ФАКУЛЬТЕТ
ВЫЧИСЛИТЕЛЬНОЙ
МАТЕМАТИКИ И
КИБЕРНЕТИКИ
МГУ ИМЕНИ
М.В. ЛОМОНОСОВА

teach-in
ЛЕКЦИИ УЧЕНЫХ МГУ

ДИСКРЕТНАЯ МАТЕМАТИКА

АЛЕКСЕЕВ
ВАЛЕРИЙ БОРИСОВИЧ

ВМК МГУ

КОНСПЕКТ ПОДГОТОВЛЕН СТУДЕНТАМИ, НЕ ПРОХОДИЛ ПРОФ. РЕДАКТУРУ И МОЖЕТ СОДЕРЖАТЬ ОШИБКИ. СЛЕДИТЕ ЗА ОБНОВЛЕНИЯМИ НА VK.COM/TEACHINMSU.

ЕСЛИ ВЫ ОБНАРУЖИЛИ ОШИБКИ ИЛИ ОПЕЧАТКИ, ТО СООБЩИТЕ ОБ ЭТОМ, НАПИСАВ СООБЩЕСТВУ VK.COM/TEACHINMSU.

БЛАГОДАРИМ ЗА ПОДГОТОВКУ КОНСПЕКТА
СТУДЕНТКУ ФАКУЛЬТЕТА ВМК МГУ
ЧЕРНИКОВУ ПОЛИНУ ГЕОРГИЕВНУ

Оглавление

Лекция 1. Функции алгебры логики.

Функции алгебры логики

При работе с функциями алгебры логики (булевских функций) будем рассматривать основное множество $E_2 = \{0,1\}$. На этом множестве функции будут задаваться и принимать значения булевские функции.

$$E_2^n = \{(\alpha_1, \alpha_2, \dots, \alpha_n) \mid \forall i\ (\alpha_i \in \{0,1\})\}$$

Опр. Функцией алгебры логики (булевой функции) называется любое отображение: $f(x_1, x_2, \dots, x_n)$: $E_2^n \to E_2$.

Рассмотрим, как задаются булевы функции.

Пусть $n = 1$. Тогда для $f(x)$можно перечислить всевозможные значения и указать значения функции:

$\boldsymbol{x}$	**0**	**1**	$\boldsymbol{x}$	$\boldsymbol{\bar{x}}$
0	0	1	0	1
1	0	1	1	0

0 и 1 − функции – константы

x − тождественная функция

$\bar{x}$ − отрицание

Других функций для $n = 1$ нет.

Функции для двух переменный, то есть функции при $n = 2$ тоже можно задать таблицей. Двоичные наборы – значения переменных можно рассматривать как двоичное представление натуральных чисел. Так для $n = 2$ получим 4 набора 00, 01, 10, 11, которые соответствуют двоичному представлению натуральных чисел 0, 1, 2, 3.

x_1	x_2	$x_1 \& x_2$	$x_1 \vee x_2$	$x_1 \oplus x_2$	$x_1 \sim x_2$	$x_1 \to x_2$	$x_1 \mid x_2$	$x_1 \downarrow x_2$
0	0	0	0	0	1	1	1	1
0	1	0	1	1	0	1	1	0
1	0	0	0	1	0	0	1	0
1	1	1	1	0	1	1	0	0

$x_1 \& x_2 = x_1 \cdot x_2 = x_1 x_2$ − конъюнкция

$x_1 \vee x_2$ − дизъюнкция

$x_1 \oplus x_2 = x_1 + x_2$ – сумма по модулю 2

$x_1 \sim x_2 = \overline{x_1 \oplus x_2}$ – эквивалентность

$x_1 \to x_2$ – импликация

$x_1 | x_2 = \overline{x_1 x_2}$ – штрих Шеффера

$x_1 \downarrow x_2 = \overline{x_1 \vee x_2}$ – стрелка Пирса

***Лемма* (о числе слов).**

В алфавите из *r* букв можно построить r^m различных слов длины *m*.

Доказательство.

Проведем индукцию по *m*.

1) $m = 1$: $r^m = r^1 = r$ слов, очевидно.
2) Пусть утверждение леммы верно для $m = k - 1$, то есть слов длины $k - 1$ всего r^{k-1}. Для каждого такого слова длины $k - 1$ существует ровно r возможностей добавить одну букву в конец и получить слово длины k. Таким образом, из одного слова длины $k - 1$ получается r слов длины k. Следовательно, всего слов $r \cdot r^{k-1} = r^k$ ■.

Теперь зададим таблицу для функции n переменных. Из леммы о числе слов следует, что всего двоичных наборов для *n* переменных будет 2^n.

Утверждение. Всего существует 2^{2^n} функций алгебры логики от *n* фиксированных переменных.

Доказательство. Воспользуемся леммой о количестве слов: $r = 2$, так как $E_2 = \{0,1\}$, а длина слова $\alpha_0 \alpha_1 \dots \alpha_{2^n-1}$ (значения функции) 2^n. Соответственно всего 2^{2^n} функций алгебры логики от *n* фиксированных переменных ■.

$x_1 x_2 \dots x_n$	f	
0 0…0 0	α_0	2^{2^n}
0 0… 0 1	α_1	
… … …	…	
1 1 … 1 1	α_{2^n-1}	

2^n

Оценим, например, число функций от 10 переменных. Всего таких функций будет $2^{2^{10}} = 2^{1024} > 2^{1000} = (2^{10})^{100} > (1000)^{100} = 10^{300}$. Таким образом при росте числа переменных число функций возрастает очень быстро, и их табличное задание становится неудобным.

Равенство функций

В обычной алгебре справедливо равенство $x + y - y = x$, хотя в левой части задана функция от двух переменных, а в правой - от одной. Но от y в левой части ничего не зависит, то есть $y -$ *фиктивная переменная*. Получается, что функции от разного числа переменных могут быть одинаковыми при наличии *фиктивных переменных.*

Опр. Переменная x_i называется *существенной* для функции алгебры логики $f(x_1, \dots, x_n)$, если существуют такие $\alpha_1, \dots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \dots, \alpha_n \in E_2$, что

$$f(\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n) \neq f(\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n)$$

Такие наборы называют *соседними по i-й переменной*. В противном случае переменная x_i называется *фиктивной*.

Опр. Переменная x_i называется *фиктивной* для функции алгебры логики $f(x_1, \dots, x_n)$, если для $\forall\ \alpha_1, \dots, \alpha_{i-1}, \alpha_i, \alpha_{i+1}, \dots, \alpha_n \in E_2$

$$f(\alpha_1, \dots, \alpha_{i-1}, 0, \alpha_{i+1}, \dots, \alpha_n) = f(\alpha_1, \dots, \alpha_{i-1}, 1, \alpha_{i+1}, \dots, \alpha_n)$$

При этом, если x_i - фиктивная переменная функции *f*, то функция f однозначно определяется некоторой функцией $g(\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n)$, полученной *изъятием* x_i переменной. Аналогично можно расширить таблицу любой функции введением любого числа фиктивных переменных.

Опр. Две функции алгебры логики называются *равными*, если одну из них можно получить из другой путем добавления и изъятия любого числа фиктивных переменных.

Чтобы на практике проверить две функции на равенство, необходимо из обеих функций изъять все фиктивные переменные и сравнить их таблицы, за исключением вырожденного случая после изъятия всех фиктивных переменных их таблицы должны совпасть. Вырожденный случай – это функция-константа, у которой все переменные 0 или 1 фиктивные.

Формулы

Опр. Пусть $A = \{f_1(\dots), f_2(\dots), \dots, f_n(\dots), \dots\}$ – некоторое множество функций.

Введем понятие *формулы над A*:

1) Любая функция из A называется *формулой над A.*
2) Пусть $f(x_1, \dots, x_n)$– функция из A, и пусть $H_1, H_2, \dots, H_n$ – либо переменная, либо формула над A.
   Тогда выражение вида $f(H_1, H_2, \dots, H_n)$ также является формулой над A, причем среди $H_1, H_2, \dots, H_n$ могут быть одинаковые.
3) Формулами над A называются только те объекты, которые можно построить по пунктам 1 и 2 за конечное число шагов.

Каждой формуле сопоставляем функцию (индуктивно):

1) Если формула – это функция из A, то этой формуле сопоставляется сама функция.
2) Если формула имеет вид $f(H_1, H_2, \dots, H_n)$, то соответствующая ей функция на любом наборе $(\alpha_1, \alpha_2, \dots, \alpha_k)$ вычисляется в 2 шага:
   1) Получаем значения всех H_i на $(\alpha_1, \alpha_2, \dots, \alpha_k)$
   2) Вычисляем $f(\beta_1, \beta 2, \dots, \beta_n)$, где $\beta_i = H_i(\alpha_1, \alpha_2, \dots, \alpha_k)$

Опр. Две формулы называются *эквивалентными*, если соответствующие им функции равны.

Основные эквивалентности

Коммутативность	*Ассоциативность*
$x \vee y = y \vee x$ $x \cdot y = y \cdot x$ $x \oplus y = y \oplus x$ $x \sim y = y \sim x$ $x \mid y = y \mid x$ $x \downarrow y = y \downarrow x$	$(x \vee y) \vee z = x \vee (y \vee z) = x \vee y \vee z$ $(x \cdot y) \cdot z = x \cdot (y \cdot z) = x \cdot y \cdot z$ $(x \oplus y) \oplus z = x \oplus (y \oplus z) = x \oplus y \oplus z$

Законы поглощения	*Другие*
$x \vee x = x$ $x \cdot x = x$ $x \vee \bar{x} = 1$	$\bar{\bar{x}} = x$ $x \oplus 1 = \bar{x}$ $x_1 \mid x_2 = \overline{x_1 x_2}$

$x \vee 1 = 1$ $x \cdot 1 = x$ $x \vee 0 = x$ $x \cdot 0 = 0$	$x_1 \downarrow x_2 = \overline{x_1 \vee x_2}$ $x \to y = \bar{x} \vee \bar{y}$ $x \oplus y = (x \cdot \bar{y}) \vee (\bar{x} \cdot y)$ $x \sim y = \overline{x \oplus y} = (x \cdot y) \vee (\bar{x} \cdot \bar{y})$

Приоритет конъюнкции выше, чем приоритеты дизъюнкции и суммы по модулю 2. Благодаря этому можно опустить ряд ненужных скобок.

Очевидны следующие утверждения:

$x_1 \cdot x_2 \cdot \ldots \cdot x_n = 1 \Leftrightarrow \forall i \;\; x_i = 1$

$x_1 \vee x_2 \vee \ldots \vee x_n = 1 \Leftrightarrow \;\; \exists i : x_i = 1$

Опр. *x в степени сигма* называется функция $x^\sigma = \begin{cases} x, & \sigma = 1 \\ \bar{x}, & \sigma = 0 \end{cases}$

$x^\sigma = 1 \Leftrightarrow x = \sigma$

Теорема о разложении функции алгебры логики по переменным.

Для любой функции алгебры логики $f(x_1, x_2, \ldots, x_n)$ и для любого k $(1 \le k \le n)$ справедливо равенство:

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{(\sigma_1, \sigma_2, \ldots, \sigma_k) \in E_2^k} x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \ldots \cdot x_k^{\sigma_k} \cdot f(\sigma_1, \sigma_2, \ldots, \sigma_k, \sigma_{k+1}, \ldots, \sigma_n)$$

Доказательство. Возьмем любой набор $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in E_2^n$ и вычислим значение правой части на этом наборе:

$$\bigvee_{(\sigma_1, \sigma_2, \ldots, \sigma_k) \in E_2^k} \alpha_1^{\sigma_1} \cdot \alpha_2^{\sigma_2} \cdot \ldots \cdot \alpha_k^{\sigma_k} \cdot f(\alpha_1, \alpha_2, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n) =$$

$$= 0 \vee 0 \vee 0 \vee \ldots \vee \alpha_1^{\alpha_1} \cdot \alpha_2^{\alpha_2} \cdot \ldots \cdot \alpha_k^{\alpha_k} \cdot f(\alpha_1, \alpha_2, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n) =$$

$$= 1 \cdot 1 \cdot 1 \cdot 1 \cdot \ldots \cdot f(\alpha_1, \alpha_2, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n) =$$

$$= f(\alpha_1, \alpha_2, \ldots, \alpha_k, \alpha_{k+1}, \ldots, \alpha_n)$$

Если хотя бы один из сомножителей будет равен нулю, то есть хотя бы один α_i не совпадет с σ_i, то вся конъюнкция обратится в нуль. Таким образом, из ненулевых конъюнкций останется лишь одна – та, в которой $\alpha_i = \sigma_i$.

Получаем равенство левой части на наборе $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in E_2^n$ ■.

Лекция 2. Элементарная конъюнкция и элементарная дизъюнкция.

Рассмотрим теперь следствия из теоремы о разложении функции алгебры логики по переменным.

Следствие 1 (разложение по одной переменной).

Для любой функции алгебры логики $f(x_1, x_2, \dots, x_n)$ верно равенство

$$f(x_1, x_2, \dots, x_n) = \overline{x_1} \cdot f(0, x_2, \dots, x_n) \vee x_1 \cdot f(1, x_2, \dots, x_n)$$

Это частный случай теоремы при $k = 1$.

Построим физический преобразователь, который вычисляет значение $f(x_1, x_2, \dots, x_n)$ (рис. 2.1)

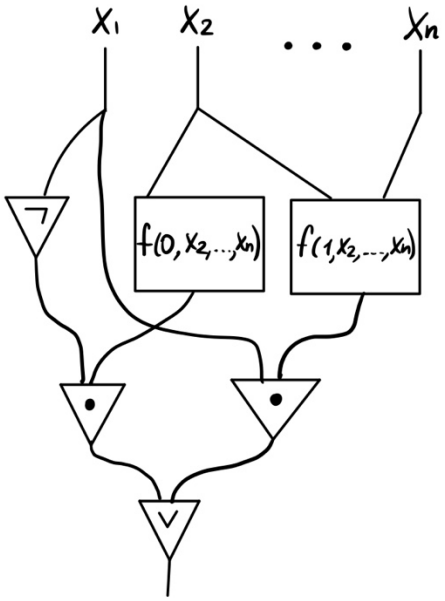

Рис. 2.1. Преобразователь $f(x_1, x_2, \dots, x_n)$

Следствие 2 (теорема о совершенной дизъюнктивной нормальной форме).

Для любой функции алгебры логики $f(x_1, x_2, \dots, x_n)$, отличной от тождественного нуля, справедливо следующее равенство:

$$f(x_1, x_2, \dots, x_n) = \bigvee_{(\sigma_1,\sigma_2,\dots,\sigma_n)\in E_2^n : f(\sigma_1,\sigma_2,\dots,\sigma_n)=1} x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_n^{\sigma_n}$$

Доказательство. По теореме о разложении по k переменным при $k = n$ получим

$$f(x_1, x_2, \dots, x_n) = \bigvee_{(\sigma_1,\sigma_2,\dots,\sigma_k)\in E_2^k} x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_k^{\sigma_k} \cdot f(\sigma_1, \sigma_2, \dots, \sigma_k, \sigma_{k+1}, \dots, \sigma_n) =$$

$$= \bigvee_{(\sigma_1,\sigma_2,\dots,\sigma_n)\in E_2^n : f(\sigma_1,\sigma_2,\dots,\sigma_n)=1} x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_n^{\sigma_n} \cdot f(\sigma_1, \sigma_2, \dots, \sigma_n) \vee$$

$$\vee \bigvee_{(\sigma_1,\sigma_2,\dots,\sigma_n)\in E_2^n : f(\sigma_1,\sigma_2,\dots,\sigma_n)=0} x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_n^{\sigma_n} \cdot f(\sigma_1, \sigma_2, \dots, \sigma_n) =$$

$$= \bigvee_{(\sigma_1,\sigma_2,\dots,\sigma_n)\in E_2^n : f(\sigma_1,\sigma_2,\dots,\sigma_n)=1} x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_n^{\sigma_n} \cdot 1 \vee$$

$$\vee \bigvee_{(\sigma_1,\sigma_2,\dots,\sigma_n)\in E_2^n : f(\sigma_1,\sigma_2,\dots,\sigma_n)=0} x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_n^{\sigma_n} \cdot 0 =$$

$$= \bigvee_{(\sigma_1,\sigma_2,\dots,\sigma_n)\in E_2^n : f(\sigma_1,\sigma_2,\dots,\sigma_n)=1} x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_n^{\sigma_n} \ \blacksquare.$$

Опр. *Совершенной нормальной дизъюнктивной формой* функции алгебры логики $f(x_1, x_2, \dots, x_n)$ называется разложение вида

$$f(x_1, x_2, \dots, x_n) = \bigvee_{(\sigma_1,\sigma_2,\dots,\sigma_n)\in E_2^n : f(\sigma_1,\sigma_2,\dots,\sigma_n)=1} x_1^{\sigma_1} \cdot x_2^{\sigma_2} \cdot \dots \cdot x_n^{\sigma_n}$$

Теорема о совершенной конъюнктивной форме.

Для любой функции алгебры логики $f(x_1, x_2, \dots, x_n)$, отличной от тождественной единицы, справедливо следующее равенство:

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{(\sigma_1,\sigma_2,\dots,\sigma_n)\in E_2^n : f(\sigma_1,\sigma_2,\dots,\sigma_n)=0} x_1^{\overline{\sigma_1}} \vee x_2^{\overline{\sigma_2}} \vee \dots \vee x_n^{\overline{\sigma_n}}$$

Дизъюнктивные нормальные формы

Опр. *Литералом* называется либо переменная, либо отрицание переменной.

То есть от n переменных есть $2n$ литералов: $x_1, x_2, \dots, x_n$ и $\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}$.

Опр. *Элементарной конъюнкцией (дизъюнкцией)* называется конъюнкция (дизъюнкция) литералов от разных переменных.

Элементарная конъюнкция – $t_1 \cdot t_2 \cdot \dots \cdot t_s$, $s \geq 1$, где все t_i – литерал от переменной.

Элементарная дизъюнкция – $t_1 \vee t_2 \vee \dots \vee t_s$, $s \geq 1$, где все t_i – литерал от переменной.

Опр. Дизъюнктивной (конъюнктивной) нормальной формой называется дизъюнкция (конъюнкция) различных элементарных конъюнкций (дизъюнкций).

Будем считать, что конъюнкции (дизъюнкции) совпадают с точностью до перестановки переменных, поскольку они коммутативны.

Опр. Элементарная конъюнкция $K = x_{i_1}^{\sigma_1} \cdot x_{i_2}^{\sigma_2} \cdot \ldots \cdot x_{i_k}^{\sigma_k}$ называется *импликантой* функции алгебры логики $f(x_1, x_2, \ldots, x_n)$, если для любого набора $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_2) \in E_2^n$ выполняется: если $K(\tilde{\alpha}) = 1$, то и $f(\tilde{\alpha})$.

Теорема 2.1.

Если любая функция алгебры логики $f(x_1, x_2, \ldots, x_n)$ представлена в виде ДНФ (дизъюнктивной нормальной формы) $f(x_1, x_2, \ldots, x_n) = K_1 \vee K_2 \vee \ldots \vee K_s$, то каждая K_i является импликантой функции *f*.

Доказательство. Пусть $\forall \tilde{\alpha} = (\alpha_1, \ldots, \alpha_2) \in E_2^n$ и $K_i(\tilde{\alpha}) = 1$.

Тогда $f(\tilde{\alpha}) = K_1(\tilde{\alpha}) \vee K_2(\tilde{\alpha}) \vee \ldots \vee K_i(\tilde{\alpha}) \vee \ldots \vee K_s(\tilde{\alpha}) = \cdots \vee 1 \vee \ldots = 1$, и по определению $\forall i$ $K_i -$ импликанта функции *f* ■.

Для поиска ДНФ на практике используют стандартные упрощения:

1) $x \cdot K \vee \bar{x} \cdot K = (x \vee \bar{x}) \cdot K = 1 \cdot K$
2) $K_1 \vee K_1 K_2 = K_1 \cdot 1 \vee K_1 \cdot K_2 = K_1 \cdot (1 \vee K_2) = K_1 \cdot 1 = K_1$ – поглощение
3) $K_1 \vee K_1 = K_1$

Опр. Пусть K и K_1 – две элементарные конъюнкции. Будем говорить, что K_1 *содержится в* K, если либо $K_1 = K$, либо K_1 можно получить из K вычеркиванием некоторых литералов ($K = K_1 \cdot K_2$).

Замечание. Если K_1 содержится в K и $K(\tilde{\alpha}) = 1$, то и $K_1(\tilde{\alpha}) = 1$.

Опр. Простой импликантой функции алгебры логики $f(x_1, x_2, \ldots, x_n)$ называется элементарная конъюнкция K такая, что сама она – импликанта функции *f*, но она не содержит других импликант функции f отличных от *K*.

Другими словами, если из *K* вычеркнуть какую-то переменную (несколько переменных), то полученная конъюнкция уже не является импликантой функции f.

Утверждение. Любая импликанта функции f содержит хотя бы одну простую импликанту функции f.

Доказательство. Пусть K - импликанта функции f. Тогда либо K – простая и утверждение доказано, либо K содержит другую импликанту $K_1 \neq K$. Теперь либо $K_1 -$ простая импликанта и утверждение доказано, либо K_1 содержит импликанту $K_2 \neq K_1$ и так далее. Процесс должен остановиться, то есть получится простая импликанта ■.

Опр. Пусть $f(x_1, x_2, \dots, x_n) \neq const$. *Сокращенной ДНФ* функции f называется дизъюнкция всех простых импликант функции f.

Теорема 2.2. Сокращенная ДНФ функции $f(x_1, x_2, \dots, x_n)$ реализует функцию $f(x_1, x_2, \dots, x_n)$.

Доказательство. Пусть $F = K_1 \vee K_2 \vee \dots \vee K_m -$ сокращенная ДНФ функции f. Возьмем любой набор $\tilde{\alpha}$. Пусть $F(\tilde{\alpha}) = 1$. Тогда $\exists i\ K_i(\tilde{\alpha}) = 1$ и так как $K_i -$ импликанта для f, то $f(\tilde{\alpha}) = 1$.

Пусть $f(\tilde{\alpha}) = 1$ и $\tilde{\alpha} = (\alpha_1, \dots, \alpha_n)$. Рассмотрим элементарную конъюнкцию $K_0 = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdot \dots \cdot x_n^{\alpha_n}$. $K_0 = 1$ только на одном наборе $(x_1 = \alpha_1, \dots, x_n = \alpha_n)$. Тогда если $K_0(\tilde{\beta}) = 1$, то и $f(\tilde{\beta}) = 1$. Следовательно, $K_0 -$ импликанта функции f и (по утверждению) существует простая импликанта K_j, которая содержится в K_0. K_j есть в сокращенной ДНФ F (по определению), $K_0(\tilde{\alpha}) = 1$, и (по замечанию) $K_j(\tilde{\alpha}) = 1$, соответсвенно и $K_j(\tilde{\alpha}) = 1$.

Получается, $F(\tilde{\alpha}) = 1 \Leftrightarrow f(\tilde{\alpha}) = 1$. Тогда $F \equiv f$ ■.

Лекция 3. Сокращённая дизъюнктивная форма. Полные системы.

Метод Нельсона

Метод Нельсона построения сокращенной ДНФ.

Пусть f и g от переменных $x_1, x_2, \dots, x_n$ заданы своими сокращенными ДНФ:

$$f = K_1' \vee K_2' \vee \dots \vee K_m', \quad g = K_1'' \vee K_2'' \vee \dots \vee K_l''$$

Рассмотрим следующий алгоритм:

1. Рассмотрим функцию $f \cdot g = (K_1' \vee K_2' \vee \dots \vee K_m') \cdot (K_1'' \vee K_2'' \vee \dots \vee K_l'') = \bigvee_{i,j} K_i' \cdot K_j''$
2. Упростим это выражение в два этапа:
   1. Упростим $\forall i \;\; K_i' \cdot K_i''$ по правилам: $t \cdot t = t, \;\; t \cdot \bar{t} = 0, \;\; 0 \cdot A = 0,$ $0 \vee A = A$. Получим выражение $D_1 -$ дизъюнкция элементарных конъюнкций.
   2. Упростим выражение D_1, используя $K \vee K = K, \; K_1 \vee K_1 \cdot K_2 = K_1$. Получим выражение D_2.

   Этап 1 и этап 2 выполняются, пока возможно.

Теорема 3.1 (о сокращенной ДНФ).

Пусть f и g от переменных $x_1, x_2, \dots, x_n$ представлены своими сокращенными ДНФ, тогда описанный алгоритм строит сокращенную ДНФ функции $f \cdot g$.

Доказательство. Так как все преобразования, описанные в алгоритме, — это тождества, то D_1 и D_2 реализуют функцию $f \cdot g$. Поэтому все слагаемые в D_1 и в D_2 – это импликанты функции $f \cdot g$.

Лемма 1. Если $f = K_1' \vee K_2' \vee \dots \vee K_m' -$ сокращенная ДНФ и $K -$ любая импликаната функции $f \cdot g$, то среди слагаемых $K_1', K_2', \dots, K_m' \; \exists \; K_j'$, которая содержится в K.

