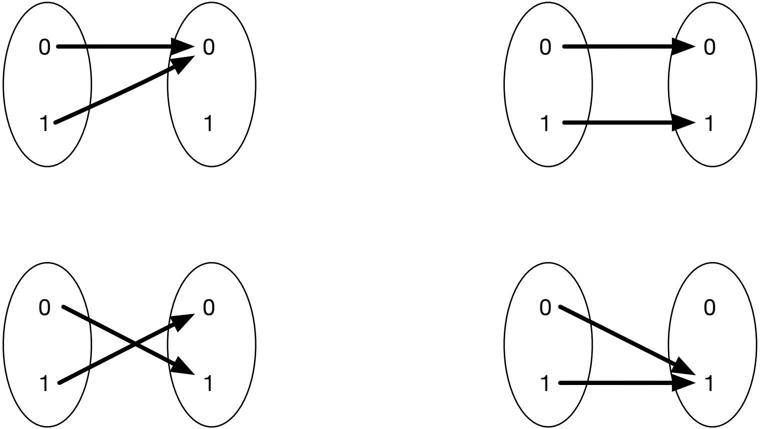
Сергей Леонидович Бабичев

Почему нас интересует логика?

- Доказывать корректность алгоритмов можно с помощью аппарата математической логики: если верно высказывание A и верны все преобразования, то будет верно и высказывание B, которое есть следствие этих преобразований.
- Представление чисел в современных компьютерах в подавляющем количестве случаев двоичное. Знание этого представления позволяет делать невероятные трюки, ускоряющие исполнение некоторых алгоритмов в десятки раз. Работа с битами в представлении числа — интересная математическая задача

Логические операции и их операнды

- Сколько имеется функций от одного логического аргумента?
- *Аргумент* (элемент множества D) может быть 0 или 1, а *значение* функции — тоже либо 0, либо 1.
- Нужна функция $\to$ образ прообраза единственен.



Все возможные логические функции от одной переменной.

Логические функции от одного аргумента

- Упорядочим все такие возможные функции по их значениям.

Функция 00	
x	f(x)
0	0
1	0

Функция 01	
x	f(x)
0	0
1	1

Функция 10	
x	f(x)
0	1
1	0

Функция 00	
x	f(x)
0	1
1	1

- Функция 00 — функция *тривиального нуля*.
- Функция 01 — тривиальная функция *тавтологии*.
- Единственная нетривиальная функция — функция 10. Функция логического отрицания, not, $\bar{x}$.
- функция 11 — функция *тривиальной единицы*, или *тривиальной истины*.

Функции от двух аргументов

- Функции от кортежей-пар.
- Множество кортежей-аргументов содержит $2^2 = 4$ элемента.
- Множество различных функций $= 2^{2^2} = 16$.
- Для функции с любым количеством аргументов строится *таблица истинности* и она получает номер по последнему столбцу.

Функция 0110		
x	y	f(x,y)
0	0	0
0	1	1
1	0	1
1	1	0

Пример одной из функций.

Другое представление функций с двумя аргументами

Функция 0000		
	0	1
0	0	0
1	0	0

Функция 0001		
	0	1
0	0	0
1	0	1

Функция 0010		
	0	1
0	0	0
1	1	0

Функция 0011		
	0	1
0	0	0
1	1	1

Функция 0100		
	0	1
0	0	1
1	0	0

Функция 0101		
	0	1
0	0	1
1	0	1

Функция 0110		
	0	1
0	0	1
1	1	0

Функция 0111		
	0	1
0	0	1
1	1	1

Функция 1000		
	0	1
0	1	0
1	0	0

Функция 1001		
	0	1
0	1	0
1	0	1

Функция 1010		
	0	1
0	1	0
1	1	0

Функция 1011		
	0	1
0	1	0
1	1	1

Функция 1100		
	0	1
0	1	1
1	0	0

Функция 1101		
	0	1
0	1	1
1	0	1

Функция 1110		
	0	1
0	1	1
1	1	0

Функция 1111		
	0	1
0	1	1
1	1	1

Классифицируем функции по номерам

- Ряд архивных (VAX-11) и современных (IBM Z390) компьютеров имеют машинные команды логических операций с тремя операндами.
- Два операнда — аргументы операции.
- Третий операнд — число от 0 до 15.
- Результат — побитовая заданная операция над операндами.