Доказательство. Возьмем любой набор $\forall \; \tilde{\alpha} \;\; K(\tilde{\alpha}) = 1$, так как $K -$ импликанта для $f \cdot g$, то $(f \cdot g)(\tilde{\alpha}) = 1 \;\; \Rightarrow \;\; f(\tilde{\alpha}) = 1 \;\; \Rightarrow$ по определению $K -$ импликанта функции f.

Тогда по утверждению, $\exists$ простая импликанта K_j' функции f, которая содержится в K. Аналогично для функции g ■.

Лемма 2. Пусть $K -$ простая импликанта для функции $f \cdot g$. Тогда она есть как слагаемое в D_1.

Доказательство. Пусть K – простая импликанта для функции $f \cdot g$. Тогда по *Лемме 1* $\exists$ такие K_j' и K_i'', что K_j' содержится в K и K_i''содержится в K. Тогда все литералы из произведения $K_j' \cdot K_i''$ содержатся в K. Так как все сомножители содержатся в K, то в этом произведении нет противоположных литералов. Следовательно, $K_j' \cdot K_i'' \not\equiv 0$ и на этапе 1 из $K_j' \cdot K_i''$ получится элементарная конъюнкция K_0, которая содержит только литералы из K. Но K_0 – импликантна функции $f \cdot g$ и K – простая импликанта. Это возможно, только если $K_0 \equiv K$ ■.

Лемма 3. Пусть K – простая импликанта для функции $f \cdot g$. Тогда она есть как слагаемое в D_2.

Доказательство. Пусть K – простая импликанта для функции $f \cdot g$. Тогда по *Лемме 2* K есть как слагаемое в D_1. Поглотиться на этапе 2 она могла бы только слагаемыми (а они все импликантны функции $f \cdot g$), которые содержатся строго в K. Но таких нет, так как K – простая импликанта ■.

Лемма 4. В D_2 содержатся только простые импликанты функции $f \cdot g$.

Доказательство. В D_2 содержатся только импликантны функции $f \cdot g$. Допустим, в D_2 есть не простая импликанта K_0 функции $f \cdot g$. По утверждению существует простая импликантна K''' функции $f \cdot g$, которая содержится в K_0 (строго). По *Лемме 3* она есть в D_2, но тогда алгоритм не мог остановиться, так как к паре K''' и K_0 можно было бы применить поглощение. Следовательно, возникло противоречие из предположения, что в D_2 есть не простая импликанта ■.

Из Лемм 1-4 вытекает **теорема 3.1 (о сокращенной ДНФ)** ■.

Лемма 5. Пусть $g(x_1, \dots, x_n) = x_{i_1}^{\sigma_1} \vee x_{i_2}^{\sigma_2} \vee \dots \vee x_{i_k}^{\sigma_k}, \quad k \geq 1$

Тогда правая часть – сокращенная ДНФ функции g.

Доказательство. $\forall$ слагаемое $x_{i_j}^{\sigma_j}$ по определению является импликантой функции g, причем простой импликантой, так как ничего нельзя больше вычеркнуть.

Если импликанта K функции g содержит хотя бы одно из выписанных слагаемых (строго), то она по определению не простая и соответственно, не входит в ДНФ.

Пусть элементарная конъюнкция K не содержит литералов $x_{i_1}^{\sigma_1}, x_{i_2}^{\sigma_2}, \dots, x_{i_k}^{\sigma_k}$, то есть либо в литералах другие переменные, либо в K литералы, противоположные литералам из g.

Положим $x_{i_1} = \overline{\sigma_1}, x_{i_2} = \overline{\sigma_2}, \dots, x_{i_k} = \overline{\sigma_k}$. Тогда $g = 0$. При этом литералы в K, зависящие от этих же переменных примут значение 1. Можем подобрать значение оставшихся переменных так, что $K(\tilde{\alpha}) = 1$. Имеем $K(\tilde{\alpha}) = 1$ и $g(\tilde{\alpha}) = 0 \Rightarrow$ по определению K не импликантна для функции g. Соответственно, в правой части уравнения *Леммы 5* стоят все простые импликанты ⇒ это сокращенная ДНФ ■.

Метод Нельсона построения сокращенной ДНФ заключается в следующем:

Для f строим какую-нибудь КНФ (например, совершенную КНФ) $D_1 \cdot D_2 \cdot \dots \cdot D_s$.

Затем постепенно раскрываем скобки, при каждом раскрытии производим упрощение (до упора) по этапам 1 и 2.

Поскольку элементарная дизъюнкция, как мы выяснили, является своей же сокращенной ДНФ, каждый раз получаем сокращенную ДНФ для определенного кусочка по теореме, доказанной выше (рис. 3.1).

Рис. 3.1. Преобразование в сокращенную ДНФ

Можно упрощать не по шагам, а сначала раскрыть все скобки, все упростить и провести поглощение.

Полные системы

Опр. Множество функций алгебры логики $A = \{f_1, f_2, \dots\}$ называется *полной системой*, если любую функцию алгебры логики можно выразить формулой над множеством A.

Самая тривиальная полная система – все функции алгебры логики.

Теорема. Система $A = \{x \vee y,\ x \cdot y,\ \bar{x}\}$ является полной.

Доказательство.

1) Пусть $f(x_1, \dots, x_n) \not\equiv 0$. Тогда ее можно представить в виде совершенной ДНФ, то есть формулой над $A = \{x \vee y,\ x \cdot y,\ \bar{x}\}$.

2) Пусть $f(x_1, \dots, x_n) \equiv 0$. Тогда $f(x_1, \dots, x_n) = x_1 \cdot \overline{x_1}$ – формула над A ■.

Лемма. Пусть $A = \{f_1, f_2, \dots\}$ - полная система в P_2 и все функции из A выражаются формулами над системой $B = \{g_1, g_2, \dots\}$. Тогда B – полная система.

Доказательство. Пусть $f(x_1, \dots, x_n)$ – любая функция алгебры логики. Тогда f можно выразить формулой, в которой участвуют только f_1, f_2, ..., так как A – полная система.

$\forall f_i$ можно заменить его представлением через $g_1, g_2, \dots$ по условию. Получим формулу, выражающую f через $g_1, g_2, \dots$. Тогда по определению B – полная система ■.

Теорема. Следующие системы являются *полными* в P_2:

1) $\{x \vee y,\ \bar{x}\}$
2) $\{x \cdot y,\ \bar{x}\}$
3) $\{x|y\}$
4) $\{x \cdot y,\ x \oplus y,\ 1\}$

Доказательство.

1) Известно, что система $A = \{x \vee y,\ x \cdot y,\ \bar{x}\}$ является полной. Покажем, что $B = \{x \vee y,\ \bar{x}\}$ тоже полна. Из закона де Моргана $\overline{x \cdot y} = \bar{x} \vee \bar{y}$ получим, что $x \cdot y = \overline{\bar{x} \vee \bar{y}}$. Тогда по *лемме* все функции системы A выражаются формулами над B, соотвественно система B полна.
2) Аналогично пункту 1: $\overline{x \vee y} = \bar{x} \cdot \bar{y} \ \Leftrightarrow\ x \vee y = \overline{\bar{x} \cdot \bar{y}}$.
3) Сведем к $\{x \cdot y, \bar{x}\}$: $x|x = \bar{x},\ x \cdot y = \left(\overline{x|y}\right) = (x|y)|(x|y)$
4) Сведем к $\{x \cdot y, \bar{x}\}$: $x \cdot y$ уже есть, а $\bar{x} = x \oplus 1$ ■.

Полиномы Жегалкина

<u>Опр.</u> *Монотонной конъюнкцией* над переменными $x_1, x_2, \dots, x_n$ называется выражение 1 или произведение k $(k \geq 1)$ <u>различных</u> переменных <u>без отрицания,</u> всего их 2^n.

<u>Опр.</u> *Полиномом Жегалкина* над переменными $x_1, x_2, \dots, x_n$ называется либо выражение 0, либо сумма по модулю 2 k $(k \geq 1)$ <u>различных</u> монотонных конъюнкций над переменными $x_1, x_2, \dots, x_n$, всего их 2^{2^n}.

$$K_1 \oplus K_2 \oplus K_3 \oplus \dots \oplus K_l$$

Теорема Жегалкина. $\forall$ функция алгебры логики $f(x_1, x_2, \dots, x_n)$ может быть *единственным образом* представлена полиномом Жегалкина.

Доказательство.

1) Пусть $f(x_1, x_2, \dots, x_n) - \forall$ функция алгебры логики. Если $f \equiv 0$, то 0 – ее полином Жегалкина.

   Если $f(x_1, x_2, \dots, x_n) \not\equiv 0$, то представим f в виде совершенной ДНФ:

   $$f(x_1, x_2, \dots, x_n) = K_1 \vee K_2 \vee \dots \vee K_s \ \ (s \geq 1)$$

   Так как $K_i -$ элементарная конъюнкция, содержащая все переменные $x_1, x_2, \dots, x_n$, то $K_i = 1$ ровно на одном наборе, причем разные K_i равны 1 на разных наборах. Следовательно, на любом наборе среди $K_1, K_2, \dots, K_s$ либо все равны 0, либо ровно одна равна 1.

   Но тогда $f(x_1, x_2, \dots, x_n) = K_1 \vee K_2 \vee \dots \vee K_s = K_1 \oplus K_2 \oplus \dots \oplus K_s$, заменяя в последней формуле все $\overline{x_j} = x_j \oplus 1$, раскрывая все скобки и приводя подобные, получим полином Жегалкина.

Лекция 4. Полиномы Жегалкина. Замкнутые классы.

Теорема Жегалкина

Теорема Жегалкина. $\forall$ функция алгебры логики $f(x_1, x_2, \dots, x_n)$ может быть *единственным образом* представлена полиномом Жегалкина.

Доказательство.

1) *Существование.*
Пусть $f(x_1, x_2, \dots, x_n) - \forall$ функция алгебры логики. Если $f \equiv 0$, то 0 – ее полином Жегалкина.
Если $f(x_1, x_2, \dots, x_n) \not\equiv 0$, то представим f в виде совершенной ДНФ:
$$f(x_1, x_2, \dots, x_n) = K_1 \vee K_2 \vee \dots \vee K_s \ \ (s \geq 1)$$
Так как $K_i -$ элементарная конъюнкция, содержащая все переменные $x_1, x_2, \dots, x_n$, то $K_i = 1$ ровно на одном наборе, причем разные K_i равны 1 на разных наборах. Следовательно, на любом наборе среди $K_1, K_2, \dots, K_s$ либо все равны 0, либо ровно одна равна 1.
Но тогда $f(x_1, x_2, \dots, x_n) = K_1 \vee K_2 \vee \dots \vee K_s = K_1 \oplus K_2 \oplus \dots \oplus K_s$, заменяя в последней формуле все $\overline{x_j} = x_j \oplus 1$, раскрывая все скобки и приводя подобные, получим полином Жегалкина.

2) *Единственность.*
Пусть $0 \leq i \leq 2^n - 1 \ \ i = (i_1, i_2, \dots, i_n)_2$.
Пусть в этом представлении единицы стоят на позициях с номерами $j_1, j_2, \dots, j_k$ и пусть зафиксированы переменные $x_1, x_2, \dots, x_n$. Тогда через $K_i(x_1, x_2, \dots, x_n)$ будем обозначать конъюнкцию $K_i(x_1, \dots, x_n) = x_{j_1} \cdot x_{j_2} \cdot \dots \cdot x_{j_k}$. Это будет взаимно однозначное соответствие между всеми целыми числами от 0 до $2^n - 1$ и всеми монотонными конъюнкциями.
Например, у нас 4 переменные: $x_1\ x_2\ x_3\ x_4$. Числу 5 в двоичной системе $5 = (0101)_2$ соответствует конъюнкция $x_2 \cdot x_4$ (выбрали переменные, равные 1). x_3 соответствует число $(0\ 0\ 1\ 0)_2 = 2$, а $(0\ 0\ 0\ 0)_2$ соответствует 1.
Всего конъюнкций от $x_1, x_2, \dots, x_n$ может быть 2^n. При этом произвольный полином Жегалкина можно представить в виде **суммы по модулю 2**:
$$\sum_{i=0}^{2^n-1} b_i \cdot K_1(x_1, \dots, x_n), \qquad b_i \in \{0,1\}$$
Тогда любой полином Жегалкина от переменных $x_1, \dots, x_n$ можно представить двоичным вектором $(b_0, b_1, b_2, \dots, b_{2^n-1})$. Всего различных полиномов Жегалкина от $x_1, \dots, x_n$ ровно 2^{2^n}.

Получается, что при фиксированных переменных $x_1, \dots, x_n$ у нас есть 2^{2^n} полиномов и 2^{2^n} функций алгебры логики. И если бы двум полиномам соответствовала одна и та же функция алгебры логики, то, учитывая, что полиномов и функций поровну, существовала бы функция алгебры логики, которая не представлялась бы полиномом. Но это противоречит пункту 1 доказательства теоремы. Следовательно, разным полиномам соответствуют разные функции ■.

Быстрый алгоритм построения полинома Жегалкина

Утверждение. При $0 \le i \le 2^{n-1} - 1$ $i = (i_2 i_3 \dots i_n)_2 = (0\ i_2 \dots i_n)_2$

1) $K_i(x_2, \dots, x_n) = K_i(x_1, x_2, \dots, x_n)$
2) $x_1 \cdot K_i(x_2, \dots, x_n) = K_{i+2^{n-1}}(x_1, x_2, \dots, x_n)$

Пример.

Чтобы построить $K_3(x_2, x_3, x_4, x_5)$, нужно записать 3 в двоичном представлении и составить конъюнкцию из переменных, соответствующих 1. Так как $3 = 0011_2$, то

$$K_3(x_2, x_3, x_4, x_5) = x_4 \cdot x_5$$

Для $K_3(x_1, x_2, x_3, x_4, x_5)$ $3 = 00011_2$

$$K_3(x_1, x_2, x_3, x_4, x_5) = x_4 \cdot x_5$$

Аналогично и для $K_3(x_3, x_4, x_5) = x_4 \cdot x_5$.

То есть если в кодировки спереди дописали 0, то ничего не изменилось, а если дописали 1, то к номеру прибавится 2^{n-1}.

Лемма. Пусть $f(x_1, \dots, x_n)$ − произвольная функция алгебры логики и f представляется **суммой по модулю 2:**

$$f(0, x_2, \dots, x_n) = \bigoplus_{i=0}^{2^{n-1}-1} c_i K_i(x_2, \dots, x_n)$$

$$f(1, x_2, \dots, x_n) = \bigoplus_{i=0}^{2^{n-1}-1} d_i K_i(x_2, \dots, x_n)$$

Тогда

$$f(x_1, \dots, x_n) = \bigoplus_{j=0}^{2^{n-1}-1} b_j K_j(x_1, x_2, \dots, x_n)$$

$$b_j = \begin{cases} c_j, & j = \overline{0, 2^{n-1}-1} \\ d_j \oplus c_j, & j = \overline{2^{n-1}, 2^n-1} \end{cases}$$

То есть первую половину вектора сохраняем, а ко второй половине вектора покоординатно добавляем первую половину вектора.

Доказательство.

$f(x_1, \dots, x_n) = \overline{x_1} \cdot f(0, x_2, \dots, x_n) \vee x_1 \cdot f(1, \dots, x_n) =$

$= \overline{x_1} \cdot f(0, x_2, \dots, x_n) \oplus x_1 \cdot f(1, \dots, x_n) = (x_1 \oplus 1) \cdot f(0, x_2, \dots, x_n) \oplus x_1 \cdot f(1, \dots, x_n)$

$= f(0, x_2, \dots, x_n) \oplus x_1 \cdot \big(f(0, x_2, \dots, x_n) \oplus f(1, x_2, \dots, x_n)\big) =$

$= \sum_{i=0}^{2^{n-1}-1} c_i \cdot K_i(x_2, \dots, x_n) \oplus x_1 \cdot \sum_{i=0}^{2^{n-1}-1} (d_i \oplus c_i) \cdot K_i(x_2, \dots, x_n) =$

$= \sum_{i=0}^{2^{n-1}-1} c_i \cdot K_i(x_1, x_2, \dots, x_n) \oplus \sum_{i=0}^{2^{n-1}-1} (d_i \oplus c_i) \cdot x_1 \cdot K_i(x_2, \dots, x_n) =$

$= \sum_{i=0}^{2^{n-1}-1} c_i \cdot K_i(x_1, x_2, \dots, x_n) \oplus \sum_{i=0}^{2^{n-1}-1} (d_i \oplus c_i) \cdot K_{i+2^{n-1}}(x_1, x_2, \dots, x_n)$

В доказательстве $\sum_{i=0}^{2^{n-1}-1}$ *– сумма по модулю 2* ■.

Пусть функция представлена своим вектором

$$f(x_1, x_2, \dots, x_n) = (a_o a_1 \dots a_{2^{n-1}-1} a_{2^{n-1}} \dots a_{2^n-1})$$

Тогда

$$f(0, x_2, \dots, x_n) = (a_o a_1 \dots a_{2^{n-1}-1})$$

$$f(1, x_2, \dots, x_n) = (a_{2^{n-1}} \dots a_{2^n-1})$$

Пусть $f(x_1, x_2, \dots, x_n) = \sum_{i=0}^{2^{n-1}-1} b_i K_i(x_1, x_2, \dots, x_n)$ – полином Жегалкина.

Будем обозначать $p(f) = (b_0 b_1 \dots b_{2^n-1})$ – вектор полинома.

Лемма утверждала, что если $p(f(0, x_2, \dots, x_n)) = (c_0 c_1 \dots c_{2^n-1}) = \tilde{c}$ и $p(f(1, x_2, \dots, x_n)) = (d_0 d_1 \dots d_{2^n-1}) = \tilde{d}$, то $p(f) = (\tilde{c}, \tilde{c} \oplus \tilde{c})$.

В свою очередь, чтобы построить $\tilde{c}$, нужно применить этот же алгоритм, «разрезав» $\tilde{c}$ пополам, еще пополам и так далее итеративно (рис. 4.1).

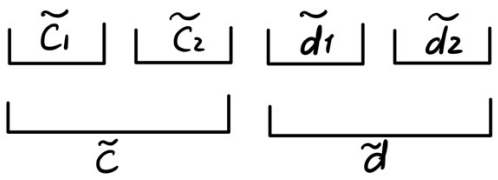

Рис. 4.1. Итеративное построение вектора

Утверждение. Пусть $f(x) = (a\ b)$. Тогда $p(f) = (a, a \oplus b)$.

1) $f(x) = (0\ 0) = 0 \Rightarrow p(f) = (0\ 0)$
2) $f(x) = (0\ 1) = x \Rightarrow p(f) = (0\ 1)$
3) $f(x) = (1\ 0) = x \oplus 1 \Rightarrow p(f) = (1\ 1)$
4) $f(x) = (1\ 1) = 1 \Rightarrow p(f) = (1\ 0)$

Замкнутые классы

Опр. Пусть A – любое подмножество функций алгебры логики ($A \subseteq P_2$). Тогда *замыканием* A называется множество всех функций алгебры логики, которое можно выразить над A.

Обозначение: $[A]$

Свойства:

1) $A \subseteq [A]$
2) $A \subseteq B \Rightarrow [A] \subseteq [B]$
3) $\big[[A]\big] = [A]$

Опр. Система функций алгебры логики называется *полной*, если $[A] = P_2$

$$A - \text{полная} \Leftrightarrow [A] = P_2$$

Опр. Множество функций алгебры логики называется *замкнутым классом*, если $[A] = A$

$$A - \text{замкнутый класс} \Leftrightarrow [A] = A$$

Утверждение. Пусть $A -$ замкнутый класс и $A \neq P_2$ и $B \subseteq A$. Тогда $B -$ не полная система.

Доказательство. $B \subseteq A \Rightarrow [A] \subseteq [B] = A \neq P_2 \Rightarrow [B] \neq P_2 \Rightarrow B -$ не полная ■.

Класс T_0

Опр. $T_0 = \{f(x_1, \dots, x_n\} \in P_2 \mid f(0,0, \dots,0) = 0\}$

Функции класса в общем виде выглядят так:

$x_1 \dots x_n$	
$0 \dots 0$	0
$\dots \dots \dots$	$\} 2^n - 1$

Все функции класса T_0 принимают на нулевом наборе нулевое значение. Таким образом, всего в классе T_0 функций столько, сколько существует булевских векторов длины $2^n - 1$ (первый разряд вектора длины 2^n должен быть равен 0), то есть $2^{2^n-1} = \frac{1}{2}2^{2^n}$.

Мощность класса $|T_0| = \frac{1}{2}2^{2^n}$.

$f \in T_0$	$f \notin T_0$
$f = 0$	$f = 1$
$f = x$	$f = \bar{x}$
$f = x \cdot y$	$f = x \to y$
$f = x \vee y$	$f = x \mid y$
$f = x \oplus y$	$f = x \downarrow y$
	$f = x \sim y$

Теорема. Класс T_0 – замкнутый.

Доказательство. Пусть $f(x_1, \dots, x_n) \in T_0$ и $g_1(y_1, \dots, y_m) \in T_0, \dots, g_n(y_1, \dots, y_m) \in T_0$ и пусть $h(y_1, .., y_m) = f\big(g(y_1, \dots, y_m), \dots, g_n(y_1, .., y_m)\big)$.

Тогда $h(0, \dots,0) = f\big(g(0, \dots,0), \dots, g_n(0, \dots,0)\big) = f(0, \dots,0) = 0$, так как все $g_i \in T_0$ и $f \in T_0$. Значит, $h(0, \dots,0) = 0$ и по определению $h \in T_0$.

Мы рассмотрели только частный случай без переменных в качестве аргументов, но по определению формулы g_i могут быть просто переменными. Но так как все функции вида $g(x_1, \dots, x_n) \equiv x_i$ входят в T_0, подстановка простых переменных эквивалентна подстановке тождественной функции, то есть это будет частный случай рассмотренного ■.

Класс T_1

Опр. $T_1 = \{f(x_1, \dots, x_n\} \in P_2 \mid f(1,1, \dots,1) = 1\}$

Число функций в T_1 аналогично числу функций в T_0 равно $2^{2^n-1} = \frac{1}{2}2^{2^n}$.

$f \in T_1$	$f \notin T_1$
$f = 1$ $f = x$ $f = x \cdot y$ $f = x \to y$ $f = x \sim y$	$f = 0$ $f = \bar{x}$ $f = x \oplus y$ $f = x \mid y$ $f = x \downarrow y$

Теорема. Класс T_1 – замкнут.

Доказательство повторяет доказательство аналогичной теоремы для T_0.

Класс L линейных функций

Опр. Функция алгебры логики $f(x_1, \dots, x_n)$ называется линейной, если ее можно представить в виде:

$$f(x_1, \dots, x_n) = C_0 \oplus C_1 \cdot x_1 \oplus C_2 \cdot x_n \oplus \dots \oplus C_n \cdot x_n$$

Все $C_i \in \{0,1\}$

Через L обозначаем множество всех линейный функций алгебры логики.

$|L| = 2^{n+1}$

$f \in T_1$	$f \notin T_1$
$f = 1$ $f = 0$ $f = \bar{x} = x \oplus 1$ $f = x \oplus y$ $f = x \sim y$	$f = xy$ $f = x \vee y$ $f = x \to y$ $f = x \mid y$ $f = x \downarrow y$

Теорема. Класс L – замкнутый.

Доказательство. Линейная функция – это либо константа 0 или 1, либо сумма нескольких переменных: $x_{j_1} \oplus x_{j_2} \oplus \dots \oplus x_{j_k} \oplus C_0$, $k \geq 1$. Подставляя такие выражения друг в друга вместо переменных и производя упрощение, будем получать выражения такого же вида ■.

Лекция 5. Замкнутые классы.

С помощью замкнутых классов мы можем доказывать полноту системы. Так, например, система $A = \{x \vee y,\ x \cdot y\}$ – не полная, так как и дизъюнкция, и конъюнкция принадлежат классу T_0.

Двойственность

Опр. Пусть $f(x_1, \dots, x_n) -$ произвольная функция алгебры логики. Тогда функцией, *двойственной* к f, называется функция алгебры логики $f^*(x_1, \dots, x_n) = \bar{f}(\overline{x_1}, \dots, \overline{x_n})$.

Например, $f(x, y) = x \vee y$. Тогда $f^*(x, y) = \overline{\bar{x} \vee \bar{y}} = \bar{\bar{x}} \cdot \bar{\bar{y}} = x \cdot y$

$f(x, y) = x \cdot y \Rightarrow f^*(x, y) = \overline{\bar{x} \cdot \bar{y}} = \bar{\bar{x}} \vee \bar{\bar{y}} = x \vee y$

$f(x) \equiv 0 \Rightarrow f^*(x) = \bar{f}(\bar{x}) \equiv 1$

$f(x, y) = x \oplus y \Rightarrow f^*(x, y) = \overline{\bar{x} \oplus \bar{y}} = (x \oplus 1) \oplus (y \oplus 1) \oplus 1 = x \oplus y \oplus 1 = x \sim y$

Утверждение. Для любой функции алгебры логики $f(x_1, \dots, x_n)$ выполняется

$$f^{**}(x_1, \dots, x_n) = f(x_1, \dots, x_n)$$

Доказательство. $f^{**}(x_1, \dots, x_n) = \bar{f}^*(\overline{x_1}, \dots, \overline{x_n}) = \bar{\bar{f}}(\overline{\overline{x_1}}, \dots, \overline{\overline{x_n}}) = f(x_1, \dots, x_n)$ ■.

Теорема (принцип двойственности).

Пусть

$$f\left(g_1(y_{11}, y_{12}, \dots, y_{1n_1}), g_2(y_{21}, y_{22}, \dots, y_{2n_2}), \dots, g_m(y_{m1}, y_{m2}, \dots, y_{nn_m})\right) = \Phi(y_1, \dots, y_k)$$

Тогда

$$\Phi^*(y_1, \dots, y_k) = f^*\left(g_1^*(y_{11}, y_{12}, \dots, y_{1n_1}), g_2^*(y_{21}, y_{22}, \dots, y_{2n_2}), \dots, g_m^*(y_{m1}, y_{m2}, \dots, y_{nn_m})\right)$$

Доказательство.

$$f^*\left(g_1^*(y_{11}, y_{12}, \dots, y_{1n_1}), g_2^*(y_{21}, y_{22}, \dots, y_{2n_2}), \dots, g_m^*(y_{m1}, y_{m2}, \dots, y_{nn_m})\right) =$$
$$= \bar{f}(\overline{g_1^*}(y_{11}, y_{12}, \dots, y_{1n_1}), \overline{g_2^*}(y_{21}, y_{22}, \dots, y_{2n_2}), \dots, \overline{g_m^*}(y_{m1}, y_{m2}, \dots, y_{nn_m}) =$$
$$= \bar{f}(\overline{\overline{g_1^*}}(\overline{y_{11}}, \overline{y_{12}}, \dots, \overline{y_{1n_1}}), \overline{\overline{g_2^*}}(\overline{y_{21}}, \overline{y_{22}}, \dots, \overline{y_{2n_2}}), \dots, \overline{\overline{g_m^*}}(\overline{y_{m1}}, \overline{y_{m2}}, \dots, \overline{y_{nn_m}}) =$$

$$= \bar{f}(g_1(\overline{y_{11}}, \overline{y_{12}}, \dots, \overline{y_{1n_1}}), g_2(\overline{y_{21}}, \overline{y_{22}}, \dots, \overline{y_{2n_2}}), \dots, g_m(\overline{y_{m1}}, \overline{y_{m2}}, \dots, \overline{y_{nn_m}}) =$$
$$= \bar{\Phi}(\bar{y}_1, \dots, \overline{y_k}) = \Phi^* \ \blacksquare.$$

Класс S

Опр. Функция алгебры логики $f(x_1, .., x_n)$ называется *самодвойственной*, если $f^*(x_1, \dots, x_n) = f(x_1, .., x_n)$.

Опр. Класс всех самодвойственных функций $S = \{f \mid f^* = f\}$.

То есть $\bar{f}(\overline{x_1}, \dots, \overline{x_n}) = f(x_1, \dots, x_n)$.

$f \in S$	$f \notin S$
$f = x$ $f = \bar{x} = x \oplus 1$ $f = x \oplus y \oplus z \oplus a$ $f = m(x, y, x) = xy \vee yz \vee zx = \begin{cases} 1, & x + y + z \geq 2 \\ 0, & x + y + z \leq 1 \end{cases}$ – *функция голосования (медиана) принимает значение, которое принимает большая часть переменных*	$f \equiv 0$ $f \equiv 1$ $f = x \vee y$ $f = x \cdot y$

Очевидно, самодвойственных функций от двух переменных нет, и на симметричных местах стоят симметричные значения. Рассмотрим таблицу задания самодвойственной функции: мы можем задать только одну половину, вторая же будет определяться симметрично заданной.

$x_1 \dots x_n$	
… … …	} 2^{n-1}
… … …	

Таким образом, $|S| = 2^{2^{n-1}} = 2^{\frac{1}{2}2^n} = \sqrt{2^{2^n}}$, то есть квадратный корень от общего числа функций от n переменных.

Теорема. Класс S является замкнутым.

Доказательство. Пусть $f(x_1, .., x_m) \in S$ и $\forall i \ \left(g_i\left(x_{i_1}, x_{i_2}, \dots, x_{i_{n_i}}\right) \in S\right)$.

Рассмотрим $f\left(g(x_{11}, \dots, x_{1n_1}), \dots, g_m(x_{m1}, \dots, x_{mn_m})\right) = \Phi(y_1, y_2, \dots, y_k)$. Докажем, что $\Phi^* \in S$. Рассмотрим $\Phi^*(y_1, y_2, \dots, y_k)$.

По принципу двойственности:

$$\Phi^*(y_1, y_2, \dots, y_k) = f^*\left(g_1^*(x_{11}, \dots, x_{1n_1}), \dots, g_m^*(x_{m_1}, \dots, xmn_m)\right) =$$
$$f\left(g(x_{11}, \dots, x_{1n_1}), \dots, g_m(x_{m1}, \dots, x_{mn_m})\right) = \Phi(y_1, y_2, \dots, y_k).$$

Следовательно, $\Phi^* = \Phi$ и по определению $\Phi \in S$. Так как функции, тождественно равные переменным $\in S$, то общий случай суперпозиции (пункт 2 из определения *формулы*) является частным случаем рассмотренного ■.

Класс M

Опр. Пусть $\tilde{\alpha} = (\alpha_1, \dots, \alpha_n) \in E_2^n$ и $\tilde{\beta} = (\beta_1, \dots, \beta_n) \in E_2^n$.

Будем считать, что $\tilde{\alpha} \leq \tilde{\beta}$, если $\forall i \;\; \alpha_i \leq \beta_i$.

Например, $(0,1,0) \leq (0,1,1)$, а $(0,1,0)$ и $(0,0,1)$ не сравнимы.

Опр. Функция алгебры логики $f(x_1, .., x_n)$ называется *монотонной*, если для $\forall\, \tilde{\alpha}, \tilde{\beta} \in E_2^n$ выполняется импликация: $\tilde{\alpha} \leq \tilde{\beta} \Rightarrow f(\tilde{\alpha}) \leq f(\tilde{\beta})$.

При это, если $\tilde{\alpha}$ и $\tilde{\beta}$ не сравнимы, то импликации нет.

Множество всех монотонных функций алгебры логики обозначаем M.

$f \in M$	$f \notin M$
$f = 1$	$f = \bar{x}$
$f = 0$	$f = x \vee y$
$f = x$	$f = x \to y$
$f = x \vee y$	$f = x \mid y$
$f = xy$	$f = x \downarrow y$
$f = m(x, y, x) = xy \vee yz \vee zx$	$f = x \oplus y$

Число функций в классе M не выражается конечной формулой. Может быть, однажды удастся доказать, что она вообще не выражается через обычные операции конечной формулой. Однако, установлено, как асимптотически ведет себя функция от n переменных класса M при росте n: $|M^n|$.

Теорема. Класс M является замкнутым.

Доказательство. Так как функции, тождественно равные переменной, являются монотонными, то общий случай суперпозиции является частным случаем следующего:

Пусть функция $f(y_1,..,y_m) \in M$ и $\forall i\ (1 \le i \le m)\ g_i(x_1,\dots,x_n) \in M$. Функции, которые будут подставляться в f каждая зависит от своих переменных, но фиктивные переменные не влияют на монотонность. Соответственно, рассмотрим

$$f\big(g_1(x_1,\dots,x_n),\dots,g_m(x_1,\dots,x_n)\big) = \Phi(x_1,\dots,x_n)$$

Докажем, что $\Phi \in M$. Пусть $\tilde{\alpha} = (\alpha_1,\dots,\alpha_n) \in E_2^n$, $\tilde{\beta} = (\beta_1,\dots,\beta_n) \in E_2^n$ и $\tilde{\alpha} \le \tilde{\beta}$.

Обозначим $g_i(\tilde{\alpha}) = \gamma_i,\ g_i\big(\tilde{\beta}\big) = \delta_i$. Тогда так как $\tilde{\alpha} \le \tilde{\beta}$ и $g_i \in M$, то

$\forall i\colon\ \gamma_i = g_i(\tilde{\alpha}) \le g_i\big(\tilde{\beta}\big) = \delta_i$, то есть $\gamma_i \le \delta_i$.

По определению $\tilde{\gamma} = (\gamma_1,\dots,\gamma_m) \le (\delta_1,\dots,\delta_m) = \tilde{\delta}$.

$\Phi(\tilde{\alpha}) = f\big(g_1(\tilde{\alpha}),\dots,g_m(\tilde{\alpha})\big) = f(\gamma_1,\dots,\gamma_m) = f(\tilde{\gamma})$

$\Phi\big(\tilde{\beta}\big) = f\left(g_1\big(\tilde{\beta}\big),\dots,g_m\big(\tilde{\beta}\big)\right) = f(\delta_1,\dots,\delta_m) = f(\tilde{\delta})$

Но $\tilde{\gamma} \le \tilde{\delta}$ и $f \in M$, следовательно $f(\tilde{\gamma}) \le f\big(\tilde{\delta}\big) \Rightarrow \Phi(\tilde{\alpha}) \le \Phi\big(\tilde{\beta}\big) \Rightarrow \Phi \in M$ ■.

Лемма о несамодвойственной функции

Лемма о немонотонной функции.

Пусть $f(x_1,..,x_n)$ – произвольная функция алгебры логики и $f \notin M$. Тогда подставляя в f на места всех переменных 0 и 1, можно получить функцию $\varphi(x) = \overline{x}$.

Пусть $f(x_1,..,x_n)$ – произвольная функция алгебры логики и $f \notin S$. Тогда подставляя в f на места всех переменных x и $\bar{x}$, можно получить функцию $\varphi(x) = const$.

Доказательство. Пусть $f(x_1,..,x_n) \notin S \Rightarrow$ не для всех наборов выполняется $\bar{f}(\overline{x_1},\dots,\overline{x_n}) = f(x_1,\dots,x_n)$.

Следовательно,

$$\exists\ \tilde{\alpha} = (\alpha_1,\dots,\alpha_n)\colon \bar{f}(\overline{\alpha_1},\dots,\overline{\alpha_n}) = f(\alpha_1,\dots,\alpha_n) \Leftrightarrow f(\overline{\alpha_1},\dots,\overline{\alpha_n}) = f(\alpha_1,\dots,\alpha_n).$$

Рассмотрим $\varphi(x) = f(x \oplus \alpha_1, x \oplus \alpha_2, \dots, x \oplus \alpha_n)$:

$\varphi(0) = f(\alpha_1,\dots,\alpha_n),\ \varphi(1) = f(\overline{\alpha_1},\dots,\overline{\alpha_n}) \Rightarrow\ \varphi(0) = \varphi(1) \Rightarrow\ \varphi(x) = const$ и

$x \oplus \alpha = \begin{cases} x, & \text{если } \alpha = 0 \\ \bar{x}, & \text{если } \alpha = 0 \end{cases}$ ■.

Лекция 6. Теорема Поста о полноте.

Лемма о немонотонной функции

Лемма о немонотонной функции.

Пусть $f(x_1,..,x_n)-$ произвольная функция алгебры логики и $f \notin M$. Тогда подставляя в f на места всех переменных 0 и 1, можно получить функцию $\varphi(x)=\overline{x}$.

Доказательство. Пусть $f(x_1,x_2,\dots,x_n) \notin M \Rightarrow$ по определению $\exists\ \tilde{\alpha}=(\alpha_1,\dots,\alpha_n)\in E_2^n$, $\tilde{\beta}=(\beta_1,\dots,\beta_n)\in E_2^n$ и $\tilde{\alpha}\le\tilde{\beta}$, но $f(\tilde{\alpha}) \not\le f(\tilde{\beta}) \Leftrightarrow\ f(\tilde{\alpha})=1, f(\tilde{\beta})=0$.

По определению $\tilde{\alpha}\le\tilde{\beta} \Leftrightarrow \forall i\ (\alpha_i\le\beta_i)$, то есть все $\binom{\alpha_i}{\beta_i}\in\left\{\binom{0}{0}\ \binom{1}{1}\ \binom{0}{1}\right\}$. То есть β_i можно получить из α_i, заменяя какие-то 0 на 1.

Пусть $\tilde{\alpha}$ и $\tilde{\beta}$ различаются ровно в k разрядах $j_1,j_2,\dots,j_k$. Тогда в этих разрядах $\alpha_{j_s}=0$, $\beta_{j_s}=1$. Рассмотрим последовательность наборов: $\tilde{\alpha}=\tilde{\alpha}_1,\tilde{\alpha}_2,\dots,\tilde{\alpha}_k=\beta$, где каждый следующий набор получается из предыдущего заменой одного 0 в позициях (координатах) $j_1,j_2,\dots,j_k$ на 1, при этом $\tilde{\alpha}=\tilde{\alpha}_0<\tilde{\alpha}_1<\tilde{\alpha}_2<\dots<\tilde{\alpha}_k=\beta$ и $f(\tilde{\alpha}_0)=1$, $f(\tilde{\alpha}_k)=0$. Следовательно, $\exists\, t: f(\tilde{\alpha}_t)=1,\ f(\tilde{\alpha}_{t+1})=0$,

$\tilde{\alpha}_t=(\gamma_1,\dots,\gamma_{i-1},0,y_{i+1},\dots,\gamma_n),\ \ \tilde{\alpha}_{t+1}=(\gamma_1,\dots,\gamma_{i-1},1,y_{i+1},\dots,\gamma_n)$

Рассмотрим $f(\gamma_1,\dots,\gamma_{i-1},x,y_{i+1},\dots,\gamma_n)=\varphi(x)$.

$$\begin{matrix}\varphi(0)=f(\tilde{\alpha}_t)=1\\ \varphi(1)=f(\tilde{\alpha}_{t+1})=0\end{matrix}\Rightarrow \varphi(x)=\bar{x}\ \ \blacksquare.$$

Следствие из доказательства. Если $f\notin M$, то $\exists$ 2 соседний набора $\tilde{\alpha}$ и $\tilde{\beta}$, на которых нарушается монотонность.

То есть чтобы определить, монотонная ли функция, достаточно проверить все соседние наборы: если на соседний наборах монотонность нарушается, то функция немонотонна.

Лемма о нелинейной функции

Лемма о нелинейной функции.

Пусть $f(x_1,..,x_n)-$ произвольная функция алгебры логики и $f\notin L$, то, подставляя в нее вместо всех переменных $0,1,x,y,\bar{x},\bar{y}$, можно получить $\varphi(x,y)=xy$ или $\varphi(x,y)=\overline{xy}$.

Доказательство. Рассмотрим полином Жегалкина $f(x_1, \dots, x_n)$ $(\notin L)$: в этом полиноме $\exists$ слагаемые, в которых есть произведение ≥ 2 переменных. Не ограничивая общности будем считать, что есть слагаемое, содержащее $x_1 \cdot x_2 \cdot \dots$. Тогда группируя слагаемые и вынося за скобку можно получить:

$$f(x_1, .., x_n) = x_1x_2P_1(x_3, \dots, x_n) \oplus x_1P_2(x_3, \dots, x_n) \oplus x_2p_3(x_3, \dots, x_n) \oplus P_4(x_3, \dots, x_n)$$

$P_1(x_3, \dots, x_n) -$ нетривиальный полином

Так как $P_1(x_3, \dots, x_n) -$ нетривиальный полином, то существует набор $(\alpha_3, \dots, \alpha_n)$: $P_1(\alpha_3, \dots, \alpha_n) = 1$, что следует о единственности полинома Жегалкина (тк есть 0 соответствует единственный полином).

Рассмотрим $f(x_1, x_2, \alpha_3, \dots, \alpha_n) = x_1 \cdot x_2 \oplus a \cdot x_1 \oplus b \cdot x_2 \oplus c$.

Рассмотрим $f(x \oplus b,\ y \oplus a, \alpha_3, \dots, \alpha_n) = (x \oplus b)(y \oplus a) \oplus a(x \oplus b) \oplus b(y \oplus a) \oplus c =$
$xy \oplus ax \oplus by \oplus ab \oplus ax \oplus ab \oplus by \oplus ab \oplus c = \begin{cases} xy, \text{если } ab \oplus c = 0 \\ \overline{xy}, \text{если } ab \oplus c = 1 \end{cases}$ ■.

Теорема Поста о полноте

Теорема (Поста о полноте). Множество функций алгебры логики *A* образует *полную систему* тогда и только тогда, когда оно не содержится целиком ни в одном из 5 классов T_0, T_1, L, S, M.

Доказательство.

1) Пусть N – один из классов T_0, T_1, L, S, M и $A \subseteq N \Rightarrow [A] \subseteq [N] = N \neq P_2 \Rightarrow [A] \neq P_2 \Rightarrow A -$ не полная система.
2) Пусть *A* не содержится целиком ни в одном из классов T_0, T_1, L, S, M.
   Тогда в $A\ \exists\ f_0 \notin T_0,\ f_1 \notin T_1,\ f_L \notin L,\ f_M \notin M,\ f_S \notin S$.
   Покажем, что формулами над *A* можно получить все функции из P_2, то есть, что $A -$ полная система. Покажем, что это можно сделать, используя только функции $f_0,\ f_1,\ f_L,\ f_M,\ f_S$.

   А) Получение $\bar{x}$
   Рассмотрим $f_0(x, x, \dots, x) = \varphi_0(x)$. $f_0 \notin T_0 \Rightarrow f_0(0,0, \dots, 0) = 1$.
   Тогда $\varphi_0(0) = f_0(0,0, \dots, 0) = 1 \Rightarrow \varphi_0(x) = \bar{x}$ или $\varphi_0(x) \equiv 1$.
   Рассмотрим $f_1(x, x, \dots, x) = \varphi_1(x)$. $f_1 \notin T_1 \Rightarrow f_1(1,1, \dots, 1) = 0$.
   Тогда $\varphi_1(1) = f_1(1,1, \dots, 1) = 0 \Rightarrow \varphi_1(x) = \bar{x}$ или $\varphi_0(x) \equiv 0$.
   Если $\varphi_0(x) = \bar{x}$ или $\varphi_1(x) = \bar{x}$, то $\bar{x}$ получили, иначе получили обе константы $\varphi_0(x) \equiv 1$ и $\varphi_1(x) = \bar{x}$. Тогда по *лемме о немонотонной*

функции подставляя в функцию f_M 0, 1 и x, можем получить формулу, задающую $\bar{x}$. Следовательно, в любом случае можем построить $\bar{x}$.

Б) Получение константы
По *лемме о несамодвойственной функции*, подставляя в $f_S \notin S$ x и $\bar{x}$, мы можем получить одну из констант.
Подставляя ее в $\bar{x}$, получим вторую константу.

В) Получение конъюнкции
По лемме о нелинейной функции, подставляя в $f_L \notin L$ 0, 1, $x, y, \bar{x}, \bar{y}$, можем получить $\varphi(x, y) = xy$ или $\varphi(x, y) = \overline{xy}$.
В последнем случае подставляем $\overline{xy}$ в $\bar{x}$, тем самым получим xy.

Так как $\{xy, \bar{x}\}$ – полная система, то далее можем получить все функции алгебры логики $\Rightarrow A$ – полная система ■.

Базис

Опр. Множество функций алгебры логики называется *базисом* (в P_2), если

1) A – полная система ($[A] = P_2$)
2) Для $\forall f \in A$: $A\backslash\{f\}$ – не полная система ($[A\backslash\{f\}] \neq P_2$)

Если полная система не базис, то она избыточна, и из нее можно выбросить некоторые функции, чтобы она стала базисом, при этом полнота сохранится.

Теорема. Максимальное число функций в базисе в P_2 равно 4.

Доказательство.

1. Докажем, что изо всякой полной системы можно выделить подмножество из не более 4 функций, которые также будут образовывать полную систему.
   Пусть A – полная система.
   По *теореме Поста* в A $\exists$ $f_0 \notin T_0,\ f_1 \notin T_1,\ f_L \notin L,\ f_M \notin M,\ f_S \notin S$
   $\Rightarrow \{f_0,\ f_1,\ f_L,\ f_M,\ f_S\}$ – полная система. Рассмотрим $f_0(x_0, \dots, x_n) \notin T_0 \Rightarrow f_0(0,0, \dots, 0) = 1$.
   А) Пусть $f_0(1,1, \dots, 1) = 0 \Rightarrow f_0 \notin M,\ f_0 \notin T_1 \Rightarrow \{f_0, f_L, f_S\}$ – полная система по теореме Поста.

Б) Пусть $f_0(1,1,\dots,1) = 1 \Rightarrow f_0 \notin S \Rightarrow \{f_0, f_1, f_L, f_M\}$ – полная система по теореме Поста.

Таким образом, если в полной системе функций ≥ 5, то она *не базис*. Следовательно, в любом базисе не более 4 функций.

2. Покажем, что существуют базисы из 4 функций.
   Рассмотрим $A = \{0, 1, xy, x \oplus y \oplus z\}$.
   $1 \notin T_0,\ 0 \notin T_1,\ xy \notin L,\ x \oplus y \oplus z \notin M$ $\left(\text{так как } \begin{Bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{Bmatrix} \begin{matrix} f = 1 \\ f = 0 \end{matrix}\right)$
   Следовательно, A – полная система по теореме Поста.
   $\{1,\ xy,\ x \oplus y \oplus z\} \subseteq T_1$
   $\{0,\ xy,\ x \oplus y \oplus z\} \subseteq T_0 \Rightarrow A$ – базис ■.
   $\{1,\ 0,\ x \oplus y \oplus z\} \subseteq L$
   $\{1,\ 0,\ xy\} \subseteq M$

Другие примеры базисов:

$\{x|y\},\ \{xy, \bar{x}\},\ \{xy,\ x \oplus y, 1\}$

Предполный класс

Опр. Множество функций алгебры логики A называется *предполным классом*, если

1) A – не полная система ($[A] \neq P_2$)
2) Для $\forall f \notin A$ система $A \cup \{f\}$ является полной ($[A \cup \{f\}] = P_2$)

Теорема. В P_2 $\exists$ 5 предполных классов, а именно: T_0, T_1, L, S, M.

Лекция 7. Обобщение алгебры логики. K-значная логика.

Предполный класс

Опр. Множество функций алгебры логики A называется *предполным классом*, если

1) $A -$ не полная система ($[A] \neq P_2$)
2) Для $\forall f \notin A$ система $A \cup \{f\}$ является полной ($[A \cup \{f\}] = P_2$)

Теорема. В P_2 $\exists$ 5 предполных классов, а именно: T_0, T_1, L, S, M.

Доказательство.

1) Докажем, что никакой из классов T_0, T_1, L, S, M не содержится в другом классе. Построим таблицу:

$\in$ \ $\notin$	T_0	T_1	L	M	S
T_0		0	xy	$x \oplus y$	0
T_1	1		xy	$x \oplus y \oplus 1$	1
L	1	0		$x \oplus y$	0
M	1	0	xy		0
S	$\bar{x}$	$\bar{x}$	$m(x,y,x)$	$\bar{x}$	

$m(x,y,x) = xy \vee yz \vee zx = xy \oplus xz \oplus xz$
$x \oplus y \oplus 1 = x \sim y$

2) Пусть N – один из классов T_0, T_1, L, S, M. Тогда $[N] = N \neq P_2 \Rightarrow N -$ не полная система. Пусть $f \notin N \Rightarrow N \cup \{f\}$ не содержится целиком ни в одном из 5 классов T_0, T_1, L, S, M, так как N не содержится в 4 классе, кроме N (пункт 1) и f не содержится в $N \Rightarrow N \cup \{f\} -$ полная система (по теореме Поста) ⇒ все 5 классов T_0, T_1, L, S, M являются предполными.

3) Докажем, что в P_2 нет других предполных классов. Пусть A – предполный класс, следовательно A – не полная система. По теореме Поста $\exists N \in \{T_0, T_1, L, S, M\}$, $A \subseteq N$. Докажем (от противного), что $A = N$.

Допустим, что $A \neq N \Rightarrow A \subset (\text{или} \neq) N \Rightarrow \exists f: f \in N,\ f \notin A \Rightarrow A \cup \{f\} \subseteq N$.

Тогда по теореме Поста $A \cup \{f\} -$ не полная система $\Rightarrow$ противоречие с тем, что $A -$ предполный класс (с пунктом 2) $\Rightarrow A = N$ ■.

Обобщение алгебры логики. К-значная логика.

В качестве основного множества для k-значной логики будем рассматривать множество:

$$E_k = \{0, 1, 2, \dots, k-1\}, \qquad k \in \mathbb{N}, \qquad k \geq 2$$

Опр. Функцией k-значной логики называется любое отображение $f(x_1, \dots, x_n)$: $E_k^n \to E_k$

Мощность $|E_k| = k^n$

Если рассмотреть все функции *k-значной логики от n переменных*, то мощность:

$|P_k^n| = k^{k^n}$

Примеры функций:

- Константы 0, 1, 2, …, k-1
- $\max(x, y)$, $\min(x, y)$
- $J_\sigma(x) = \begin{cases} k-1, & \text{если } x = \sigma \\ 0, & \text{если } x \neq \sigma \end{cases}$, $\sigma = 0,1, \dots, k-1$

Теорема. Множество функций в P_k

$$\{0,1, \dots, k-1, \max(x, y), \min(x, y), J_0(x), J_1(x), \dots., J_{k-1}(x)\}$$

является полной системой в P_k.