Наблюдения над таблицей

- Применение not к первому аргументу — меняются местами строки.
- Применение not к второму аргументу — меняются местами столбцы.
- Применение not к таблице — инвертируются все значения.

0000 — тривиальный ноль, 1111 — тривиальная единица

Функция 0000		
	0	1
0	0	0
1	0	0

Функция 0001		
	0	1
0	0	0
1	0	1

Функция 0010		
	0	1
0	0	0
1	1	0

Функция 0011		
	0	1
0	0	0
1	1	1

Функция 0100		
	0	1
0	0	1
1	0	0

Функция 0101		
	0	1
0	0	1
1	0	1

Функция 0110		
	0	1
0	0	1
1	1	0

Функция 0111		
	0	1
0	0	1
1	1	1

Функция 1000		
	0	1
0	1	0
1	0	0

Функция 1001		
	0	1
0	1	0
1	0	1

Функция 1010		
	0	1
0	1	0
1	1	0

Функция 1011		
	0	1
0	1	0
1	1	1

Функция 1100		
	0	1
0	1	1
1	0	0

Функция 1101		
	0	1
0	1	1
1	0	1

Функция 1110		
	0	1
0	1	1
1	1	0

Функция 1111		
	0	1
0	1	1
1	1	1

Функции, зависящие ровно от одного аргумента

Функция 0000		
	0	1
0	0	0
1	0	0

Функция 0001		
	0	1
0	0	0
1	0	1

Функция 0010		
	0	1
0	0	0
1	1	0

Функция 0011		
	0	1
0	0	0
1	1	1

Функция 0100		
	0	1
0	0	1
1	0	0

Функция 0101		
	0	1
0	0	1
1	0	1

Функция 0110		
	0	1
0	0	1
1	1	0

Функция 0111		
	0	1
0	0	1
1	1	1

Функция 1000		
	0	1
0	1	0
1	0	0

Функция 1001		
	0	1
0	1	0
1	0	1

Функция 1010		
	0	1
0	1	0
1	1	0

Функция 1011		
	0	1
0	1	0
1	1	1

Функция 1100		
	0	1
0	1	1
1	0	0

Функция 1101		
	0	1
0	1	1
1	0	1

Функция 1110		
	0	1
0	1	1
1	1	0

Функция 1111		
	0	1
0	1	1
1	1	1

Функции ровно с одной единицей

Функция 0000		
	0	1
0	0	0
1	0	0

Функция 0001		
	0	1
0	0	0
1	0	1

Функция 0010		
	0	1
0	0	0
1	1	0

Функция 0011		
	0	1
0	0	0
1	1	1

Функция 0100		
	0	1
0	0	1
1	0	0

Функция 0101		
	0	1
0	0	1
1	0	1

Функция 0110		
	0	1
0	0	1
1	1	0

Функция 0111		
	0	1
0	0	1
1	1	1

Функция 1000		
	0	1
0	1	0
1	0	0

Функция 1001		
	0	1
0	1	0
1	0	1

Функция 1010		
	0	1
0	1	0
1	1	0

Функция 1011		
	0	1
0	1	0
1	1	1

Функция 1100		
	0	1
0	1	1
1	0	0

Функция 1101		
	0	1
0	1	1
1	0	1

Функция 1110		
	0	1
0	1	1
1	1	0

Функция 1111		
	0	1
0	1	1
1	1	1

Функции с тремя единицами — отрицание предыдущих

Функция 0000		
	0	1
0	0	0
1	0	0

Функция 0001		
	0	1
0	0	0
1	0	1

Функция 0010		
	0	1
0	0	0
1	1	0

Функция 0011		
	0	1
0	0	0
1	1	1

Функция 0100		
	0	1
0	0	1
1	0	0

Функция 0101		
	0	1
0	0	1
1	0	1

Функция 0110		
	0	1
0	0	1
1	1	0

Функция 0111		
	0	1
0	0	1
1	1	1

Функция 1000		
	0	1
0	1	0
1	0	0

Функция 1001		
	0	1
0	1	0
1	0	1

Функция 1010		
	0	1
0	1	0
1	1	0

Функция 1011		
	0	1
0	1	0
1	1	1

Функция 1100		
	0	1
0	1	1
1	0	0

Функция 1101		
	0	1
0	1	1
1	0	1

Функция 1110		
	0	1
0	1	1
1	1	0

Функция 1111		
	0	1
0	1	1
1	1	1

Все функции в одной таблице

0000: $\bot$

0001: $x \wedge y$

0010: $x \wedge \overline{y}$

0011: x