Доказательство. $\max(x_1, x_2, x_3) = \max(\max(x_1, x_2), x_3)$

Соответственно, $\max(x_1, \dots, x_{n-1}, x_n) = \max(max(x_1, \dots, x_{n-1})), x_n)$. Таким образом, индуктивно можно построить все функции максимум от любого числа переменных. Аналогично для min. Теперь докажем, что для любой функции $f(x_1, \dots, x_n) \in P_k$ верно представление (первая форма):

$$f(x_1, \dots, x_n) = \max_{(\sigma_1, \dots, \sigma_n) \in E_k^n} \{\min(J_{\sigma_1}(x_1), \dots, J_{\sigma_n}(x_n), f(\sigma_1, \dots, \sigma_n))\}$$

Пусть $\tilde{\alpha} = (\alpha_1, \dots, \alpha_n) \in E_k^n -$ любой набор. Тогда правая часть на наборе $\tilde{\alpha}$ равна:

$$\max_{(\sigma_1, \dots, \sigma_n) \in E_k^n} \{\min\left(J_{\sigma_1}(\alpha_1), \dots, J_{\sigma_n}(\alpha_n), f(\sigma_1, \dots, \sigma_n)\right)\}$$

Если хотя бы одно $\sigma_i \neq \alpha_i$, то соответствующий min = 0, так как $J_{\sigma_i}(\alpha_i) = 0$.

Тогда

$$\max_{(\sigma_1,\dots,\sigma_n)\in E_k^n}\{\min\left(J_{\sigma_1}(\alpha_1),\dots,J_{\sigma_n}(\alpha_n),f(\sigma_1,\dots,\sigma_n)\right)\} =$$
$$= \max\{0,0,\dots,0,\min(J_{\alpha_1}(\alpha_1),\dots,J_{\alpha_n}(\alpha_n),f(\alpha_1,\dots,\alpha_n))\} =$$
$$= \min\left(J_{\alpha_1}(\alpha_1),\dots,J_{\alpha_n}(\alpha_n),f(\alpha_1,\dots,\alpha_n)\right) =$$
$$= \min\left(k-1,k-1,\dots,k-1,f(\alpha_1,\dots,\alpha_n)\right) = f(\alpha_1,\dots,\alpha_n)$$

Это есть левая часть на наборе $\tilde{\alpha}$ ■.

Сложение и умножение по модулю k

$x + y(mod\ k)$ – сложение по модулю k

$x \cdot y\ (mod\ k)$ – умножение по модулю k

Обе операции коммутативны, ассоциативны и есть дистрибутивность справа и слева:

$$(a \oplus b) \cdot c = (a \cdot c) \oplus (b \cdot c)$$

$\bar{x} = x \oplus 1\ (mod\ k) -$ отрицание Поста

Если $x \in (0; k-2)$, то $\bar{x} = x \oplus 1$, если $x = k-1$, то $\bar{x} = 0$.

$\sim x = (k-1) - x$ – отрицание Лукасевича (Лукашевича)

$$j_\sigma(x) = \begin{cases} 1, & x = \sigma \\ 0, & x \neq \sigma \end{cases}$$

Теорема. Для $\forall f(x_1,..,x_n)$ из P_k верно представление (вторая форма):

$$f(x_1,\dots,x_n) = \sum_{(\sigma_1,\dots,\sigma_n)\in E_k^n} j_{\sigma_1}(x_1)\cdot j_{\sigma_2}(x_2)\cdot\dots\cdot j_{\sigma_n}(x_n)\cdot f(\sigma_1,\dots,\sigma_n)$$

$\sum -$ сумма по модулю k

Доказательство. Пусть $\tilde{\alpha} = (\alpha_1,\dots,\alpha_n) \in E_k^n -$ любой набор. Тогда правая часть выражения на этом наборе равна:

$$\sum_{(\sigma_1,\dots,\sigma_n)\in E_k^n} j_{\sigma_1}(\alpha_1)\cdot j_{\sigma_2}(\alpha_2)\cdot\dots\cdot j_{\sigma_n}(\alpha_n)\cdot f(\sigma_1,\dots,\sigma_n)$$

Если хотя бы одно $\sigma_i \neq \alpha_i$, то соответствующий произведение = 0, так как $j_{\sigma_i}(\alpha_i) = 0$. Поэтому

$$\sum_{(\sigma_1,\dots,\sigma_n)\in E_k^n} j_{\sigma_1}(\alpha_1)\cdot j_{\sigma_2}(\alpha_2)\cdot \dots \cdot j_{\sigma_n}(\alpha_n)\cdot f(\sigma_1,\dots,\sigma_n) =$$

$$= 0+0+\dots+0+j_{\sigma_1}(\alpha_1)\cdot j_{\sigma_2}(\alpha_2)\cdot \dots \cdot j_{\sigma_n}(\alpha_n)+0+0+\dots+0 =$$

$$= 1\cdot 1\cdot \dots \cdot 1\cdot f(\alpha_1,\dots,\alpha_n) = f(\alpha_1,\dots,\alpha_n)$$

Это есть левая часть на наборе $\tilde{\alpha}$ ■.

Из этой теоремы следует, что система, состоящая из всех j, произведения по модулю k и константы, является полной системой.

Для целых чисел $a = b(mod\ k)$ означает, что при делении на k a и b дают равные остатки.

Например, $7 = -5(mod\ 4)$, так как $-5 = -8+3,\ 7 = 4+3$.

Также $a = b(mod\ k) \Leftrightarrow a - b \vdots k$

Лекция 8. Вычисления по модулю k. Теория графов.

Вычисления по модулю k

$\mathbb{Z}-$ множество целых чисел, $k \in \mathbb{N}\ k \geq 2$

Говорят, что a делится на k:

$$a \vdots k \Leftrightarrow \exists\, n \in \mathbb{Z}\,(a = nk)$$

Утверждение. $\forall a \in \mathbb{Z}\ \ \exists\, n, r \in \mathbb{Z}\ \ a = nk + r,\ \ 0 \leq r \leq k-1$

$n-$ частное

$r-$ остаток от деления a на k

Например, $-5 = (-2)\cdot 4 + 3$, то есть остаток от деления -5 на 4 равен 3.

Опр. $a = b(mod\ k) \Leftrightarrow$ у a и b одинаковые остатки при делении на k.*Y*

Например, $-1 = k - 1(mod\ k)$, так как $-1 = (-1)\cdot k + (k-1)$

Также $a = b(mod\ k) \Leftrightarrow a - b \vdots k$

Утверждение. (для целых чисел)

$$\begin{matrix} a = b(mod\ k) \\ c = d(mod\ k) \end{matrix} \Rightarrow \begin{matrix} a + c = b + d\ (mod\ k) \\ ac = bd\ (mod\ k) \end{matrix}$$

Доказательство.

1) $(a + c) - (b + d) = (a - b) + (c - d)$, причем $(a - b) \vdots k$ и $(c - d) \vdots k$, так как по условию a и b, c и d сравнимы по модулю $k \Rightarrow \vdots k$.

2) Аналогично:

$$\begin{matrix} a = b + nk \\ c = d + mk \end{matrix} \Rightarrow ac = \underbrace{bd + bmk + dnk + nmk^2}_{\vdots k} \Rightarrow ac - bd \vdots k \Rightarrow ac = bd\ (mod\ k)\ \blacksquare.$$

Опр. Полиномом в k-значной логике называется сумма одночленов (по $mod\ k$), где одночлен – это произведение (по $mod\ k$) степеней переменных и коэффициента. Подобные одночлены являются приведенными.

Теорема. Если k −составное число, то не все формулы из P_k представимы полиномами.

Если k −простое число, то каждая функция из P_k представима полиномом.

Доказательство.

1) Пусть k −составное число $\Rightarrow k = k_1 k_2$, где $2 \le k_1 \le k_2 \le k,\ \ k_1, k_2 \in \mathbb{N}$.

Докажем, что тогда функция $j_0(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0 \end{cases}$ не представима полиномом.

(От противного). Докажем, что $j_0(x)$ представима полиномом:

$$j_0(x) = a_0 + a_1 x + a_2 x^2 + \cdots a_m x^m (\text{операции по } mod\ k)$$

Тогда $j_0(0) = a_0 = 1 \ \Rightarrow a_0 = 1 \ \Rightarrow$

$$j_0(x) = 1 + a_1 x + a_2 x^2 + \cdots a_m x^m (\text{операции по } mod\ k)$$

$j_0(k_1) = 0 \Rightarrow$

$$\underbrace{1 + a_1 \cdot k_1 + a_2 \cdot k_1^2 + \cdots + a_m k_1^m}_{\text{операции обычные}} \vdots\ k(= k_1 k_2) = 1 + a_1 k_1 + a_2 k_1^2 + \cdots + a_m k_1^m \vdots\ k_1 \quad (1)$$

и $\ a_1 k_1 + a_2 k_1^2 + \cdots + a_m k_1^m \vdots\ k_1 \ \ (2)$.

Тогда вычтя (2) из (1) получим:

$1 \vdots\ k_1$ − противоречие.

Следовательно, от противного, $j_0(x)$ не представима полиномом.

2) Пусть k −простое число.

Лемма (малая теорема Ферма).

Пусть k −простое число, и пусть $a \in \mathbb{Z}$ и $a \not\vdots\ k$. Тогда $a^{k-1} \vdots\ k$.

Доказательство. Рассмотрим числа $a \cdot 1, a \cdot 2, \ldots, a \cdot (k-1)$ (умножение обычное).

Докажем, что все $k-1$ чисел дают при делении на k разные остатки:

рассмотрим $a \cdot m_1$ и $a \cdot m_2$, где $m_1 \neq m_2$ и $1 \le m_1 \le m_2 \le k-1$

$\Rightarrow a \cdot m_2 - a \cdot m_1 = a \cdot (m_2 - m_1),\ \ 1 \le m_2 - m_1 \le k-2$, так как $k-$ простое, то

$m_2 - m_1$ не имеет общих множителей с k. По условию a тоже не имеет общих множителей с k, следовательно $a \cdot (m_2 - m_1) \not\vdots\ k \Rightarrow$ у am_1 и am_2 разные остатки при делении на k.

Тогда среди остатков от деления чисел $a \cdot 1, a \cdot 2, \dots, a \cdot (k-1)$ на k каждый остаток $1,2,3, \dots, k-1$ встречается ровно 1 раз.

Поэтому $a \cdot 1, a \cdot 2, \dots, a \cdot (k-1) = 1 \cdot 2 \cdot 3 \cdot \dots \cdot k-1 \ (mod\ k)$

$$\underbrace{a^{(k-1)} \cdot 1 \cdot 2 \cdot 3 \cdot \dots \cdot (k-1)}_{\text{не имеет общих множителей с } k} \cdot (a^{k-1} - 1) \vdots k$$

то есть $\cdot (a^{k-1} - 1) \vdots k$ ■.

Рассмотрим полином $1 - x^{k-1} = 1 + (k-1) \cdot x^{k-1}$ (в P_k) $= \begin{cases} 1 & x = 0 \\ 0 & \underbrace{x \neq 0}_{1 \le x \le k-1} \end{cases} = j_0(x)$ по малой теореме Ферма.

Выразим $j_\sigma(x) = j_0\left(\underbrace{x - \sigma}_{\text{в } P_k}\right) = 1 - \underbrace{(x-\sigma)^{k-1}}_{\text{в } P_k \ -1=k-1}$ – полином. Так как каждая функция из P_k представима *второй формой*, а в ней участвуют только константы, сложение и умножение по $mod\ k$ и все $j_0(x), j_1(x), \dots, j_{k-1}(x)$, то заменяя функции $j_0(x), j_1(x), \dots, j_{k-1}(x)$ их представлениями полиномом и упрощая выражение, получим представление функции полиномом (по $mod\ k$) ■.

Примечание: $x^k = x$, так как $x^{k-1} = 1$ по малой теореме Ферма.

Особенности k-значной логики

Особенности в P_k при $k \geq 3$:

1) Полиномы
2) При $k = 2$ замкнутых классов счетное множество (Пост описал структуру всех замкнутых классов, построив из 8 основных цепочек и еще некоторые классы).
   При $k \geq 3$ замкнутых классов континуум
3) При $k = 2$ в ∀ замкнутом классе ∃ конечный базис.
   При $k \geq 3$ ∃ замкнутый класс с бесконечным базисом, а также ∃ замкнутый класс без базиса.

Возвращаясь к малой теореме Ферма, дополнительно можно рассмотреть книгу «Великая теорема Ферма», Саймон Синх про историю развитию математики и теорему Ферма.

Теория графов

Опр. Графом называется пара $G = (V, E)$, где V – множетсов элементов, E – множество неупорядоченных пар элементов из V. При этом считается, что элементы в паре разные и пары не повторяются. *Элементы* из V называют вершинами графа, а пары из E – *ребра*.

Опр. Если пары рассматриваются как упорядоченные, то граф называется *ориентированным*, а пары называют *дугами* (ориентированными ребрами).

Опр. Если пары могут повторяться, то граф называется *мультиграфом*. Если дополнительно в парах могут быть одинаковые элементы, то называется граф называется *псевдографом,* а такие пары называют *петлями.*

Например, псевдограф из 7 вершин изображается следующим образом (рис. 8.1):

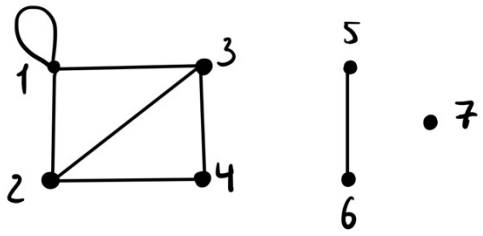

Рис. 8.1 Псевдограф из 7 вершин

Чтобы изобразить ориентированный граф, на ребрах помечают стрелки, показывающие направление.

Опр. Две вершины графа называют *смежными*, если они входят в одно и то же ребро (соединены ребром).

Опр. Говорят, что вершина и ребро *инцидентны*, если вершина входит в данное ребро (как в пару).

Опр. Степенью вершины v ($\deg v$) в неориентированном графе называется число ребер, инцидентных вершине v. При этом в псевдографе петля в вершине учитывается в степени дважды.

То есть фактически считаем сколько ребер исходит из вершины. Так на рис. 8.1. из вершины 2 исходит 3 ребра, соответственно $\deg 2 = 3$. А у вершины 1, из которой исходит петля, степень равна 4.

Лекция 9. Графы.

Утверждение. (о сумме степеней всех вершин графа).

В любом графе (псевдографе) выполняется равенство:

$$\sum_{i=1}^{p} \deg v_i = 2q,$$

где $\deg v_i -$ степень вершины v_i, $p -$ число вершин в графе, $q -$ число ребер в графе.

Доказательство. При подсчете степени вершины мы считаем количество исходящих из нее ребер. Вычисляя сумму всех степеней, мы получаем, что каждое ребро считается дважды, так как оно инцидентно двум вершинам (петли по определению степени вершины также посчитаются дважды). Поэтому $\sum_{i=1}^{p} \deg v_i = 2q$ ■.

Рассмотрим пример задачи.

Вопрос: можно ли построить граф с 77 вершинами степени 15?

Ответ: $p = 77,\ \forall i\ (\deg v_i = 15)$, $\sum_{i=1}^{p} \deg v_i = 15 \cdot 77 = 2q$. Но такого q не существует, соответственно такого графа не существует.

Способы задания графов

1) Список вершин и список ребер
2) Списки смежности – для всех p вершин задаем списки вершин, с которыми они соединены:
   $v_1: v_{i_1}, v_{i_2}, \dots, v_{i_k}$
   $v_2: v_{j_1}, v_{j_2}, \dots, v_{j_s}$
   ...
   $v_p: v_{r_1}, v_{r_2}, \dots, v_{r_l}$
3) Матрицы смежности

Опр. Пусть $G -$ граф с p вершинами $v_1, v_2, \dots, v_p$. Тогда *матрицей смежности графа* G называется матрица размера $p \times p$ с элементами 0 и 1, где

$$a_{ij} = \begin{cases} 1, \text{если } (v_i, v_j) - \text{ребра графа } G \\ 0, \qquad\qquad \text{в ином случае} \end{cases}$$

Ориентированные графы, мультиграфы и псевдографы также можно задавать матрицей смежности. При этом если в матрице смежности обычного графа на диагонали стоят 0, то в матрице псевдографа на диагонали будут уже 1. При задании

матрицы ориентированного графа будет учитываться порядок вершин. Например, если ребро v_{ij}, то $a_{ij} = 1$, но $a_{ji} = 0$. Также если граф неориентированный, то матрица симметрична.

Изоморфизм графов

Опр. Граф $G_1 = (V_1, E_1)$ называется *подграфом* графа $G = (V, E)$, если $V_1 \subseteq V,\ E_1 \subseteq E$.

Опр. Подграф называется остовным в G, если $V_1 = V$.

То есть остовные подграфы получаются удалением части ребер при сохранении вершин. Сам граф является своим остовным графом.

Опр. Два графа (или псевдографа) $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называются *изоморфными*, если существует взаимно однозначное отображение $\varphi: V_1 \to V_2$ такое, что

$$\forall v_i, v_j \in V_1 \ \left(\varphi(v_i), \varphi\left(v_j\right)\right) \in E_2 \Leftrightarrow (v_i, v_j) \in E_1$$

Рассмотрим два графа на рис. 9.1. Изоморфны ли они?

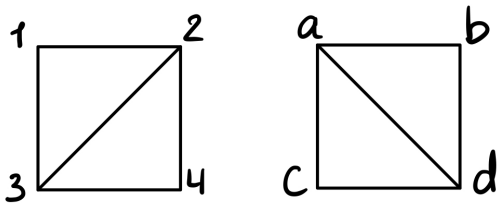

Рис. 9.1. Два изоморфных графа

Попробуем сопоставить вершины так, чтобы степени и расположение сохранялось:

$1 \leftrightarrow b,\ 2 \leftrightarrow a,\ 3 \leftrightarrow d,\ 4 \leftrightarrow c$. Соответственно, графы изоморфны.

При проверке на изоморфизм двух графов стоит сначала проверить, что

1) Число вершин совпадает
2) Число ребер совпадает
3) Степени вершин совпадают

то есть их структура полностью совпадает.

Опр. Путем в графе (или псевдографе) $G = (V, E)$ называется любая последовательность вида:

$$v_0, (v_0, v_1), v_1, (v_1, v_2), v_2, \dots, v_{n-1}, (v_{n-1}, v_n), v_n$$

$n -$ *длина* пути, путь из v_0 в v_n.

В частности, $v -$ путь длины 0.

Опр. Путь, в котором $v_0 \neq v_n$ и ребра не повторяются называют *цепью.*

Опр. Цепь, в которой не повторяются вершины, называется *простой цепью*.

Утверждение. Любой путь из v_0 в v_n, где $v_0 \neq v_n$ содержит подпуть из v_0 в v_n, который является простой цепью.

Доказательство. Пусть в графе $G = (V, E)$ задан путь L из v_0 в v_n, где $v_0 \neq v_n$. Если в L_1 вершины не повторяются, то $L_1 -$ искомый путь.

Иначе если $L_1 = v_0 C_1 v C_2 v C_3 v_n$, то L_1 содержит подпуть из v_0 в v_n $L_2 = v_0 C_1 v C_3 v_n$. Если в L_2 вершины не повторяются, то $L_2 -$ искомый подпуть, иначе можно выбросить еще кусок пути, получив подпуть из v_0 в v_n. Далее рекурсивно. Процесс должен завершиться, поскольку исходный путь конечен ■.

Опр. Граф $G = (V, E)$ называется *связным,* если для любых двух вершин графа v_i и v_j существует путь из v_i в v_j.

Опр. Пусть $G = (V, E)$. Введем на множестве V *бинарное отношение*

$$v_i \to v_j \equiv (\exists \text{ путь в графе } G \text{ из } v_i \text{ в } v_j).$$

Тогда для этого отношения выполняются свойства:

1) $\forall v_i \in V \quad v_i \to v_i$ – рефлексивность
2) $\forall v_i, v_j, v_k \in V \quad (v_i \to v_j) \& (v_j \to v_k) \Rightarrow (v_i \to v_k)$ –транзитивность
3) $\forall v_i, v_j \in V \quad (v_i \to v_j) \Rightarrow (v_j \to v_i)$ –симметричность

Тогда $v_i \to v_j$ – *отношение эквивалентности*, и V распадается на непересекающиеся классы эквивалентности.

При этом $V = V_1 \cup V_2 \cup \ldots \cup V_k$ так, что любые две вершины из одного класса удовлетворяют $v_i \to v_j$ и для любых двух вершин из разных классов $v_i \nrightarrow v_j$ (рис. 9.2).

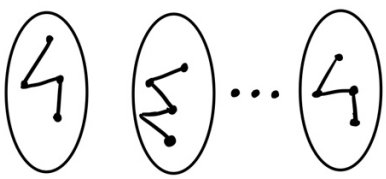

Рис. 9.2. Класс эквивалентности

Каждая часть, изображённая на рис. 9.2, является связным подграфом и называются *связной компонентой.*

Опр. Путь называется *замкнутым*, если $v_n = v_0$. В частности путь из одной вершины замкнутый.

Опр. Замкнутый путь, в котором ребра не повторяются называется *циклом.* Если при этом и вершины не повторяются (кроме $v_n = v_0$), то называется *простым циклом.*

Деревья

Опр. *Деревом* называется любой связный граф без циклов.

Опр. Подграф $G_1 = (V_1, E_1)$ называется *остовным деревом*, если $V_1 = V$ и $G_1 -$ дерево.

Лемма 1. Пусть $G = (V, E) -$ связный граф и пусть ребро $e = (v_i, v_j) \in E$, причем e входит хотя бы в один цикл в графе G. Тогда граф $G\backslash\{e\}$ остается связным.

Доказательство.

При выбрасывании ребра e из графа G вершины v_i, v_j все равно остаются в одной и той же связной компоненте, поскольку из v_i в v_j можно пройти по оставшейся части цикла (см. рис. 9.3.) ■.

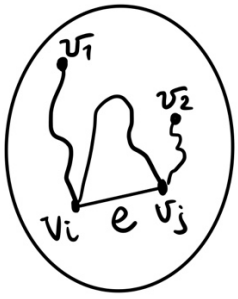

Рис. 9.3. Граф G с ребром e в цикле

Теорема. В любом связном графе $G = (V, E)$ существует хотя бы один подграф, являющийся остовным деревом.

Доказательство. Если в G нет циклов, то G сам является остовным деревом в графе G. Иначе в графе G есть хотя бы один цикл, и следователь существует ребро e_1, входящее в цикл. Тогда выбросим ребро e_1 и по *лемме 1* получим остовный подграф G, который останется связным. Если в G_1 нет циклов, то $G_1 -$ искомое остовное дерево. Иначе в G_1 существует ребро e_2, входящее в цикл. Выбросим его и получим связный остовный

подграф G_2. Если в нем нет циклов, то G_2 – искомое остовное дерево. Иначе… Процесс обязан закончиться, так как G – конечный граф ■.

Дополнительно

Сколько остовных деревьев в графе? Чтобы ответить на этот вопрос, необходимо рассмотреть матрицу смежности для графа, домножить ее на -1, на диагонали поставить степени вершин. Алгебраическое дополнение к каждому диагональному элементу будет равно количеству остовных деревьев в графе.

Лемма 2. Если любому связному графу добавить новое ребро на тех же вершинах, то образуется хотя бы один *простой цикл.*

Доказательство. Пусть v_i, v_j – вершины в связном графе $G = (V, E)$ $(v_i, v_j) \notin E, v_i \neq v_j$. Тогда существует путь из v_i в v_j в графе G и существует простая цепь из v_i в v_j. Тогда при добавлении ребра (v_i, v_j) образуется простой цикл из этой простой цепи и ребра (v_j, v_i) (рис. 9.4) ■.

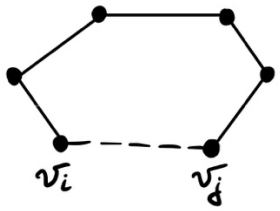

Рис. 9.4. Замыкание простой цепи в простой цикл

Лемма 3. Если в графе $G = (V, E)$ p вершин и q ребер, то в нем $\geq p - q$ связных компонент. При это если в графе G нет циклов, то в графе G ровно $p - q$ связных компонент.

Доказательство. Будем строить граф G постепенно, начиная с p вершин и добавляя на каждом шаге одно ребро. Как ведет себя число связных компонент? В исходном графе (без ребер) p связных компонент. При добавлении одного ребра число связных компонент либо не изменяется (рис. 9.5)., либо уменьшается на 1 (рис. 9.6).