0100: $\overline{x} \wedge y$

0101: y

0110: $x \oplus y$, $x \not\equiv y$.

0111: $x \vee y$

1000: $\overline{x \vee y}$, $x \downarrow y$

1001: $x \equiv y$, $x \Leftrightarrow y$

1010: $\overline{y}$

1011: $x \vee \overline{y}$

1100: $\overline{x}$

1101: $\overline{x} \vee y$, $x \Rightarrow y$

1110: $\overline{x \wedge y}$, $x \overline{\wedge} y$, $x'y$, $x|y$

1111: $\top$

Запись логических функций

Мы можем задать одну и ту же логическую функцию несколькими способами.

1. С помощью таблицы истинности.
2. С помощью комбинации логических функций. Мы уже убедились, что одна и та же функция может быть записана по-разному.
3. Векторный. $f(x, y) = (0001)$. Элементы вектора соответствуют аргументам, расположенным в *лексикографическом* порядке.

Суперпозиция функций.

Задача. По функциям $f(x_1, x_2) = (1011)$ и $g(x_3, x_4) = (1001)$ построить векторный вид функции $h(x_2, x_3, x_4) = f(g(x_3, x_4), x_2)$.

Решение. *Суперпозиция*: функция g — один из входных аргументов функции f.

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

x_3	x_4	$g(x_3, x_4)$
0	0	1
0	1	0
1	0	0
1	1	1

Решение задачи

- Как строится суперпозиция?
- 6-я строка. $g(0,1) = 0$ f: $f(g(0,1),1) = f(0,1) = 0$.
- Так проделываем для всех строк таблицы. Итог: (11111001).

x_2	x_3	x_4	$g(x_3, x_4)$	$f(g(x_3, x_4), x_2)$
0	0	0	1	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	1
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Тождественность функций. Тавтологии.

Задача. Тождественны ли функции

$$f_1(x, y, z) = (x \to y) \vee (z \equiv (\overline{x} \vee y)) \tag{1}$$

и

$$f_2(x, y, z) = \overline{x \wedge \overline{y} \wedge z}. \tag{2}$$

Решение задачи. Первый способ.

Строим таблицы истинности для обеих функций.

x	y	z	$x \to y$	$\overline{x} \vee y$	$z \equiv (\overline{x} \vee y)$	f_1	f_2
0	0	0	1	1	0	1	1
0	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1
0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1
1	0	1	0	0	0	0	0
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

Решение задачи. Второй способ.

- Найдём все значения функции f_1, в которых она обращается в 0.
- Чтобы дизъюнкция двух переменных обратилась в ноль, необходимо, чтобы оба аргумента были нулями.

$$x \to y = 0 \tag{3}$$

$$z \equiv (\overline{x} \vee y) = 0 \tag{4}$$

- Импликация даст 0 только в случае входного вектора (1,0).
- Следовательно, функция f_1 обращается в 0 только в точке $(1, 0, 1)$. На всех остальных наборах функция равна 1.
- Рассмотрим вторую функцию. Конъюнкция трёх величин равна 1 тогда и только тогда, когда каждая величина равна 1. Отсюда решение: $(1, 0, 1)$.
- Вывод: функции равны.

Существенная зависимость от переменных

Определение: Функция $f(x_1, \ldots, x_n)$ *существенно зависит* от x_i, если существует некоторый набор величин $\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha_n)$, $a_i \in \{0, 1\}$, такой, что

$$f(\alpha_1, \ldots, \alpha_{i-1}, 0, \alpha_{i+1}, \ldots, \alpha n) \neq f(\alpha_1, \ldots, \alpha_{i-1}, 1, \alpha_{i+1}, \ldots, \alpha_n). \quad (5)$$

- Оставим из функций нетривиальные:
  $\wedge = 0001$, $\vee = 0111$, $\rightarrow = 1101$, $\equiv = 1001$, $\oplus = 0110$, $| = 1110$, $\downarrow = 1000$.
  Формулы отрицания:
  - $\overline{x \wedge y} = \overline{x} \vee \overline{y}$
  - $\overline{x \vee y} = \overline{x} \wedge \overline{y}$
  - $\overline{x \oplus y} = x \equiv y$
  - $\overline{x \equiv y} = x \oplus y$
  - $\overline{x \rightarrow y} = x \wedge \overline{y}$
  - $\overline{x|y} = x \wedge y$
  - $\overline{x \downarrow y} = x \vee y$
  - $\overline{\overline{x}} = x$

Коммутативность и ассоциативность

Определение: Для бинарной функции $\circ$ имеет место *коммутативность*, если для любых $x, y \in \{0, 1\}$ выполняется:

$$x \circ y = y \circ x.$$

Например, функции $\{\wedge, \vee, \oplus, \equiv, |, \downarrow\}$ являются коммутативными.

Определение: Для бинарной функции $\circ$ имеет место *ассоциативность*, если для любых $x, y, z \in \{0, 1\}$ выполняется:

$$x \circ (y \circ z) = (x \circ y) \circ z.$$

Например, функции $\{\wedge, \vee, \oplus, \equiv\}$ являются ассоциативными.

Дистрибутивность

Определение: Для бинарной функций $\circ$ имеет место *дистрибутивность* относительно функции $\diamond$, если для любых $x, y, z \in \{0, 1\}$ выполняется:

$$x \circ (y \diamond z) = (x \circ y) \diamond (z \circ z).$$

Некоторые из пар, для которых выполняется дистрибутивность:
$\{\wedge, \vee\}; \{\vee, \wedge\}; \{\rightarrow, \rightarrow\}, \{\wedge, \oplus\}$

Стрелка Пирса и штрих Шеффера.

Они интересны тем, что через них можно выразить все остальные, включая одноместную.

Задача. Выразить все элементарные ФАЛ через штрих Шеффера.
Решение:

- Операция отрицания.

$$\overline{x} = \overline{x \& x} = x|x$$

- Операция конъюнкции:

$$x \& y = \overline{\overline{x \& y}} = \overline{x|y} = (x|y)|(x|y)$$

Здесь мы воспользовались формулой двойного отрицания, формулой для отрицания дизъюнкции и уже выведенной формулой для выражения отрицания через штрих Шеффера.

Выражение операций через штрих Шеффера

- Операция дизъюнкции.

$$x \vee y = \overline{\overline{x \vee y}} = \overline{\overline{x}\&\overline{y}} = \overline{x}|\overline{y} = (x|x)|(y|y)$$

- Операция импликации.

$$x \to y = \overline{\overline{\overline{x} \vee y}} = \overline{x\&\overline{y}} = x|\overline{y} = x|(y|y)$$

СКНФ и СДНФ

Введём обозначение:

$$x^\sigma = \begin{cases} x, & \text{если } \sigma = 1, \\ \overline{x}, & \text{если } \sigma = 0. \end{cases}$$

Для любой функции имеет место разложение СДНФ:

$$f(x_1, \dots, x_n) = \bigvee_{\substack{(\sigma_1,\dots,\sigma_n) \\ f(\sigma_1,\dots,\sigma_n)=1}} x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n}.$$

Для любой функции имеет место разложение СКНФ:

$$f(x_1, \dots, x_n) = \bigwedge_{\substack{(\sigma_1,\dots,\sigma_n) \\ f(\sigma_1,\dots,\sigma_n)=0}} x_1^{\overline{\sigma}_1} \vee \dots \vee x_n^{\overline{\sigma}_n}.$$

Задача на разложения

Задача Построить СКНФ и СДНФ для функции с вектором (0110).