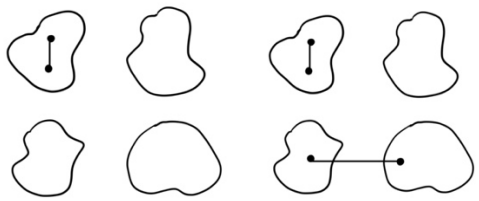

Рис. 9.5. Ребро в одной связной компоненте

Рис. 9.6. Ребро, соединяющее 2 связные компоненты

При добавлении всех q ребер число компонент число компонент уменьшится не более, чем на q. Следовательно, станет $\geq p - q$ связных компонент.

Если в G нет циклов, то не может добавляться ребро с обоими концами в одной компоненте (рис. 9.5), иначе по *лемме 2* в этой компоненте возникнет цикл. Следовательно, при каждом добавлении ребра число связных компонент обязательно уменьшается на 1. В конце останется ровно $p - q$ связных компонент ■.

Лекция 10. Деревья.

Теорема. (об эквивалентных определениях дерева)

Следующие классы графов G совпадают (p - число вершин, q – число ребер):

1) G – дерево
2) G без циклов и $q=p-1$
3) G – связный и $q=p-1$
4) G - связный, но при удалении любого ребра становится несвязным
5) G без циклов, но при добавлении любого нового ребра на тех же вершинах появляется цикл

Доказательство. Докажем $1) \to 2) \to 3) \to 4) \to 5) \to 1)$

$1) \to 2)$: «Без циклов» переносится автоматически по определению дерева. По лемме 3 в графе G связных компонент в точности $p - q$, так как G без циклов. Так как G – связный $p - q = 1 \Rightarrow q = p - 1$.

$2) \to 3)$: G без циклов $\Rightarrow$ по лемме 3 число связных компонент $p - q = p - (p - 1) = 1 \Rightarrow G -$ связный.

$3) \to 4)$: Если в G удалить любое ребро, то останется ребер $q' = p - 2$ и по лемме 3 число связных компонент будет $\geq p - q' = p - (p - 2) = 2 \Rightarrow G -$ не связный.

$4) \to 5)$:

а) Если бы в G был цикл, то удаляя из G ребро из цикла, получили бы по лемме 1 связный граф – противоречие с 4) $\Rightarrow$ в G нет циклов.

б) По лемме 2 при добавлении ребра появится цикл.

$5) \to 1)$: «Без циклов» переносится автоматически. Допустим, что $G -$ не связный, следовательно у него есть как минимум 2 компоненты (рис. 10.1). Возьмем вершины u и v в разных компонентах и добавим ребро (u, v). Очевидно, что через (u, v) не проходит цикл – противоречие с 5) $\Rightarrow$ (от противного) $G -$ связный ■.

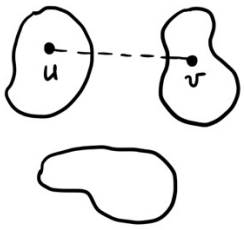

Рис. 10.1. Граф G

Кратчайшее остовное дерево (КОД)

Нужно построить алгоритм для решения следующей задачи:

Вход: полный граф (все ребра между всеми вершинами) K_n на n вершинах

Каждому ребру e сопоставлен вес $w(e)$. Вес любого поддерева – сумма весов его ребер (все веса неотрицательны).

Требуется: найти в K_n остовное дерево* Dс минимальным весом.

*Всего остовных деревьев n^{n-2} для *полного* дерева K_n (у которого n вершин и $n-1$ ребер).

Алгоритм построения КОД.

Пусть $V -$ множество вершин графа K_n. Строим подграф $(V, E) = D$ по шагам. в начале $E = \emptyset$.

1) Выбираем ребро наименьшего веса (если его нет, то берем любое) e_1 и заносим его в E.
2) Пусть в E уже занесены $e_1, e_2, \dots, e_{k-1}$.
   Если $k - 1 = n - 1$, то STOP.
   Иначе: в E заносим ребро e_k, имеющее минимальный вес среди всех ребер, не образующих циклов с уже выбранным ребрами $e_1, e_2, \dots, e_{k-1}$.

Теорема. Описанный алгоритм построения КОД корректно строит кратчайшее остовное дерево.

Доказательство.

1) Если $k - 1 \le n - 2$, то граф $(V, \{e_1, e_2, \dots, e_{k-1}\})$ имеет по лемме 3 связных компонент $\ge n - (n-2) = 2 \Rightarrow$ уже построенный граф - не связный $\Rightarrow$ $\exists$ ребро, не образующее циклов с $e_1, e_2, \dots, e_{k-1}$ $\Rightarrow$ при $k - 1 \le n - 2$ пункт 2 алгоритма сработает. Тогда алгоритм остановится по оператору STOP.
2) Алгоритм построить подграф $D = (V, \underbrace{\{e_1, e_2, \dots, e_{k-1}\}}_{E})$.

   В D по построению нет циклов и $p = n,\ q = n - 1 \Rightarrow$ по теореме $D -$ дерево, причем остовное, поскольку построено на всяких вершинах.
3) Докажем, что $D -$ кратчайшее остовное дерево (от противного).
   Допустим, что $D -$ не кратчайшее остовное дерево $\Rightarrow$ рассмотрим все кратчайшие остовные деревья в K_n и выберем среди них то дерево K, которое имеет с деревом D максимальное число общих ребер $(K \ne D)$. Так как в K и в D по $n - 1$ ребер, то $\exists$ ребро в D, которого нет в K. Ребра в D включались в порядке $e_1, e_2, \dots, e_{n-1}$. Пусть $e_m -$ первое ребро в этом списке, которого нет в K (такое существует).

$$e_1, e_2, \dots, e_{m-1} \in K, \qquad e_m \notin K$$

Рассмотрим граф $K_1 = K \cup \{e_m\}$. По лемме 2 в K_1 появился цикл (рис. 10.2). Так как $D -$ дерево, не весь цикл входит в $D \Rightarrow \exists$ ребро e в цикле: $e \notin D \Rightarrow e \neq e_m,\ e \in K$.

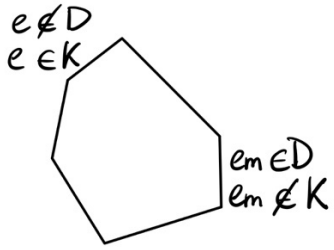

Рис. 10.2. Цикл в графе K_1

Пусть граф $K_2 = K_1 \cup \{e_m\}\backslash\{e\}$. $K_2 -$ связный по лемме 1 и ребер в нем $n-1 \Rightarrow$ по теореме $K_2 -$ дерево, причем остовное. Возможны 3 варианта:

1) $w(e_m) < w(e) \Rightarrow w(K_2) = w(K) + w(e_m) - w(e) < w(K) -$ противоречит тому, что $K -$ кратчайшее остовное дерево.
2) $w(e_m) < w(e) \Rightarrow$ так как $e_1, e_2, \dots, e_{m-1} \in K$, и $e \in K$, и $K -$дерево, то e не образует цикла с $e_1, e_2, \dots, e_{m-1}$, но $w(e) < w(e_m) -$ противоречие с выбором очередного ребра (алгоритм не мог выбрать e_m).
3) $w(e_m) = w(e) \Rightarrow w(K_2) = w(K) + w(e_m) - w(e) = w(K) \Rightarrow K_2 -$ тоже кратчайшее остовное дерево. Но у K_2 больше общих ребер с D, чем у K, потому что выкинули ребро e_m, которое не входило в K, а добавили ребро, которое входит в K – противоречие с выбором K.

Все варианты невозможны ⇒ неверное допущение ⇒ (от противного) $D -$ кратчайшее остовное дерево ■.

Корневые деревья

Опр.1. *Корневым деревом* называется любое дерево с одной выделенной вершиной, которая называется *корнем.*

Опр.2. (индуктивное)

1) Граф с одной вершиной, которая выделена, называется *корневым деревом*

2) Пусть $D_1 = (V_1, E_1),\ D_2 = (V_2, E_2), \dots, D_m = (V_m, E_m)$ являются корневыми деревьями с корнями $v_1, v_2, \dots, v_m$ и пусть для $\forall i \neq j\ V_i \cap V_j = \emptyset$. Тогда корневым деревом также называется граф $D = (V, E)$, где

$V = V_1 \cup V_2 \cup \dots \cup V_m \cup \{v_0\}$, где $\forall i\ (v_0 \notin v_i)$

$E = E_1 \cup E_2 \cup \ldots \cup E_m \cup (v_0, v_1) \cup (v_0, v_2) \cup \ldots \cup (v_0, v_m)$ и корнем объявляется v_0 (рис. 10.3).

При этом $D_1, D_2, \ldots, D_m$ называются корневыми *поддеревьями.*

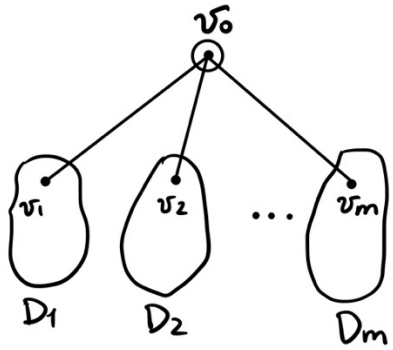

Рис. 10.3. Корневое дерево

3) *Корневым деревом* называют **только** те объекты, которые можно построить по пунктам 1) и 2).

Утверждение. Определение 1 и определение 2 эквивалентны.

Опр. Упорядоченным корневым деревом называется корневое дерево, в котором:

1) задан порядок поддеревьев;
2) каждое поддерево является упорядоченным корневым поддеревом;
3) корневое дерево с 1 вершиной тоже считается упорядоченным.

Лекция 11. Геометрическая реализация графов. Планарные графы.

Число корневых деревьев

Для оценки сложности функций алгебры логики используются схемами. Причем схем сложности $H(n) < 2^{2^n}$, то есть хотя бы для одной функции нужна схема с большой сложностью. В свою очередь число схем заданной сложности оценивается через число корневых деревьев с заданным числом ребер.

Теорема. (о числе корневых деревьев). Число различных упорядоченных корневых деревьев с q ребрами не превосходит 4^q.

Доказательство. Рассмотрим следующий алгоритм обхода упорядоченного корневого дерева «в глубину»:

1. Начать с корня. Пока есть не пройденные поддеревья выполнять:
2. Перейти в корень очередного поддерева, обойти это поддерево «в глубину».
3. Вернуться в корень исходного дерева.

При этом вдоль каждого ребра алгоритм проходит дважды. В соответствии с алгоритмом будем записывать последовательность из 0 и 1: если обход движется по ребру от корня, то пишем 0, если к корню, то 1. Получим последовательность из 0 и 1 длины $2q$ (код дерева). По коду исходное упорядоченное корневое дерево восстанавливается однозначно, так как очередной 0 говорит о том, что надо начинать строить очередное поддерево, а 1 говорит о том, что надо вернуться на один ярус ближе к корню. Из однозначности вытекает, что число упорядоченных корневых деревьев с q ребрами $\leq$ число последовательностей из 0 и 1 длины $2q$, то есть $2^{2q} = 4q \Rightarrow 2^q \leq 4^q$ ■.

Алгоритм, использованный для обхода деревьев, используется для поисковых систем.

Рассмотрим пример обхода «в глубину» дерева на рис. 11.1, считая, что снизу корень, а поддеревья упорядочены слева направо:

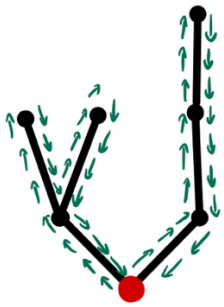

Рис. 11.1. Обход дерева «в глубь»

Изоморфизм корневых деревьев

Изоморфизм деревьев определяется так же, как изоморфизм деревьев, но дополнительно требуется, чтобы корень отображался в корень; а при изоморфизме упорядоченных корневых деревьев дополнительно требуется, чтобы сохранялся порядок поддеревьев.

Число неизоморфных деревьев с q ребрами $\leq$ Число неизоморфных корневых деревьев с q ребрами $\leq$ Число неизоморфных упорядоченных корневых деревьев с q ребрами

Следствие. Число неизоморфных корневых деревьев с q ребрами и число неизморфных деревьев с q ребрами 4^q.

Геометрическая реализация графов

Опр. Пусть $G = (V, E) -$ граф и $V = \{v_1, v_2, \dots, v_p\},\ E = \{e_1, e_2, .., e_q\}$. Будем говорить, что задана некоторая *геометрическая реализация графа* G в некотором пространстве *M*, если

1. Каждой вершине v_i графа G сопоставлена точка a_i в *M,* причем $a_i \neq a_j$ при $i \neq j$.
2. Каждому ребру $e_k = (v_i, v_j)$ сопоставлена непрерывная несамопересекающаяся кривая l_k, содержащая точки a_i и a_j, причем l_k не проходит через точки a_s при $s \neq i,\ s \neq j$.
3. Любые 2 прямые l_k и l_m не имеют общих внутренних точек (то есть кроме концов).

Теорема. Для любого конечного графа $G = (V, E)$ существует реализация в трехмерном евклидовом пространстве.

Доказательство. Пусть $V = \{v_1, v_2, \dots, v_p\},\ E = \{e_1, e_2, .., e_q\}$. Выберем в пространстве произвольную прямую и разместим на ней точки $a_1, a_2, \dots, a_p$ как образы вершин $v_1, v_2, \dots, v_p$ (рис. 11.2). Далее надо провести q ребер так, чтобы они не пересекались. Для этого проведем q полуплоскостей (рис. 11.3).

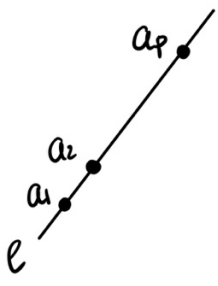

Рис. 11.2. Прямая с образами вершин

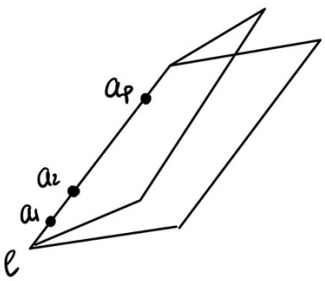

Рис. 11.3. Полуплоскости-ребра

Теперь каждое ребро будем проводить в своей полуплоскости. Очевидно, что у этих полуплоскостей общие точки – только на прямой l. Очевидно, что у ребер нет общих точек (рис. 11.4) ■.

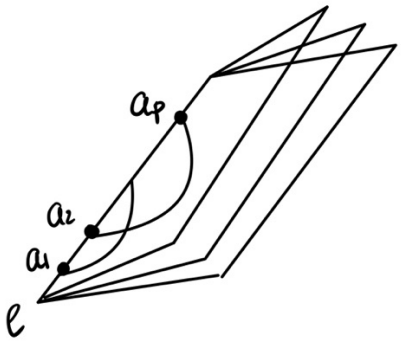

Рис. 11.4. Ребра в полуплоскостях не имеют общих точек

Планарные графы

Опр. Граф называется *планарным,* если существует его геометрическая реализация на плоскости.

Пример реализации планарного графа на рис. 11.5.

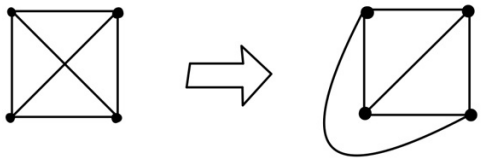

Рис. 11.5. Граф переходит в планарный граф

Если задана геометрическая реализация графа G на плоскости, и мы проведем разрезы плоскости по всем линиям этой геометрической реализации, то плоскость распадется на части. Эти части называются *гранями.* Одна из них не ограничена и называется *внешней гранью*.

У геометрической реализации графа K_4 на рис. 11.5. 4 грани (см. рис. 11.6).

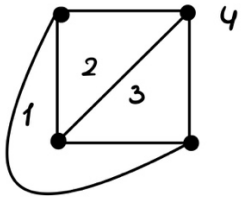

Рис. 11.6. Геометрическая реализация графа K_4

Теорема. Формула Эйлера для планарных графов.

Для любой геометрической реализации связного графа $G = (V, E)$ на плоскости выполняется равенство

$$\boldsymbol{p - q + r = 2}$$

где $p = |V|,\ q = |E|,\ r -$ число граней

В примере на рис.11.5 – 11.6. $p = 4, q = 6,\ r = 4: 4 - 6 + 4 = 2.$

Доказательство (индукция по q при фиксированном p).

1) Базис: $q = p - 1$ так как при $q \le p - 2$ граф связный (по лемме 3)
   По условию $G -$ связный $\Rightarrow G -$ дерево $\Rightarrow r = 1$ (так как нет циклов) $\Rightarrow p - q + r = p - (p - 1) + 1 = 2.$
2) Пусть равенство верно при всех $q: p - 1 \le q \le q_0 - 1.$
   Докажем, что оно верно при $q = q_0$. Пусть в G число ребер $q = q_0 \Rightarrow q_0 \ge p \Rightarrow G -$ не дерево. Но $G -$ связный $\Rightarrow$ в G есть цикл. Пусть $e -$ ребро из цикла. Удалим его из графа G и его образ из геометрической реализации. Получим геометрическую реализацию графа $G_1 = G \backslash \{e_1\}$. G_1 остался связным, так как выброшено ребро из цикла (лемма 1) и для G_1 имеем геометрическую реализацию с числом вершин p, с числом ребер $q_0 - 1$, с числом граней $r - 1$, так как 2 разные грани слились в одну. Так как число ребер $q_0 - 1$, то по предположению индукции для G_1
   $p - (q_0 - 1) + r - 1 = 2 \Rightarrow p - q_0 + r = p - q_0 + r = 2$, то есть для G тоже верно ■.

Замечание. Эта теорема также справедлива для графов, изображенных на сфере, так как рисунок на сфере можно легко перенести на плоскость, взяв рисунок на сфере, вырезав на сфере маленькую дырочку в месте, где нет реализации графа, растянув эту дырочку, получив из сферы диск с изображением графа.

Не планарные графы

Опр. Граф K_5 − полный граф с 5 вершинами (рис. 11.7).

Теорема. Граф K_5 не является планарным.

Доказательство. (от противного). Допустим, что существует геометрическая реализация графа K_5. В ней $p = 5,\ q = 10$ и $p - q + r = 2$ по формуле Эйлера, так как K_5 − связный $\Rightarrow r = 7$.

Пусть q_i − число сторон (ребер) в i-ой грани. Тогда $\sum_{i=1}^{r} q_i = 2q = 20$. Но $\forall i\ q_i \geq 3 \Rightarrow$

$20 = \sum_{i=1}^{r} q_i \geq 21 \Rightarrow 20 \geq 21$ − противоречие ⇒ от противного K_5 не является планарным ■.

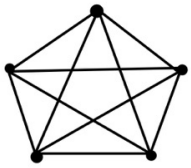

Рис. 11.7. Граф K_5

Опр. Граф $K_{3,3}$ − полный двудольный граф с 6 вершинами.

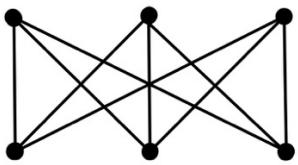

Рис. 11.8. Граф $K_{3,3}$

Теорема. Граф $K_{3,3}$ не является планарным.

Доказательство. Допустим, что существует геометрическая реализация графа$K_{3,3}$. В ней $p = 6,\ q = 9$ и $p - q + r = 2$ по формуле Эйлера, так как $K_{3,3}$ − связный $\Rightarrow r = 5$. Пусть q_i − число сторон (ребер) в i-ой грани. Тогда $\sum_{i=1}^{r} q_i = 2q = 18$.

Но $\forall i\ q_i \geq 4$ (так как нет циклов длины 3) ⇒

$18 = \sum_{i=1}^{5} q_i \geq 20 \Rightarrow 18 \geq 20$ − противоречие ⇒ от противного $K_{3,3}$ не является планарным ■.

Лекция 12. Раскраски графов.

Критерий планарности графа

Опр. *Операцией подразделения ребра* (u, v) в графе $G = (V, E)$ называется операция, при которой:

1) удаляется ребро (u, v)
2) добавляется новая для V вершина w
3) добавляются 2 ребра (u, w) и (w, v) (рис. 12.1).

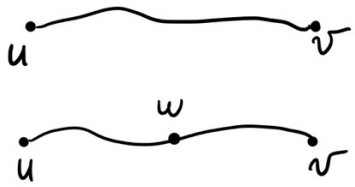

Рис. 12.2. Подразделение ребра

Опр. Граф H называется подразделением графа G, если H можно получить из G путем конечного числа подразделения ребер.

Сам граф также является своим подразделением.

Опр. Два графа G_1 и G_2 называются *гомеоморфными*, если существуют их подразделения, которые изоморфны между собой.

То есть практически гомеоморфные граф имеют одинаковую геометрическую реализацию.

Теорема. (Понтрягина-Куратовского).

Граф является планарным тогда и только тогда, когда он не содержит ни одного подграфа гомеоморфного графу K_5 или графу $K_{3,3}$.

Доказательство.

Необходимость. Пусть $G -$ планарный граф. Тогда существует геометрическая реализация графа G на плоскости. Допустим, что в графе G есть подграф G_1 гомеоморфный либо K_5, либо $K_{3,3}$. Тогда в геометрической реализации графа G на плоскости удалим все точки и линии, которые соответствуют вершинам и ребрам графа G, не входящим в G_1. Мы получим геометрическую реализацию графа G_1 на плоскости. Если в этой геометрической реализации проигнорировать вершины степени 2, то

получим геометрическую реализацию на плоскости графа K_5 или $K_{3,3}$. Это невозможно, так как K_5 и $K_{3,3}$ не планарные ⇒ противоречие ⇒ в G нет подграфов гомеоморфных K_5 или $K_{3,3}$.

Достаточность. Без доказательства ■.

Верхняя оценка числа ребер в планарном графе

Утверждение. Пусть в геометрической реализации связного планарного графа на плоскости r граней и пусть $q_i -$ число сторон в i-ой грани. Тогда

$$\sum_{i=1}^{r} q_i = 2q$$

$q -$ число ребер.

Доказательство следует из того, что в левой части каждое ребро учитывается ровно 2 раза (либо в 2 разных гранях, либо 2 раза в одной грани ■.

Рассмотрим пример графа на рис. 12.3. У его внутренней грани 3 стороны, у внешней – 4, а у средней – 9 (считаем, идя по границе грани).

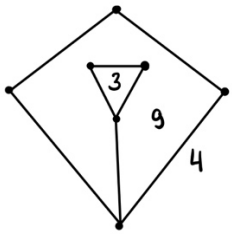

Рис. 12.3. Пример графа

Теорема. Если в связном планарном графе отличном от дерева нет циклов длины $< k$ $(k \geq 3)$, то

$$q \leq \frac{k}{k-2}(p-2)$$

где $p -$ число вершин, $q -$ число ребер.

Доказательство. $G -$ планарный ⇒ рассмотрим его геометрическую реализацию на плоскости. По утверждению, $\sum_{i=1}^{r} q_i = 2q$. Из условия: $\forall i\ q_i \geq k \Rightarrow 2q \geq kr \Rightarrow r \leq \frac{2q}{k}$.
По формуле Эйлера $p - q + r = 2 \Rightarrow r = 2 - p + q \Rightarrow 2 - p + q \leq \frac{2q}{k} \Rightarrow$

$qk \leq 2q + k(p-2) \Rightarrow q(k-2) \leq k(p-2) \Rightarrow q \leq \frac{k}{k-2}(p-2)$ ■.

Следствие. В любом планарном графе с p вершинами, где $p \geq 3$

$$q \leq 3(p-2)$$

Доказательство. Пусть $G -$ связный планарный граф. Если $G -$ не дерево, то (так как в G нет циклов длины меньше 3) по теореме

$$q \leq \frac{3}{3-2}(p-2) = 3(p-2)$$

Если $G -$ дерево, то $q = p - 1$. Проверим $q \leq 3(p-2)$.

$p - 1 \leq 3(p-2) \Leftrightarrow 5 \leq 2p \Leftrightarrow p \geq 3 -$ верно.

Пусть $G -$ планарный, но связный. Тогда можно рассмотреть каждую компоненту на плоскости без пересечения ребер (рис. 12.4). Возьмем вершины на внешних гранях двух компонент и соединим их и так далее. То есть добавляя ребра, получим изображение связного планарного графа с q' ребрами. При этом $q' \geq q$ и $q'\text{Б} = 3(p-2)$, так как новый граф связный $\Rightarrow q \leq q' \leq 3(p-2) \Rightarrow q \leq 3(p-2)$ ■.

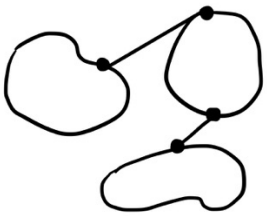

Рис. 12.4. Соединение несвязных компонент

Раскраски графов

Опр. Пусть $G = (V, E)$ и пусть дано некоторое множество $\text{С} = \{c_1, c_2, .., c_k\}$, элементы которого называются цветами или красками.

Раскраской (вершинной) называется любое отображение $\varphi: V \to C$.

Раскраска называется *правильной*, если для любого ребра $(u, v) \in E$ выполняется

$\varphi(u) \neq \varphi(v)$, то есть смежные вершины должны быть раскрашены в разные цвета.

Теорема. Достаточно 5 красок, чтобы правильно раскрасить вершины любого планарного графа.

Доказательство.

Лемма. В любом планарном графе есть вершина степени ≤ 5.

Доказательство.

$\sum_{i=1}^{p} \deg v_i = 2q$ и в планарном графе $q \leq 3p - 6 \Rightarrow \sum_{i=1}^{p} \deg v_i \leq 6p - 12$.

Пусть $d_0 -$ минимальная степень вершины $\Rightarrow \sum_{i=1}^{p} \deg v_i \geq pd_0 \Rightarrow pd_0 \leq 6p - 12 \Rightarrow d_0 < 6 \Rightarrow d_0 \leq 5$ ■.

Индукция по числу вершин p.

1) Базис: $p = 1 -$ очевидно (и для $p \leq 5$).
2) Индуктивный переход: пусть верно, что любой планарный граф с k вершинами правильно раскрашивается в ≤ 5 цветов. Докажем, что это верно для любого планарного графа с $k + 1$ вершиной.
   Пусть граф $G -$ планарный граф с $k + 1$ вершиной $\Rightarrow$ по лемме в G сущесвует вершина v с $\deg v \leq 5$. Рассмотрим геометрическую реализацию графа G на плоскости. Удалим из нее и из графа G вершину v и все ребра инцидентные вершине v. Получим геометрическую реализацию подграфа G_1 с k вершинами $\Rightarrow$ $G_1 -$ планарный и по предположению индукции его вершины можно раскрасить в ≤ 5 цветов. Рассмотрим такую раскраску. Возможны 2 варианта:
   а) У соседей (в G) вершины v используется ≤ 4 цветов. Тогда вершину v красим в цвет, которого нет у соседей и получаем правильную раскраску графа G.
   б) $\deg v = 5$ и у соседей вершины v есть все 5 красок c_1, c_2, c_3, c_4, c_5 (рис. 12.5).

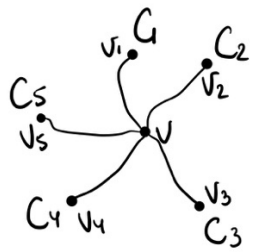

Рис. 12.5. Вершина степени 5 с 5 цветами

Пусть $A -$ множество всех вершин графа G_1, до которых можно дойти из v_1 по ребрам графа G_1 и только по вершинам цветов c_1 и c_3. Возможны 2 варианта:

б1) $v_3 \notin A \Rightarrow$ поменяем в A все цвета $c_1 \rightarrow c_3$ и $c_3 \rightarrow c_1$. Получим правильную раскраску графа G_1, так как смежные вершины с вершинами, раскрашенными в

c_1 и c_3 не могли быть раскрашены в те же цвета, так как они не входят в множество A (рис. 12.6). При этом цвет v_1 и v_3 совпадают и равны c_3. Вершину v покрасим в цвет c_1. Получаем правильную раскраску графа G.

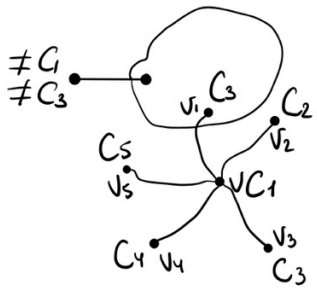

Рис. 12.6. перекраска вершин v и v_1.

б2) $v_3 \in A$ – рис. 12.7.

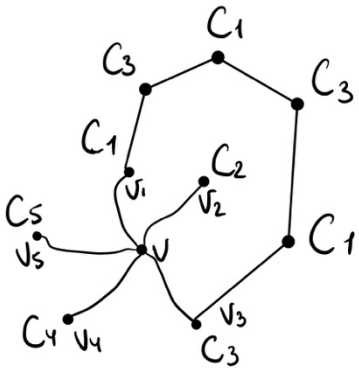

Рис. 12.7. чередование вершин с цветами c_1 и c_3.

Пусть $B -$ множество вершин в G_1, до которых можно дойти по ребрам графа G_1 и только по вершинav цветов c_2 и c_4. Так как существует путь L из v_1 в v_3 по ребрам графа G_1 и только по вершинам цветов c_1 и c_3, то этот путь L вместе с ребрами (v_3, v) и (v, v_1) образует цикл, причем вершины v_2 и v_4 лежат по разные стороны от этого цикла $\Rightarrow v_4 \notin B$, так как любой путь из v_2 в v_4 пересекает этот цикл. Но он не может пересекать по ребру (нет пересечения ребер, так как реализация планарная) $\Rightarrow$ должен пересекать по вершине цвета c_1 или c_3. Далее, как в пункте б1) только для c_2 и c_4: v_2 и v_4 покрасим в c_4, а вершину v в c_2 ■.

Можно ли раскрасить граф в 4 цвета? Это задача была поставлена Хивудом в 1870 г. Меньше, чем в 4 цвета раскрасить не получится. Например, полный граф на 4 вершинах (рис. 12.8) нельзя раскрасить менее, чем 4 цветами.

В 1970-х доказали, что любой планарный граф можно раскрасить 4 цветами. Для доказательства использовали поиск подграфов, которые можно докрасить. За 2 года было найдено порядка 1500 конфигураций, которые оказались сводимыми. Также было доказано, что в каждом планарном графе есть хотя бы одна такая конфигурация.

Эта проблема 4 красок внесла основной вклад в развитие теории графов.

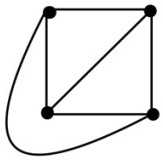

Рис. 12.8. полный граф на 4 вершинах

Лекция 13. Теория кодирования.

Раскраска произвольных графов

Вырожденный случай: в один цвет можно раскрасить граф из произвольного числа вершин, но без ребер.

Теорема Кёнинга. Вершины графа $G = (V, E)$ можно правильно раскрасить в 2 цвета тогда и только тогда, когда в нем нет простых циклов нечетной длины.

Доказательство.

Необходимость. Пусть в графе G есть простой цикл нечетной длины $2k + 1$. Попробуем раскрасить вершины в цвет 1 и цвет 2 (рис. 13.1). Этот цикл не раскрашивается правильно в 2 цвета, следовательно и весь G не раскрашивается правильно в 2 цвета.

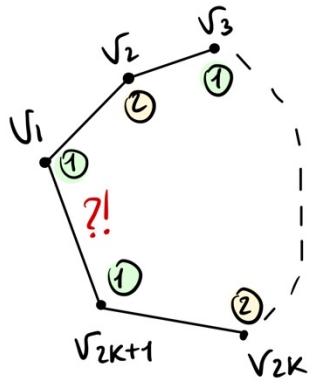

Рис. 13.1 раскраска графа с нечетным простым циклом

Достаточность. Пусть в графе G нет простых циклов нечетной длины.

Лемма 1. Если в графе есть замкнутый путь нечетной длины, то. в этом графе есть простой цикл нечетной длины.

Доказательство. Рассмотрим замкнутый путь L нечетной длины. Если в L вершины не повторяются, то сам L является искомый простой цикл нечетной длины. Иначе пусть v повторяется дважды (≥ 2) в L: $\underbrace{v \dots \dots \dots \dots v}_{L_1} \underbrace{\dots \dots \dots v}_{L_2}$ (рис. 13.2). То есть путь распадается на 2 пути L_1 и L_2, один из которых будет нечетной длины, так как длина L нечетна. То есть либо L_1, либо L_2 – замкнутый путь нечетной длины. Рассмотрим этот замкнутый подпуть L_i ($i = 1$ или $i = 2$). Если в L_i не повторяются вершины, то L_i – простой цикл нечетной длины. Иначе опять берем в L_i повторяющуюся вершину и повторяем предыдущий шаг и так далее. Процесс разбиения конечный, поскольку граф конечный.

Следовательно, на каком-то шаге получим замкнутый путь нечетной длины без повторяющихся вершин. Это и есть искомый простой цикл ■.

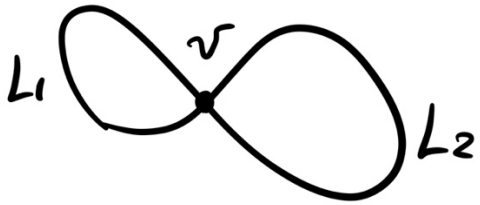

Рис. 13.2 путь с дважды повторяющейся вершиной

Лемма 2. Пусть в графе G нет простых циклов нечетной длины и пусть $a, b - 2$ вершины графа G, лежащие в одной компоненте связности. Тогда длины всех путей из a в b имеют одинаковую четность.

Доказательство (от противного). Допустим, что в графе G существует два пути из a в b L_1, L_2, длины которых имеют разную четность. Обозначим L_2^{-1} – путь L_2, проходимый из b в a. Тогда путь $L_1, L_2^{-1} = L$ имеет нечетную длину. Тогда в графе G существует простой цикл нечетной длины (по лемме 1) – противоречие. Тогда от противного длины всех путей из a в b имеют одинаковую четность ■.

Продолжим доказательство теоремы.

Так как по условию в графе G нет простых циклов нечетной длины, то по лемме 2 для любых двух вершин a и b длина всех путей из a в b имеет одинаковую четность. Будем раскрашивать связные компоненты графа G независимо.

Пусть G_1 – любая связная компонента графа G. Выберем в G_1 любую вершину v_0. Для любой вершины v из G_1: если все пути из v_0 в v имеют четную длину, то красим v в цвет c_1, если у всех длина нечетная, то в цвет c_2. В частности, v_0 будет покрашена в c_1. Эта процедура покраски корректна по лемме 2.

Покажем, что раскраска G_1 правильная (рис. 13.3). Если существует путь из v_0 в v_i нечетной длины, то существует путь из v_i в v_j четной длины (или наоборот).

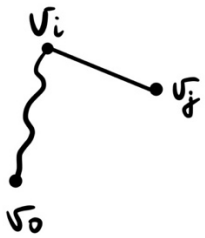

Рис. 13.3. Путь из вершины из v_0 в v_i и из v_i в v_j

В любом случае v_i в v_j окрашены в разные цвета. В любой компоненте раскраска правильная. Тогда и во всем графе получим правильную раскраску (проделывая эту операцию для всех компонентов связности и учитывая, что между собой компоненты связности не связаны) ■.

Коды

Пусть $A, B -$ два конечных алфавита. Через A^* (соответственно B^*) будем обозначать множество всех (конечных) слов в алфавите A (в B).

Опр. Кодированием из алфавита A в B называется любое отображение φ: $A^* \to B^*$.

Рассмотрим частный случай – алфавитное кодирование.

Опр. Алфавитное кодирование из A в B задается отображением φ: $a_i \to B_i \in B$,

где $A = \{a_1, a_2, \dots, a_r\}, i = 1,2, \dots, r$ и дополнительным условием, что слова из A^* кодируются побуквенно, то есть $\varphi(a_{i_1} a_{i_2} \dots a_{i_s}) = B_{i_1} B_{i_2} \dots B_{i_s}$.

Будем считать, что $B_i \neq B_j$.

B_i называют *кодовыми словами*, а $\{B_1, B_2, \dots, B_r\}$ называют *кодом.*

Например, можем закодировать так:

$a \to \quad 0$
$b \to 01$
$c \to 10$

Тогда $\underbrace{10}_{c}\underbrace{0}_{a}$, но последовательность $010 - \begin{matrix} ac \\ ba \end{matrix}$ неоднозначно дешифруется. Такие кодировки мы рассматривать не будем, так как они приводят к потери информации.

Опр. Алфавитное кодирование (код) называется *взаимно однозначным (разделимым, или однозначно декодируемым)*, если для любых двух различных слов $\bar{a}_1 \in A^*$ и $\bar{a}_2 \in A^*$ выполняется $\varphi(\bar{a}_1) \neq \varphi(\bar{a}_2)$.

Опр. Алфавитный код называется *равномерным*, если все кодовые слова имеют одинаковую длину.

Утверждение. Любой равномерный код является взаимно однозначным.

Доказательство. Так как код равномерный, то декодируемое слово можно разбить на кодовые слова одним способом на m кусков, где $m -$ длина кодового слова (рис. 13.4) ■.

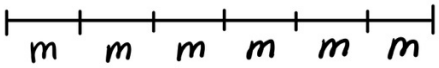

Рис. 13.4. Разбиение равномерного кода

Опр. Алфавитный код называется *префиксным*, если никакое кодовое слово не является началом (префиксом) другого кодового слова.

Утверждение. Любой префиксный код является однозначным.

Доказательство. Если первое кодовое слово можно отрезать двумя разными способами (рис. 13.5), то одно кодовое слово будет являться началом другого кодового слова, что невозможно по определению префиксного кода. Аналогично второе кодовое слово выделяется однозначно и так далее ■.

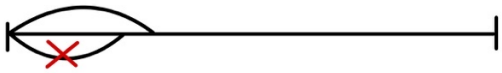

Рис. 13.5. 2 способа выделить первое кодовое слово

Опр. Алфавитный код называется *суффиксным* (постфиксным), если никакое кодовое слово не является кондом другого кодового слова.

Утверждение. Любой суффиксный код является однозначным.

Доказательство повторяет доказательство аналогичную теорему для префиксного кода. Однако, отрезаем слова с конца (рис. 13.6) ■.

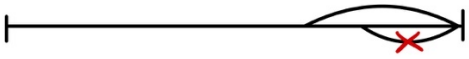

Рис. 13.6. 2 способа выделить последнее кодовое слово

Задача. Дано алфавитное кодирование (код). Требуется выяснить, является ли это кодирование взаимно-однозначным.

Алгоритм.

Вход: код $\{B_1, B_2, \dots, B_r\}$ (кодирование φ).

Построим ориентированный граф $G_\varphi = (V, E)$.

Вершинами графа являются все слова $\beta \in B^*$ такие, что β является собственным началом хотя бы у одного кодового слова и собственным концом хотя бы у одного кодового слова. Собственное начало у слова B_i — это его начало, которое не пусто и не совпадает со всем

словом B_i. Плюс к V добавляется вершина $\Lambda = \mathrm{B}_0 -$ пустое слово. В итоге $V = \{\beta_0, \beta_1, \dots, \beta_k\}$, где β_0 −пустая вершина, $\beta_1, \dots, \beta_k -$ собственные начала и собственные концы.

Пусть β_i и $\beta_j - 2$ вершины (фактически, это слова). Тогда из β_i в β_j проводится дуга ⇔ существует такое кодовое слово B_l, имеющая вид: $B_l = \beta_i D \beta_j$ и $D = B_{j_1} B_{j_2} \dots B_{j_s}$. При этом:

- Если $i \neq 0$ и $j \neq 0$, то $s \geq 0$.
- Если $i = 0$ или $j = 0$, то $s \geq 1$
- Если $i = 0$ и $j = 0$, то $s \geq 2$

Далее алгоритм принимает решение в соответствии с со следующей теоремой:

Теорема. Кодирование φ является взаимно однозначным тогда и только тогда, когда в ориентированном графе G_φ нет ориентированных циклов, проходящих через $\beta_0 = \Lambda$ (в частности, нет петли в Λ).

Доказательство.

Необходимость. Пусть в графе G_φ есть ориентированный цикл, проходящий через Λ. Тогда существует ориентированный цикл, начинающийся в Λ, кончающийся в Λ, и больше Λ не повторяется. Пусть в этом цикле вершины встречаются в порядке: $\Lambda, \beta_1, \beta_2, \beta_3, \dots, \beta_m, \Lambda = \beta_0$. Тогда по построению графа ∃ слова $D_1, D_2, \dots, D_{m+1}$ такие, что $\forall D_j -$ это последовательность новых слов и $\beta_i D_{i+1} \beta_{i+1}$.

Лекция 14. Теорема о взаимной однозначности кодирования. Теорема Маркова.

Теорема о взаимной однозначности кодирования

Теорема. Кодирование φ является взаимно однозначным тогда и только тогда, когда в ориентированном графе G_φ нет ориентированных циклов, проходящих через $\beta_0 = \Lambda$ (в частности, нет петли в Λ).

Доказательство.

Необходимость. Пусть в графе G_φ есть ориентированный цикл, проходящий через вершину $\Lambda = \beta_0$. Этот цикл может быть петлей в вершине $\Lambda = \beta_0 \Rightarrow$ по построению графа существует кодовое слово $B_k = \beta_0 B_{i_1} \dots B_{i_r}\beta_0,\ r \geq 2, B_j$ −кодовые слова. Так как $\Lambda = \beta_0$, то $\underbrace{B_k}_{a_k} = \underbrace{\beta_0 B_{i_1} \dots B_{i_r}\beta_0}_{a_{i_1}\dots a_{i_r}},\ r \geq 2$. Тогда это слово можно по-разному декодировать за счет того, что $r \geq 2$: $a_k \neq a_{i_1} \dots a_{i_r}$. Получили неоднозначно декодируемое слово ⇒ код не взаимно однозначный.

Пусть цикла нет. Тогда существует ориентированный цикл $\Lambda = \beta_0, \beta_1, \beta_2, \dots \beta_n, \beta_0$, причем можно считать, что β_0 больше не встречается, то есть $\beta_j \neq \beta_0$ при $1 \leq j \leq n$. Тогда $\forall i\ \exists\ D_i\ \ \beta_i D_i \beta_{i+1} -$ кодовое слово и D_i распадается на несколько кодовых слов (может быть и 0), причем $D_0 \neq \Lambda$.

Рассмотрим слово $\beta_0 D_0 \beta_1 D_1 \beta_2 D_2 \dots \beta_{n-1} D_{n-1} \beta_n D_n \beta_0$. Его можно разбить на кодовые слова 2 способами:

$\underbrace{\beta_0 D_0 \beta_1}_{\text{кодовое}}, \underbrace{D_1}_{\text{разбивается на кодовые}}, \underbrace{\beta_2 D_2 \beta_3}_{\text{кодовое}}, D_3 \dots, \beta_{n-1} D_{n-1} \beta_n, D_n$ при $n -$ нечетным.

$\underbrace{D_0}_{\text{разбивается на кодовые слова}}, \underbrace{\beta_1 D_1 \beta_2}_{\text{кодовое}}, \underbrace{D_2}_{\text{распадается на кодовые слова}}, \beta_3 D_3 \beta_4, \dots, \underbrace{D_{n-1}}_{\text{распадается на кодовое}}, \underbrace{\beta_n D_n \beta_0}_{\text{кодовое}}$

Получаем 2 разных разбиения, так как $D_0 \neq \Lambda,\ \beta_1 \neq \Lambda \Rightarrow \mathrm{D}_0\beta_1 \neq \underbrace{D'}_{\text{часть } D_0}$

При $n -$ четном аналогично получаются 2 разбиения. Таким образом, в любом случае слово с 2 разными расшифровками. Следовательно, кодирование φ не вазимно однозначное.

Достаточность. Пусть $\varphi -$ не взаимно однозначное кодирование. Тогда существует слово, которое неоднозначно декодируется. Возьмем самое короткое такое слово. В точках 1 слово режется на части первым разбиением, в точках 2 – вторым. Точки 1 и 2 не могут совпасть, так как, если бы эти разбиения совпали в какой-то точке, то либо в левой части (от этой точки) есть разница между ними, либо в правой, а в конце они

сходятся, поэтому они полностью разбивают эту часть, то есть было бы более короткое слово, которое не однозначно расшифровывается (рис. 14.1)

Рис. 14.1. Если бы разбиения 1 и 2 совпали в точке, было бы более короткое слово, которое расшифровывается неоднозначно

То есть, так как мы взяли самое короткое слово, точки разбиения не совпадают. Разрежем все слово во всех точках обоих разбиений и рассмотрим слова, заключенные между точками разных разбиений (рис. 14.2).

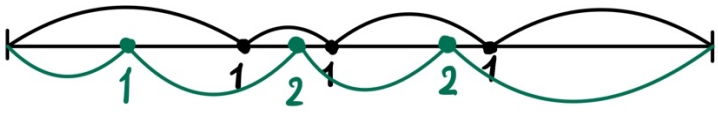

Рис. 14.2. Слова, заключенные между точками разных разбиений

Обозначим их слева направо $\beta_1, \beta_2, \dots, \beta_n$. Каждое такое слово выглядит следующим образом (рис. 14.3):

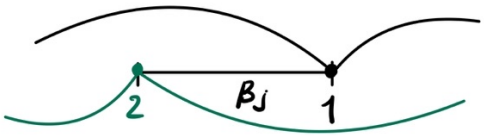

Рис. 14.3. Обозначение слов, заключенных между двумя разбиениями

Тогда β_j – собственное начало некоторого кодового слова и β_j – собственный конец некоторого слова. Тогда по определению G_φ – вершина некоторого G_φ.

Рассмотрим, что происходит в случае, описанном на рис. 14.4.

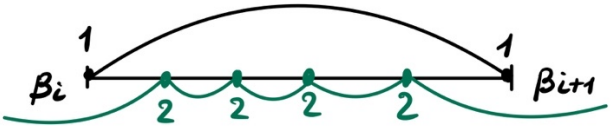

Рис. 14.4. Между точками разбиения 1 лежит несколько точек разбиения 2 и два неполных куска

Законченные дуги 1-1 и 2-2 обозначают кодовые слова, и по определению G_φ в G_φ есть дуга из β_i в β_{i+1}.

В случае, изображенном на рис. 14.4 в G_φ есть дуга из $\beta_0 = \Lambda$ в β_1.

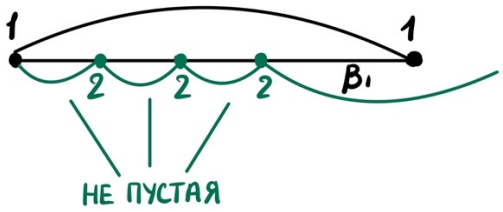

Рис. 14.4. Между точками разбиения 1 сначала лежат несколько точек, потом неполный кусок разбиения 2

Аналогично и в случае на рис. 14.5 в G_φ есть дуга из β_n в $\beta_0 = \Lambda$.

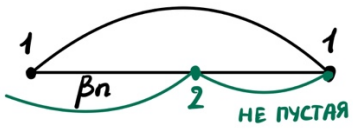

Рис. 14.5. Между точками разбиения 1 сначала лежитнеполный кусок, слово разбиения 2

Тогда в G_φ существует ориентированный замкнутый путь из $\beta_0 = \Lambda$ в β_0. При этом β_0 не встречается внутри пути. Если в этом пути повторяется некоторое β_j, то выбросив из этого пути часть от β_j до β_j, опять получим замкнутый путь из $\beta_0 = \Lambda$ в β_0, но более короткий. Проделав это несколько раз, получим замкнутый ориентированный путь из β_0 в β_0 без повторения вершин, то есть простой цикл.

Возможен случай на рис. 14.6. Здесь в нижнем разбиении ≥ 2 кодовых слов $\Rightarrow$ в G_φ есть петля из β_0 в $\beta_0 = \Lambda$ (то есть частный случай ориентированного цикла из β_0 ■.

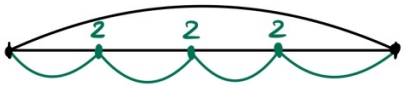

Рис. 14.6. Между точками разбиения 1 лежит несколько точек разбиения 2

Теорема Маркова. Пусть $\varphi -$ алфавитное кодирование: $\varphi(a_i) = B_i \in B^*, i = \overline{1,r}$. Пусть $l_i =$ длина(B_i) и $L = \sum_{i=1}^{r} l_i$. Рассмотрим представление кодовых слов в виде

$B_j = C'B_{i_1}B_{i_2} \dots B_{i_k}C''$, где все $B_j -$ кодовые слова, $C', C'' -$ любые слова в B^* (могут быть пустыми) и пусть $W = \max k$ по всем таким представлениям.Тогда, если $\varphi -$ не взаимно однозначное кодирование, то существует 2 разных слова $\bar{a}' \in A^*, \bar{a}'' \in A^*$, такие что $\varphi(\bar{a}') = \varphi(\bar{a}'')$ и длина:

$$\text{длина}(\bar{a}') \le \left[\frac{(L-r+2)(w+1)}{2}\right], \text{длина}(\bar{a}'') = \left[\frac{(L-r+2)(w+1)}{2}\right]$$

$L -$ суммарная длина всех кодовых слов, $r -$ число кодовых слов, $W -$ максимальное количество кодовых слов, которые могут стоять в кодовом слове.

Доказательство. Пусть $\varphi -$ не взаимно однозначное кодирование $\Rightarrow$ (по теореме о взимной однозначности кодирования) в ориентированном графе G_φ существует ориентированный цикл $\beta_0, \beta_1, \beta_2, \dots \beta_n, \beta_0$, причем (см. док-во досаточности теоремы) существует ориентированный простой цикл, то есть без повторения вершин. По этому циклу построим (как в док-ве необходимости теоремы) построим слово из B^*, которое неоднозначно дешифруется: $\beta_0 D_0 \beta_1 D_1 \beta_2 D_2 \dots \beta_{n-1} D_{n-1} \beta_n D_n \beta_0$, причем все β_j при $1 \le j \le n$ разные и $\ne \beta_0$.