- СДНФ. Наборы, при которых функция равна 1 — $(0, 1)$ и $(1, 0)$.

$$f(x, y) = (x^0 \wedge y^1) \vee (x^1 \wedge y^0) = \overline{x} \wedge y \vee x \wedge \overline{y}. \tag{6}$$

- СКНФ. Нужны наборы, при которых функция равна 0. Это $(0, 0)$ и $(1, 1)$.

$$f(x, y) = (x^{\overline{0}} \vee y^{\overline{0}}) \wedge (x^{\overline{1}} \vee y^{\overline{1}}) = (x \vee y) \wedge (\overline{x} \vee \overline{y}). \tag{7}$$

Логика и числа.

Числа и их представление

- Для процессора числа представляются набором логических значений на множестве их n элементов.
- Мощность такого множества 2^n.
- Значение числа определяется соглашениями о кодировании.
- Кодирование чисел — биективное отображение множества чисел мощностью 2^n на множество кортежей $\{0, 1\}^n$.
- Имеются различные способы кодирования множества чисел.
- $\mathbb{N}_{2^n} = S_u = [0 \ldots 2^n)$

Знаковое кодирование

- Для кодирования чисел, имеющих знак имеются варианты:
  - Кодировать знак в одной из позиций. Остальные позиции кодируют абсолютную величину. *Прямой код.*
  - Все позиции кодируются инверсией. *Обратный код.*
  - Все позиции кодируются операцией вычитания числа из нуля. *Дополнительный код.*

Примеры знакового кодирования

Для $n = 8$:

x	10	-10	0
Прямой код	00001010	10001010	00000000 или 10000000
Обратный код	00001010	11110101	00000000 или 11111111
Дополнительный код	00001010	11110100	00000000

Перевод числа в дополнительный код

- Процессор оперирует числами в булеане $\{0, 1\}^n$.
- И операнды и результат есть элементы этого множества.
- При логических операциях конъюнкции, дизъюнкции, исключающего или отрицания размер результата совпадает с размеров операндов.
- При логических операциях сдвига и арифметических операциях результат может содержать более n разрядов.
- Все лишние разряды отбрасываются.
- В дальнейшем будем проводить действия в булеане $\{0, 1\}^n$, отображаемом на множество $\mathbb{N}_{2^n}$.

Реализация арифметических операций

- Пусть имеются логические элементы:
  - not с одним входом и одним выходом.
  - and с двумя входами и одним выходом.
  - or с двумя входами и одним выходом.

Определить операции сложения двух чисел при $n = 1$ (два раздельных входа) с получением результата $n = 2$ (два раздельных выхода).

> Абсолютно все операции на компьютерах производятся на логических элементах.

Схема задачи.

Схема логических действий

x	y	r_1	r_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Схема логических действий

x	y	r_1	r_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$r_1 = x \wedge y$$

Коммутационная схема 1

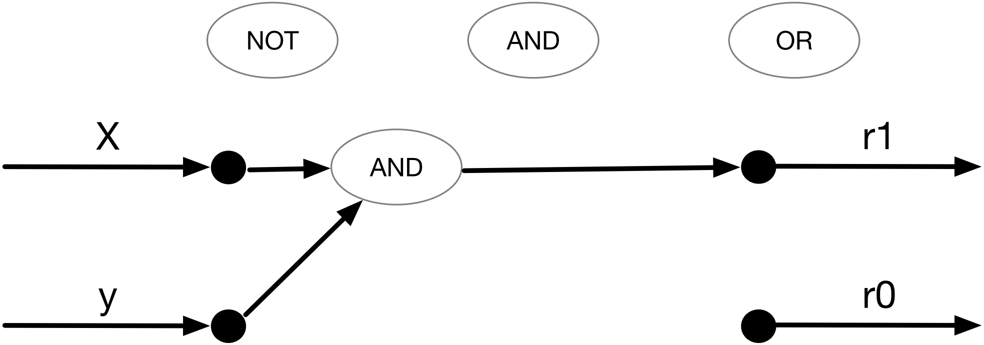


Схема логических действий

x	y	r_1	r_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$r_1 = x \wedge y$$

$$r_0 = x \oplus y = x \vee y \wedge \overline{x \wedge y}$$

Коммутационная схема 2