$\underbrace{\beta_0 D_0 \beta_1}_{\text{кодовое}}, \underbrace{D_1}_{\text{разбивается}}, \underbrace{\beta_2 D_2 \beta_3}_{\text{кодовое}}, D_3 \dots, \beta_{n-1} D_{n-1} \beta_n, D_n$ - расшифровка дает $\bar{a}'$

$\underbrace{D_0}_{\text{разбивается}}, \underbrace{\beta_1 D_1 \beta_2}_{\text{кодовое}}, \underbrace{D_2}_{\text{разбивается}}, \beta_3 D_3 \beta_4, \dots, \underbrace{D_{n-1}}_{\text{разбивается}}, \underbrace{\beta_n D_n \beta_0}_{\text{кодовое}}$ – расшифровка дает $\bar{a}''$

$\varphi(\bar{a}') = \varphi(\bar{a}''), \bar{a}' \ne \bar{a}''$

Длина $\bar{a}' -$ число кодовых слов в первом разбиении, длина $\bar{a}'' -$ длина кодовых слов во втором разбиении. Каждое β_j является собственным началом некоторого кодового слова B_i.

У B_i таких начал $l_i - 1 \Rightarrow$ у всех кодовых слов собственных начал $\le \sum_{i=1}^{r} l_i - 1 = L - r$. Так как все $\beta_j -$ разные, то $n \le L - r \Rightarrow$ вставок $D_k \le L - r + 1$. Будем рассматривать эти вставки парами: $(D_0, D_1), (D_2, D_3)$, ... вместе с соседними β.

Тогда пар $\le \left[\frac{L-r+1}{2}\right] \le \left[\frac{L-r+2}{2}\right]$. Рассмотрим такую пару $D_{i-1}\beta_i D_i$.

В первой расшифровке включаем β_{i-1}: $\underbrace{\beta_{i-1} D_{i-1} \beta_i}_{1} \quad \underbrace{D_i}_{\text{распадается на} \le w}$

(так как начинать можно с $\beta_0 = \Lambda$), i – нечетное. Вставка D_i так распадается, потому что тоже находится внутри кодового слова, а по условию слово максимально распадается на w кодовых слов.

Аналогично во второй расшифровке $\underbrace{D_{i-1}}_{\text{распадается на} \le w}\ \underbrace{\beta_i D_i \beta_{i+2}}_{1}$.

Таким образом, на любой паре вставок с присоединенными концами β образуется $\le \underbrace{(w+1)}_{\text{длина}}\ \underbrace{\frac{l-r+2}{2}}_{\text{число пар}} = \frac{(L-r+2)(w+1)}{2}$ кодовых слов. Так как число кодовых слов в каждом из разбиений целое, то оно $\le \left[\frac{(L-r+2)(w+1)}{2}\right]$. Это оценка сверху на длину $\bar{a}', \bar{a}''$ ■.

Замечание: эта верхняя оценка на длину кодируемых (исходных) слов, на которых нарушается взаимная однозначность.

Неравенство Макмиллана

Теорема. Пусть $\varphi: a_i \to B_i \in B^*$ взаимнооднозначное кодирование из алфавита $A = \{a_1, .., a_r\}$ в алфавит B с q буквами. Тогда, если $l_i -$ длина(B_i), то

$$\sum_{i=1}^{r} \frac{1}{q^{l_i}} \le 1$$

$r -$ количество кодовых слов, $q -$ мощность кодирующего алфавита B, $l_i -$ длины кодовых слов.

Доказательство. Пусть $x = \sum_{i=1}^{r} \frac{1}{q^{l_i}}$ Пусть $\forall n \in \mathbb{N}$. Рассмотрим

$$x^n = \left(\sum_{i=1}^{r} \frac{1}{q^{l_{i_1}}}\right) \cdot \left(\sum_{i=1}^{r} \frac{1}{q^{l_{i_2}}}\right) \cdot \ldots \cdot \left(\sum_{i=1}^{r} \frac{1}{q^{l_{i_n}}}\right) = \sum_{i_1=1}^{r} \sum_{i_2=1}^{r} \ldots \sum_{i_n=1}^{r} \frac{1}{q^{l_{i_1}+l_{i_2}+\cdots+l_{i_n}}}$$

Положим $l_{max} = \max\limits_{1 \le i \le r} l_i$

Тогда $\sum_{i_1=1}^{r} \sum_{i_2=1}^{r} \ldots \sum_{i_n=1}^{r} \frac{1}{q^{l_{i_1}+l_{i_2}+\cdots+l_{i_n}}} = \sum_{k=1}^{n \cdot l_{max}} \frac{c_k}{q^k}$, где $c_k -$ число наборов $i_1, i_2, .., i_n$, для которых $l_{i_1} + l_{i_2} + \cdots + l_{i_k} = k$, где все $1 \le i_j \le r$.

Лемма. Для $\forall k\ c^k \le q^k$

Доказательство. Пусть $F_k = \{(i_1, i_2, .., i_n) \mid \forall j\ 1 \le j \le r\ l_{i_1} + l_{i_2} + \cdots + l_{i_n} = k\}$. Тогда $c_k = |F_k|$. Сопоставим каждому набору $i_1, i_2, .., i_n \in F_k$ слово в алфавите B: $B_{i_1} B_{i_2} \ldots B_{i_n} = \psi(i_1, i_2, .., i_n) \Rightarrow$ длина $\psi(i_1, i_2, .., i_n) = l_{i_1} + l_{i_2} + \cdots + l_{i_n} = k$. В силу взаимной однозначности кодирования φ, все слова $\psi(i_1, i_2, .., i_n)$ разные. Следовательно, отображение ψ сопоставляет всем наборам из F_k разные слова длины k в алфавите B с $|B| = q$. А таких слов всего $q^k \Rightarrow |F_k| \le q^k \Rightarrow c_k = |F_k|$ ■.

Продолжим доказательство теоремы.

$$x^n = \sum_{k=1}^{n \cdot l_{max}} \frac{c_k}{q^k} \leq \{\text{по лемме}\} \sum_{k=1}^{n \cdot l_{max}} n \cdot l_{max} \Rightarrow x \leq \sqrt[n]{n \cdot l_{max}} \; \forall n \in \mathbb{N}$$

Рассмотрим $\lim\limits_{n\to\infty} x \leq \lim\limits_{n\to\infty} \sqrt[n]{n \cdot l_{max}}$

$x \leq \lim\limits_{n\to\infty} \sqrt[n]{n} \cdot \lim\limits_{n\to\infty} \sqrt[n]{l_{max}} \leq 1 \cdot 1 = 1$ ■.

Замечание: обратное утверждение (выполняется неравенство Макмиллана ⇒ взаимно однозначное кодирование) в общем случае неверно.

Лекция 15. Неравенство Макмиллана. Оптимальное кодирование.

Теорема (обратная к неравенству Макмиллана). Пусть даны натуральные числа $l_1, l_2, \dots, l_r$ и q. Пусть $A = \{a_1, \dots, a_r\}, B = \{b_1, \dots, b_q\} - 2$ алфавита и пусть выполняется неравенство:

$$\sum_{i=1}^{r} \frac{1}{q^{l_i}} \leq 1$$

Тогда существует *префиксное кодирование* $\varphi: a_i \to B_i \in B^*$ такое, что длина$(B_i)= l_i$.

Доказательство. Пусть $l_{max} = \max\limits_{1 \leq i \leq r} l_i$ и пусть для $\forall k\ 1 \leq k \leq l_{max}$ d_k обозначает число тех l_i, которые равны k. Тогда исходное неравенство можно переписать в виде

$\sum_{k=1}^{l_{max}} \frac{d_k}{q^k} \leq 1$ −дано. Нам надо построить префиксный код в алфавите B $(|B| = q)$ такой, что в этом коде d_1 слов длины 1, d_2 слов длины 2, …, d_n слов длины l_{max}.

Пусть $\forall m \in \mathbb{N}\ 1 \leq m \leq l_{max}$ из $\sum_{k=1}^{l_{max}} \frac{d_k}{q^k} \leq 1$ получаем $\sum_{k=1}^{m} \frac{d_k}{q^k} \leq 1 \Rightarrow$

$\Rightarrow \sum_{k=1}^{m} d_k q^{m-k} \leq q^m$

$d_1 q^{m-1} + d_2 q^{m-2} + \dots + d_{m-1} q + d_m \leq q^m \ \Rightarrow$

$\Rightarrow d_m \leq q^m - (d_1 q^{m-1} + d_2 q^{m-2} + \dots + d_{m-1} q)$ ***(15.1)***

Будем строить искомый префиксный код в следующем порядке: сначала d_1 слов длины 1, затем еще d_2 слов длины 2, …, d_m слов длины m. При этом будем следить, чтобы не нарушалось свойство префиксного кода.

Базис. Надо выбрать d_1 слов длины 1. Возьмем $m = 1$ для неравенства 15.1: $d_1 \leq q \Rightarrow$ можно выбрать.

Переход. Пусть уже выбраны d_1 слов длины 1, d_2 слов длины 2, …, d_{m-1} слов длины $m - 1$ и пусть пока свойство префиксного кода не нарушается. Нам надо добавить в код d_m слов длины m так, чтобы не нарушалось свойство префиксного кода. Выбранные слова запрещают добавлять в код некоторые слова длины m. Одно слово p длины l, уже включенное в код запрещает добавлять (рис. 15.1) q^{m-l} слов.

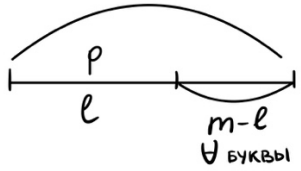

Рис. 15.1. Ограничение на слова в коде из-за слова p длины l

Все d_l слов длины l запрещают добавлять $\leq (=) d_l \cdot q^{m-l}$ слов. Тогда все уже выбранные слова запрещают $\leq d_1 \cdot q^{m-1} + d_2 \cdot q^{m-2} + \cdots + d_{m-1} \cdot q$ $\Rightarrow$ разрешенных слов длины m $\geq q^m - (d_1 \cdot q^{m-1} + d_2 \cdot q^{m-2} + \cdots + d_{m-1} \cdot q)$. Тогда из неравенства 15.1 следует, что $d_m \leq$ числа разрешенных слов. Таким образом, можно выбрать d_m слов длины m, не нарушая свойства префиксного кода.

Берем $\forall d_m$ слов из разрешенных. Так поступаем в цикле при $m = 2,3, \dots, l_{max}$. Получим искомый префиксный код ■.

Следствие из теоремы о неравенства Макмиллана и обратной теоремы. Пусть в алфавите B существует взаимно однозначный код с длинами кодовых слов $l_1, l_2, \dots, l_r$. Тогда в алфавите B существует префиксный код с длинами кодовых слов $l_1, l_2, \dots, l_r$.

Доказательство. Пусть $|B| = q$. Так как есть взаимно однозначный код с длинами $l_1, l_2, \dots, l_r$, то (по теореме о неравенстве Макмиллана) выполняется неравенство Макмиллана: $\sum_{i=1}^{r} \frac{1}{q^{l_i}} \leq 1$. Но тогда (по обратной теореме) существует префиксный код с длинами $l_1, l_2, \dots, l_r$ ■.

Оптимальные коды

Рассмотрим только случай, когда $B = \{0,1\}$. Пусть исходный алфавит $A = \{a_1, \dots, a_r\}$ – закодирован словами $B_1, B_2, \dots, B_r$ длины соответственно $l_1, l_2, \dots, l_r$, а также известно, что каждый символ a_i появляется с некоторой частотой p_i $(\forall i \ \ p_i > 0)$. Частота буквы – это количество вхождений данной буквы, деленное на общее количество символов в тексте.

$p_1 - a_1 \rightarrow \quad B_1 - l_1$
$p_2 - a_2 \rightarrow \quad B_2 - l_2$
$p_r - a_r \rightarrow \quad B_r - l_r$

Во сколько раз при кодировании растягивается текст?

Пусть в исходном тексте N букв. И пусть a_i встречается N_i раз.

Тогда $\frac{N_i}{N} \approx p_i \ \Rightarrow N_i \approx N \cdot p_i$. При кодировании каждый символ a_i порождает l_i букв алфавита B, а все буквы a_i в тексте порождают $\approx N p_i \cdot l_i$ букв алфавита B. Тогда длина сообщения после кодирования $\approx \sum_{i=1}^{r} N p_i l_i = N \cdot \underbrace{\sum_{i=1}^{r} p_i l_i}_{\text{коэффициент растяжения}}$

Опр. Пусть зафиксированы частоты исходных символов $p_1, p_2, \dots, p_r$ $(\forall j \ p_j > 0)$, $\sum_{i=1}^{r} p_i = 1$. Тогда кодирование $\varphi: a_i \rightarrow B_i$ с длиной$(B_i) = l_i, i = \overline{1,r}$ называется *оптимальным* (для заданных частот), если на нем достигается $\inf_{\Psi - \text{взаимно одн.код}} c(\psi)$, где $c(\psi) = \sum p_i l_i'$, где $l_i' -$ длины кодовых слов в ψ.

Опр. $c(\psi)$ называется *ценой* (стоимостью).

Теорема. Для любого набора положительных частот $p_1, p_2, \dots, p_r$ оптимальные (двоичные) коды существуют и их цена $c(\psi) \leq [\log_2 r]$.

Доказательство. Пусть $k = [\log_2 r] \Rightarrow 2^k \geq r$. Тогда в качестве кода можно взять r различных слов из 0 и 1 длины r. Этот код будет равномерным, а значит взаимно однозначным. Его цена $\sum_{i=1}^{r} p_i k = k \sum_{i=1}^{r} p_i = k \cdot 1 = [\log_2 r]$. Тогда при поиске $\inf c(\psi)$ достаточно рассматривать только те взаимно однозначные коды, у которых цена $\leq k = [\log_2 r]$. Тогда у них для $\forall i \;\; p_i \cdot l_i' \leq k \;\; \Rightarrow l_i' \leq \frac{k}{p_i} \Rightarrow l_i'$ может принимать только конечное число значений $(\forall i) \Rightarrow$ приходим к поиску inf на конечном множестве кода $\Rightarrow$ $inf = min \Rightarrow$ inf достигается $\Rightarrow$ существуют оптимальные коды ■.

Утверждение. Для любого набора положительных частот $p_1, p_2, \dots, p_r$ существует оптимальный префиксный код.

Доказательство. По предыдущей теореме для этих частот существует оптимальный взаимно однозначный код с некоторыми длинами кодовых слов $l_1, l_2, \dots, l_r \Rightarrow$ по следствию существует префиксный код с длинами кодовых слов $l_1, l_2, \dots, l_r$. Если слова этого префиксного кода сопоставить исходным символам, чтобы длина слова B_i оставалась такой же, как в оптимальном взаимно однозначном коде, то цена не изменится и префиксный код тоже будет оптимальным ■.

Свойства оптимального кода

Лемма 1. Пусть $\varphi -$ оптимальное кодирование и пусть $p_i > p_j$. Тогда $l_i \leq l_j$

Доказательство. (от противного). Допустим, что $l_i > l_j$. Рассмотрим новое кодирование ψ, в котором

$$\psi(a_s) = \varphi(a_s), \qquad \begin{matrix} \psi(a_i) = \varphi(a_j) = B_j \\ \psi(a_j) = \varphi(a_i) = B_i \end{matrix}$$

Рассмотрим $c(\varphi) - c(\psi) = (p_i l_i + p_j l_j) - (p_i l_j + p_j l_i) = \underbrace{(p_i - p_j)}_{>0} \underbrace{(l_i - l_j)}_{>0} \Rightarrow$

$c(\psi) < c(\varphi)$ и $\psi -$ взаимно однозначный код $\Rightarrow$ не оптимальный код $\Rightarrow$ противоречие с условием $\Rightarrow$ от противного $l_i \leq l_j$ ■.

Лемма 2. Пусть $\varphi -$ оптимальное префиксное кодирование и пусть $l_{max} = \max\limits_{1 \leq i \leq r} l_i$. Пусть в коде φ есть слово $B = B'\alpha$ $(\alpha \in \{0,1\})$ с длиной$(B) = l_{max}$. Тогда в коде φ есть и слово $B = B'\bar{\alpha}$.

Доказательство. (от противного) Пусть в префиксном коде φ слово $B'\alpha$ с длиной$(B'\alpha) = l_{max}$. Допустим, в φ нет слова $B'\bar{\alpha}$. Тогда построим новое кодирование ψ (для тех же частот), заменив в коде φ слово $B'\alpha$ на B.

Тогда $\mathrm{c}(\varphi) - c(\psi) = p_s \cdot \text{длина}(B'\alpha) - p_s \cdot \text{длина}(B') = p_s > 0$, где $p_s -$ частота входного символа в φ кодируется как $B'\alpha$. Отметим, что при этом $\psi -$ это тоже префиксный код:

1) так как $B'\alpha \in \varphi$, то $B' \notin \varphi$ и $\varphi -$префиксный код $\Rightarrow$ в коде ψ нет равных слов;
2) префиксность в ψ могла бы нарушиться, только если B' стало началом другого слова, но $B'\alpha$ мы выкинули, $B'^{\bar{\alpha}}$не было по нашему допущению, а более длинных слов, начинающихся с B' в коде ψ нет, так как $\text{длина}(B'\alpha) = l_{max}$.

$\Rightarrow B'$ не может стать началом другого кодового слова, и $\psi -$ это тоже префиксный код ($\Rightarrow$ взаимно однозначный), и $\mathrm{c}(\varphi) > c(\psi) \Rightarrow \varphi -$ не оптимальный код $\Rightarrow$ от противного $B'\bar{\alpha} \in \varphi$ ■.

Лекция 16. Теорема редукции. Коды, исправляющие ошибки.

Свойства оптимальных кодов

Лемма 3. Пусть $\varphi -$ оптимальный префиксный код, и пусть $p_1 \geq p_2 \geq \cdots \geq p_{r-1} \geq p_r$. Тогда можно так переставить слов в коде φ, что получится снова оптимальный префиксный код, в котором частотам p_{r-1} и p_r соответствуют кодовые слова, отличающиеся только последним символом ($B'0$ и $B'1$).

Доказательство. По лемме 2 в коде φ существуют 2 слова длины l_{max} вида

$B_i = B'0$ и $B_j = B'1$. Переставим в коде φ слова B_k и B_m и пусть $p_k \geq p_m$ и $l_k \geq l_m$. Тогда получаем некоторый код ψ и $\mathrm{c}(\varphi) - c(\psi) = (p_k l_k + p_m l_m) - (p_k l_m + p_m l_k) = \underbrace{(p_k - p_m)}_{\geq 0}\underbrace{(l_k - l_m)}_{\geq 0} \geq 0 \Rightarrow c(\psi) \leq \mathrm{c}(\varphi)$.

Поменяем местами пару слов B_i, B_j с парой слов B_{r-1}, B_r.

Если i или j совпадают с $r-1$ или r, то меняются местами только 2 слова (или даже ни одного). Имеет место ситуация, описанная выше, так как

$\text{длина}(B_i) = \text{длина}(B_j) = l_{max} \geq \text{длина}(B_{r-1})$ и длина (B_r).

Тогда после перестановки получим префиксный код ψ с $c(\psi) \leq \mathrm{c}(\varphi)$, так как $\varphi -$ оптимальный, а $\psi -$ взаимно однозначный, то не может $c(\psi) < \mathrm{c}(\varphi) \Rightarrow c(\psi) = \mathrm{c}(\varphi)$. Таким образом, получили оптимальный префиксный код ψ, в котором $B'0$ и $B'1$ стоят на двух последних местах (соответствуют частотам p_{r-1} и p_r) ■.

Замечание. В этой лемме не утверждалось, что для любого оптимального кода самым маленьким частот соответствуют 2 слова, отличающиеся 0 и 1.

Лемма 4. Рассмотрим 2 кодирования φ и φ' ($p' + p'' = p_k$):

$$\varphi: \begin{matrix} p_1 & p_2 \dots & p_{k-1} & p_k \\ B_1 & B_2 \dots & B_{k-1} & B_k \end{matrix} \quad \text{и} \quad \varphi': \begin{matrix} p_1 & p_2 \dots & p_{k-1} & p' & p'' \\ B_1 & B_2 \dots & B_{k-1} & B_k 0 & B_k 1 \end{matrix}$$

Если одно из этих кодирований является префиксным, то и второе – тоже префиксное и при этом $\mathrm{c}(\varphi) = \mathrm{c}(\varphi') + p_k$.

Доказательство.

1) Первая часть доказывается перебором случаев.

   Если в одном кодировании ни одно слово не является началом другого, то и во втором кодировании. ни одно слово не является началом другого. Допустим, $\varphi -$ префиксное кодирование. Тогда в φ' префиксность может испортиться только на $B_k 0$ и $B_k 1$. Но тогда какое-то слово должно являться началом B_k, и тогда в коде

φ так же какое-то слово является началом B_k, чего быть не может, так как $\varphi -$ префиксное кодирование. Кроме того, слово $B_k 0$ не может совпасть ни с одним словом из φ, так как тогда бы B_k было бы его началом. Аналогичные рассуждения в обратную сторону.

2) $c(\varphi') - c(\varphi) = \big(p' \text{дл}(B_k 0) + p'' \text{дл}(B_k 1)\big) - p_k \text{дл}(B_k) =$
3) $= (p' + p'')(\text{дл}(B_k) + 1) - p_k \text{дл}(B_k) = p_k(\text{дл}(B_k) + 1) - p_k \text{дл}(B_k) = p_k$ ■.

Теорема редукции

Теорема редукции. Пусть φ и φ' 2 кодирования ($p' + p'' = p_k$):

$$\varphi: \begin{matrix} p_1 & p_2 \dots & p_{k-1} & p_k \\ B_1 & B_2 \dots & B_{k-1} & B_k \end{matrix} \quad \text{и} \quad \varphi': \begin{matrix} p_1 & p_2 \dots & p_{k-1} & p' & p'' \\ B_1 & B_2 \dots & B_{k-1} & B_k 0 & B_k 1 \end{matrix}$$

1) Если $\varphi' -$ оптимальное префиксное кодирование (для своих частот), то $\varphi -$ тоже оптимальное префиксное кодирование (для своих частот).
2) Если $\varphi -$ оптимальное префиксное кодирование (для своих частот) и $p_1 \geq p_2 \geq \cdots \geq p_{k-1} \geq p' \geq p''$, то $\varphi -$ тоже оптимальное префиксное кодирование (для своих частот).

Доказательство.

1)(От противного): пусть $\varphi' -$ оптимальное префиксное кодирование (для своих частот). Допустим, что φ не является оптимальным префиксным кодированием. По лемме 4 $\varphi -$ префиксное $\Rightarrow c(\varphi) -$ не минимально $\Rightarrow$ существует оптимальный префиксный код ψ для $p_1, p_2, \dots, p_{k-1}, p_k$:

$$\psi: \begin{matrix} p_1 & p_2 \dots & p_{k-1} & p_k \\ D_1 & D_2 \dots & D_{k-1} & D_k \end{matrix}$$
$$c(\psi) < c(\varphi)$$

Построим кодирование ψ' для $p_1 \geq p_2 \geq \cdots \geq p_{k-1} \geq p' \geq p''$:

$$\psi': \begin{matrix} p_1 & p_2 \dots & p_{k-1} & p' & p'' \\ D_1 & D_2 \dots & D_{k-1} & D_k 0 & D_k 1 \end{matrix}$$

$\Rightarrow$ по лемме 4 $\psi' -$ тоже префиксный код и $c(\psi') = c(\psi) + p_k$, $c(\varphi') = c(\varphi) + p_k \Rightarrow$ $c(\psi') < c(\varphi') -$ противоречие с оптимальностью $\varphi' \Rightarrow$ от противного φ является оптимальным префиксным кодом.

2)Пусть $\varphi -$ оптимальное префиксное кодирование и $p_1 \geq p_2 \geq \cdots \geq p_{k-1} \geq p' \geq p''$. Допустим, что φ' не является оптимальным префиксным кодом. Но по лемме 4 $\varphi' -$ префиксный $\Rightarrow c(\varphi') -$ не минимальный. Рассмотрим оптимальный префиксный код ψ', $c(\psi') < c(\varphi')$, причем такой, у которого 2 последних слова различаются только в

последнем символе (такой код существует по лемме 3). Построим кодирование ψ для частот $p_1 \geq p_2 \geq \cdots \geq p_{k-1} \geq p_k$, заменив 2 последних слова D_k0 и D_k1 на D_k и сложив частоты $p' + p'' = p_k$. Тогда по лемме 4 кодирование ψ тоже префиксное и при этом $c(\psi') = c(\psi) + p_k, \quad c(\varphi') = c(\varphi) + p_k \quad \Rightarrow c(\psi') < c(\varphi')$ - противоречие $\Rightarrow \varphi' -$ оптимальное префиксное кодирование ■.

Смысл теоремы редукции в том, что, сложив 2 самые маленькие частоты, мы можем перейти к задаче построения оптимального кода для склеенных частот. Тогда решив задачу построения для склеенных частот, можно перейти к оптимальному коду для исходных частот, расклеив обратно частоту путем добавления 0 и 1. Таким образом, чтобы найти оптимальный код для k частот, складываем 2 частоты и так далее, пока не дойдем до 2 частот, которые можно закодировать 0 и 1. Такой алгоритм построения называется *методом Хаффмана.*

Коды, исправляющие ошибки

Рассматриваем только двоичные равномерные коды с длиной слов *n*. Допускаются *ошибки типа замещения*: $0 \to 1, \ 1 \to 0$.

При этом пришедшее сообщение однозначное режется на куски длины *n* и проблема: по каждому куску узнать, каким он был исходно.

Опр. Пусть дан двоичный равномерный код $\varphi = \{B_1, B_2, \dots, B_K\}$ (все слова длины *n*). Будем говорить, что *код исправляет r ошибок*, если при наличии не более *r* ошибок типа замещения всегда можно сказать, каким было исходное кодовое слово.

Введем на множестве всех наборов из E_2^n *расстояние Хэмминга:*

$\rho(\tilde{\alpha}, \tilde{\beta})$ –это число разрядов, в которых $\tilde{\alpha}$ и $\tilde{\beta}$ различаются ($\alpha_i \neq \beta_i$). Можем рассматривать расстояние Хэмминга как метрику.

Опр. Шаром радиуса r в E_2^n с центром $\tilde{\alpha} \in E_2^n$ называется множество всех наборов $\tilde{\beta} \in E_2^n$ таких, что $\rho(\tilde{\alpha}, \tilde{\beta}) \leq r$.

Опр. Кодовым расстоянием $\rho_{min}(K)$ кода $K = \{B_1, B_2, \dots, B_k\}$ называется $\min\limits_{i \neq j} \rho\,(B_i, B_j)$.

Утверждение. Код K исправляет r ошибок $\Leftrightarrow \rho_{min}(K) \geq 2r + 1$.

Доказательство.

Все слова, которые могут получиться при ошибке типа замещения при передаче слова B_1 образуют шар радиуса r, так при передаче возможно $\leq r$ ошибок типа замещения. Аналогично для остальных B_i. Соответственно для того, чтобы можно было восстановить код, необходимо, чтобы шары радиуса r с центрами в кодовых словах β_i не пересекались (рис. 16.1) $\Leftrightarrow \forall i \neq j\ \rho(B_i, B_j) \geq 2r + 1$ ■.

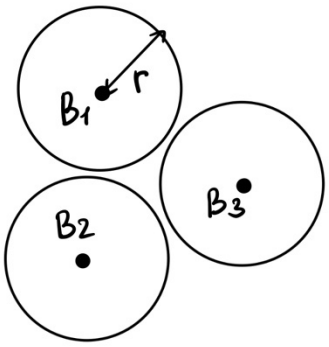

Рис. 16.1. Непересекающиеся шары

Таким образом, чтобы узнать сколько ошибок исправляет код, необходимо посчитать минимальное кодовое расстояние и вычислить, при каком r выполняется такое соотношение: $\rho_{min}(K) \geq 2r + 1$.

Опр. Будем говорить, что код K *обнаруживает* r *ошибок*, если при наличии в любом кодовом слове $\leq r$ ошибок замещения можно сказать, были ли ошибки.

Утверждение. Код K обнаруживает r ошибок $\Leftrightarrow \rho_{min}(K) \geq r + 1$.

Доказательство. Код обнаруживает r ошибок $\Leftrightarrow$ никакое кодовое слово не лежит в шаре радиуса r с центром в другом кодовом слове (рис. 16.2), то есть шары могут пересекаться, но ни один не содержит другой $\Leftrightarrow\ \forall i \neq j\ \rho(B_i, B_j) \geq r + 1\ \Leftrightarrow \rho_{min}(K) \geq r + 1$ ■.

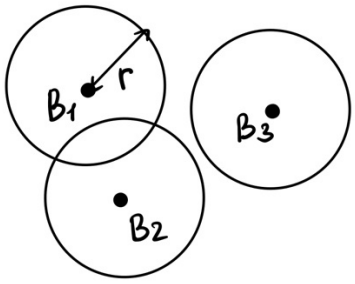

Рис. 16.2. Шары кода, обнаруживающего r ошибок, могут пересекаться

Лекция 17. Коды Хэмминга.

Коды, исправляющие r ошибок

Опр. Пусть $S_r(n)$ обозначает число точек (наборов длины n) в шаре радиуса r в E_2^n.

Утверждение. $S_r(n) = 1 + n + C_n^2 + \cdots + C_n^r$

Доказательство. Шар состоит из центра $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$, наборов, отличающихся от $\tilde{\alpha}$ в одном разряде – их $C_n^1 = n$, наборов, отличающихся от $\tilde{\alpha}$ в двух разрядах – их C_n^2 и так далее до наборов, отличающихся от $\tilde{\alpha}$ в r разрядах. Соответственно, $S_r(n) = 1 + n + C_n^2 + \cdots + C_n^r$ ■.

Опр. Определим функцию $M_r(n)$ как максимальное число наборов из E_2^n, которые образуют код, исправляющий r ошибок.

Теорема. $\frac{2^n}{S_{2r}(n)} \le M_r(n) \le \frac{2^n}{S_r(n)}$

Доказательство. Пусть дан любой код $C = \{\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_m\}$, исправляющий r ошибок. Тогда шары радиуса r с центрами в точках $\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_m$ не имеют общих точек. в каждом таком шаре число точек $S_r(n)$, а во всех шарах точек $mS_r(n)$. Так как шары не пресекаются и всего точек (наборов длины n) 2^n, то $S_r(n) \le 2^n \Rightarrow \frac{2^n}{S_r(n)}$ для любого кода, исправляющего r ошибок $\Rightarrow \max m = M_r(n) \le \frac{2^n}{S_r(n)}$.

Будем строить код C, исправляющий r ошибок.

Выберем (включим в код C) любой набор $\tilde{\alpha}_1 \Rightarrow C$ уже нельзя включать наборы, отстоящие от $\tilde{\alpha}_1$ на расстоянии $\le 2r$, то есть все точки из шара радиуса $2r$ с центром в $\tilde{\alpha}_1$, а остальные можно. В качестве $\tilde{\alpha}_2$ включим в C любой из разрешенных наборов.

Пусть уже выбраны $\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_k$. Тогда запрещено брать наборы, попадающие хотя бы в один шар радиуса $2r$ с центром в одной из точек $\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_k$. Если остается хотя бы одна разрешенная точка, то добавляем к C любой разрешенный набор. И так поступаем «до упора». Пусть остановились на коде $C = \{\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_m\} \Rightarrow$ шары радиуса $2r$ с центрами $\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_m$ покрывают все 2^n наборов. Тогда общее число точек в этих шарах $mS_{2r}(n) \ge 2^n \Rightarrow m \ge \frac{2^n}{S_{2r}(n)} \Rightarrow M_r(n) \ge \frac{2^n}{S_{2r}(n)}$ ■.

Коды Хэмминга

Пусть $n -$ длина кодовых слов из E_2^n.

Выберем $\forall k \in \mathbb{N}$: $2^{k-1} \le n \le 2^k - 1 \Leftrightarrow \begin{cases} n \ge 2^{k-1} \\ n \le 2^k - 1 \end{cases} \Rightarrow \begin{cases} k = 1 + [\log_2 n] \\ k = [\log_2 (n+1)] \end{cases}$

$n \le 2^k - 1 \Rightarrow$ любое натуральное число от 1 до n, можно представить k разрядами, дописывая при необходимости впереди 0.

Введем множества $D_i = \{m \mid 1 \le m \le n,\ m = (m_{k-1} m_{k-2} \dots m_1 m_0)_2 \ \Rightarrow m_i = 1\}$.

$D_0 = \{1,3,5,7, \dots\}$

$D_1 = \{2,3,6,7, \dots\}$

$D_2 = \{4,5,6,7,12,13,14,15,20 \dots\}$

Опр. Кодом Хэмминга порядка (длины) n называется множество всех наборов

$\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n) \in E_2^n$, удовлетворяющих системе уравнений (сложение по $mod\ 2$)

$$\begin{cases} \sum_{j \in D_0} \alpha_j = 0 \\ \sum_{j \in D_1} \alpha_j = 0 \\ \dots \\ \sum_{j \in D_{k-1}} \alpha_j = 0 \end{cases}, \quad k = 1 + [\log_2 n]$$

Теорема. Код Хэмминга порядка n содержит 2^{n-k} наборов, где $k = 1 + [\log_2 n]$ и исправляет 1 ошибку.

Доказательство.

1)Система имеет вид:

$$\begin{cases} \alpha_1 + (\alpha_3 + \cdots) = 0 \\ \alpha_2 + (\alpha_3 + \cdots) = 0 \\ \alpha_2 + (\dots\dots\dots) = 0 \\ \alpha_8 + (\dots\dots\dots) = 0 \\ \dots \\ \alpha_{2^{k-1}} + (\dots\dots) = 0 \end{cases}$$

причем $\alpha_1, \alpha_2, \alpha_4, \alpha_8$, ... больше нигде в системе не встречаются.

Чтобы получить все решения системы, можно произвольно выбирать все α_j, кроме $\alpha_1, \alpha_2, \alpha_4, \alpha_8, \dots, \alpha_{2^{k-1}}$ (таких α_j $n - k$ штук). Тогда их можно выбрать 2^{n-k} способами. При этом $\alpha_1, \alpha_2, \alpha_4, \alpha_8, \dots, \alpha_{2^{k-1}}$однозначно определяются из системы ⇒ всего решений $n - k$ (можно доказывать через СЛАУ в общем случае).

2)Докажем, что код Хэмминга исправляет 1 ошибку. Пусть передавалось слово из кода Хэмминга $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)$ и упсть произошла 1 ошибка в позиции с номером $d \le n$.

$d = (\gamma_{k-1}\gamma_{k-1} \dots \gamma_1\gamma_0)$.

Тогда на выходе будет слово $\tilde{\beta} = (\beta_1\beta_2 \dots \beta_n)$, где $\beta_j = \alpha_j$ при $j \neq d$ и $\beta_d = \alpha_d \oplus 1$. Тот, кто получил слово $\tilde{\beta}$ с ошибкой, должен посчитать следующие суммы по mod 2:

$$\begin{cases} \delta_0 = \sum_{j \in D_0} \beta_j \\ \delta_1 = \sum_{j \in D_1} \beta_j \\ \dots\ \dots\ \dots \\ \dots\ \dots\ \dots \\ \dots\ \dots\ \dots \\ \delta_{k-1} = \sum_{j \in D_{k-1}} \beta_j \end{cases}$$

Утверждение. $(\delta_{k-1}\delta_{k-2} \dots \delta_1\delta_0)_2 = d$

Доказательство. Рассмотрим 2 случая:

1)Пусть $\gamma_p = 0 \Rightarrow d \notin D_p \Rightarrow \sum_{j \in D_p} \beta_j = \sum_{j \in D_p} d_j = 0$ по определению кода Хэмминга $\Rightarrow$ $\delta_p = \sum_{j \in D_p} \beta_j = 0 \Rightarrow \delta_p = \gamma_p$.

2) Пусть $\gamma_p = 1 \Rightarrow d \in D_p \Rightarrow \sum_{j \in D_p} \beta_j = (\sum_{j \in D_p} d_j) + 1 = 0 + 1 = 1$ по определению кода Хэмминга $\Rightarrow \delta_p = \sum_{j \in D_p} \beta_j = 1 \Rightarrow \delta_p = \gamma_p$.

Следовательно, во всех случаях $\delta_p = \gamma_p \Rightarrow (\delta_{k-1}\delta_{k-2} \dots \delta_1\delta_0)_2 = (\gamma_{k-1}\gamma_{k-2} \dots \gamma_1\gamma_0)_2 = d$ ■.

Теорема. $\frac{2^n}{2n} \le M_1(n) \le \frac{2^n}{n+1}$ – оценка для максимального числа наборов в коде, исправляющем 1 ошибку.

Доказательство. 1)Ранее было доказано: $\frac{2^n}{S_{2r}(n)} \le M_r(n) \le \frac{2^n}{S_r(n)}$**.** При $r = 1$ правая часть имеет вид: $M_1(n) \le \frac{2^n}{S_1(n)} = \frac{2^n}{n+1}$, так как число наборов в шаре радиуса 1 – центральная точка и n точек, отстоящих от центра на 1.

2)Рассмотрим код Хэмминга порядка n.

Его мощность $m = 2^{n-k}$, где $k = 1 + [\log_2 n] \le 1 + \log_2 n \Rightarrow m \ge 2^{n-(1+\log_2 n)} = \frac{2^n}{2n}$. Так как по теореме код Хэмминга исправляет 1 ошибку, то $M_1(n) = \max m \ge \frac{2^n}{2n}$ ■.

Замечание: при $n = 2^{t-1}, t \in \mathbb{N}$ (например, 15, 31 итп) можно добиться верхней оценки $\frac{2^n}{n+1}$.

Линейные коды

Множество E_2^n можно рассматривать как линейное пространство над полем Галуа $GF(2) = \{0,1\}$ (операции по mod 2 покоординатно).

Опр. Линейным кодом порядка n называется любое линейное подпространство L в E_2^n.

Замечание: $\tilde{0}$ лежит в любом линейном коде.

Опр. Весом набора $\tilde{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n) \in E_2^n$ называют число единиц в $\tilde{\alpha}$.

Теорема. Кодовое расстояние линейного кода равно минимальному весу его ненулевых элементов.

Доказательство. Пусть дан линейный код C и пусть в нем кодовое расстояние ρ_{min}, а минимальный вес ненулевых наборов равен w_{min}. Докажем

1) В C существуют 2 разных набора $\tilde{\alpha}$ и $\tilde{\beta}$: $\rho(\tilde{\alpha}, \tilde{\beta}) = \rho_{min} \Rightarrow$
   в C есть набор $\tilde{\alpha} + \tilde{\beta}$ $(\tilde{\alpha} + \tilde{\beta} \neq 0)$ и $w(\tilde{\alpha} + \tilde{\beta}) = \rho(\tilde{\alpha}, \tilde{\beta}) = \rho_{min} \Rightarrow w_{min} \leq \rho_{min}$
2) В C есть ненулевой набор $\tilde{\gamma}$: $w(\tilde{\gamma}) = w_{min} \Rightarrow$ в C есть вектор $\tilde{0}$ и $\rho(\tilde{\gamma}, \tilde{0}) = w(\tilde{\gamma}) = w_{min} \Rightarrow \rho_{min} \leq \mathrm{w_{min}}$.

Из 1) и 2) вытекает, что $w_{min} = \rho_{min}$ ■.

Лекция 18. Автоматы. Часть 1.

Опр. Автоматом (инициальным) называется любая шестерка вида (A, B, Q, G, F, q_0), где

$A, B, Q -$ конечные алфавиты
$G: A \times Q \to Q$

$F: A \times Q \to B$

$q_0 \in Q$

$A -$ входной алфавит
$B -$ выходной алфавит

$Q -$ множество состояний

$G -$ функция перехода

$F -$ функция выхода
$q_0 -$ начальное состояние

Будем обозначать $A^\infty -$ бесконечные последовательности в алфавите A.

Автомат (A, B, Q, G, F, q_0) преобразует входные последовательности из A^∞ в выходные последовательности из B^∞. Пусть $x -$ переменная, описывающая вход $\Rightarrow$
$x = x(1)x(2)x(3) \ldots$
$y = y(1)y(2)y(3) \ldots$
$q = q(0)q(1)q(2) \ldots$

Мы потребуем, чтобы значения этих переменных были связаны равенствами:

$$\begin{cases} y(t) = F\big(x(t), q(t-1)\big), & t = 1,2,3 \ldots \\ q(t) = G\big(x(t), q(t-1)\big), & t = 1,2,3 \ldots \\ q(0) = q_0 \end{cases}$$

Эта система называется *канонические уравнения автомата*.

Пусть дан автомат (A, B, Q, G, F, q_0) и пусть дан вход $a = a(1)a(2)a(3) \ldots \in A^\infty$ $x(t) = a(t)\ \forall t = 1,2,3 \ldots$

Из второго уравнения системы канонических уравнений автомата определяются $q(1) = G\big(a(1), q(0)\big) = G\big(a(1), q(0)\big)$, затем определяется

$q(2) = G\big(a(2), q(1)\big)$ и т. д. $\Rightarrow$ определяются однозначно все $q(t) -$ состояния во все моменты времени $\Rightarrow$ однозначно определяются все значения $y(t) = F\big(a(t), q(t-1)\big)$ на выходе.

Замечание: если на вход подать только первые k значений входа $a = a(1)a(2) \ldots a(k)$, то на выходе однозначное определяются первые k значений. Следовательно, можно

считать, что автомат преобразует конечные входные слова в выходные слова той же длины.

Опр. Отображение φ: $A^\infty \to B^\infty$ называется *автоматной функцией*, если существует автомат, который его реализует.

Очевидно, что не всякое отображение является автоматом.

[Пример]

$A = B = Q = \{0,1\}$ и канонические уравнения вида:

$$\begin{cases} y(t) = q(t-1) \\ q(t) = x(t) \\ q(0) = 0 \end{cases}$$

$\boldsymbol{t}$	0	1	2	3	4	5
$\boldsymbol{x}$		$a(1)$	$a(2)$	$a(3)$	$a(4)$	$a(5)$
$\boldsymbol{q}$	0	$a(1)$	$a(2)$	$a(3)$	$a(4)$	$a(5)$
$\boldsymbol{y}$		0	$a(1)$	$a(2)$	$a(3)$	$a(4)$

То есть $a(1)a(2)a(3) \ldots \to 0a(1)a(2) \ldots -$ *автомат задержки.*

Автоматы можно задавать каноническими уравнениями, формулами и диаграммами Мура.

Диаграммы Мура

Опр. *Диаграммой Мура* для автомата (A, B, Q, G, F, q_0) называется ориентированный граф с множеством вершин Q. В этом графе для каждой пары $(a, q) \in A \times Q$ проводится дуга из вершины q в вершину $q' = G(a, q)$ и на дуге ставится пометка $\big(a, F(a, q)\big)$. Начальное состояние q_0 помечается в графе специальным образом. Диаграмма Мура однозначно задает автомат.

Но на практике наиболее важное представление автомата схемами.

Схемы из функциональных элементов

Опр. *Истоком* в ориентированном графе называется вершина, в которую не входит ни одной дуги (рис. 18.1).

Рис. 18.1. Исток

Опр. Ориентированный граф называется *ациклическим*, если в нем нет ориентированных циклов (простых циклов).

Утверждение. В любом ориентированном ациклическом графе есть хотя бы один исток.

Доказательство. Докажем больше, а именно, что в ориентированном ациклическом графе для любой вершины v существует исток a такой, что из a в v есть ориентированный путь. Будем двигаться из вершины v против направления дуг «до упора» (рис. 18.2) и достигнем истока, так как граф конечный, а вершины повторяться не могут, так как нет ориентированных циклов ■.

Рис. 18.2. Движение против направления дуг к истоку

Пусть задано некоторое множество функций алгебры логики:

$Б = \{g_1(\dots), g_2(\dots), \dots, g_k(\dots)\}$ – базис функциональных элементов.

Опр. Схемой из функциональных элементов (СФЭ) называется ориентированный ациклический граф, в котором

1) Каждому истоку приписана переменная x_i, причем разным истокам соответствуют разные переменные (истоки называют входы схемы, переменные x_i – входные переменные).
2) Каждой вершине v, в которую входит $m \geq 1$ дуг, приписана функция из базиса Б, зависящая ровно от m переменных, и переменные взаимно однозначно сопоставлены дугам, входящим в вершину v (вершина с приписанной функцией называется *функциональным элементом*).
3) Некоторые вершины выделены *как выходы* (входы могут быть выходами).
4) .

Опр. Глубиной вершины $d(v)$ в ориентированном ациклическом графе называется максимальное число дуг в ориентированных путях из истоков в эту вершину.

Утверждение. Если в ациклическом ориентированном графе есть дуга (u, v), то $d(v) > d(u)$.

Доказательство. Рассмотрим дугу из u в v (рис. 18.3). Существует путь из истока в v с числом дуг $d(u) + 1 \Rightarrow d(v) \geq d(u) + 1 \Rightarrow d(v) > d(u)$ ■.

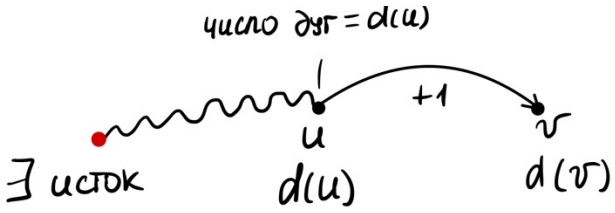

Рис. 18.3. Путь от истока до u и v

Определим функционирование СФЭ: индукции по глубине вершин определим для каждой вершины v функцию f_v от входных элементов, которая реализуется в вершине v.

Базис. Пусть $d(v) = 0 \Rightarrow v -$ исток $\Rightarrow$ по определению v приписана входная переменная $x_i \Rightarrow$ определим $f_v \equiv x_i$.

Индуктивный переход. Пусть функции f_v определены уже для всех v с $d(v) < d_0$, и пусть $d(w) = d_0 > 0$ (рис. 18.4). Тогда $\forall i \left(d(v_i) < d(w)\right)$ по утверждению $\Rightarrow$ по предположению индукции определены функции f_{v_i} $\forall i \Rightarrow$ по определению положим $f_w = g(f_{v_1}, \dots, f_{v_k})$. Таким образом, мы получили функцию, реализуемую в произвольной вершине.

Будем говорить, что СФЭ реализует набор функций, которые реализуются в выходах.

[Пример]

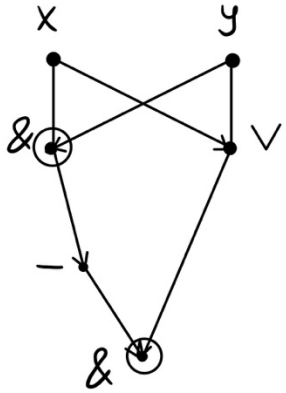

Рис. 18.4. Пример СФЭ

На схеме (рис. 18.4), выделенные вершины – выходы.

На первом выходе реализована функция $z_1 = x \cdot y$.

На втором выходе реализована функция $z_2 = \overline{x \cdot y} \cdot (x \vee y) = x \oplus y$.

Схематично эту СФЭ можно изобразить как на рис. 18.5. Эта схема называется *полусумматором.* В ней 4 функциональных элемента.

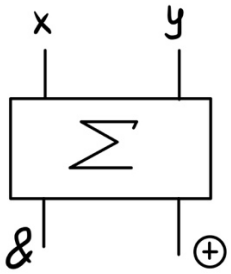

Рис. 18.5. Полусумматор

Формулы в базисе $Б = \{g_1(\dots), g_2(\dots), \dots, g_k(\dots)\}$ устроение рекурсивно: в каждую формулу можно поставить в качестве переменных формулы, полученные на предыдущем шаге. В этом смысле формула является корневым деревом (рис. 18.6). Если все дуги сориентировать снизу вверх, получим частный случай СФЭ (рис. 18.7).

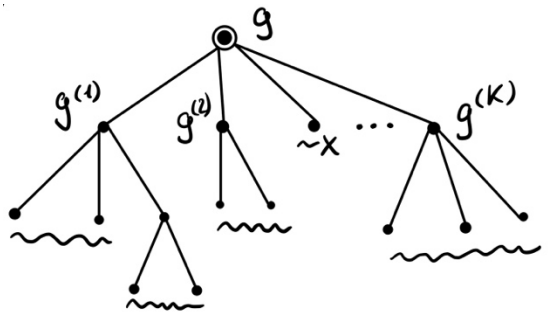

Рис. 18.6. Корневое дерево

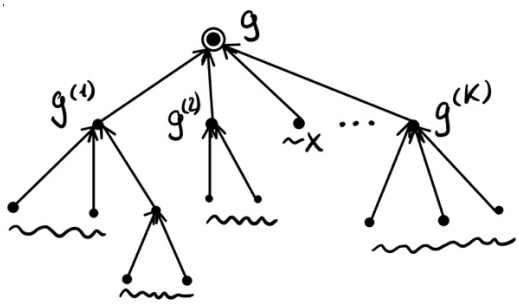

Рис. 18.7. СФЭ

Следовательно, формулы над Б – частный случай СФЭ в базисе Б.

Следствие. С помощью СФЭ в базисе $Б_0 = \{\vee, \wedge, -\}$ можно реализовать любую функцию алгебры логики.

Схема из функциональных элементов и задержек

Схемы из функциональных элементов и задержек (СФЭЗ) в базисе Б $= \{g_1(\dots), g_2(\dots), \dots, g_k(\dots)\}$ (g_i −функция алгебры логики) определяется так же, как СФЭ, но со следующими отличиями:

1) К Б добавляется элемент «задержка» (з) и если в ориентированном графе в вершину v входит ровно 1 дуга, то этой вершине может быть приписана либо функция от одной переменной из Б, либо задержка.
2) В ориентированном графе могут быть ориентированные циклы, но каждый ориентированный цикл должен проходить хотя бы через одну вершину, которой приписана задержка.

Функционирование СФЭЗ

Будем считать, что есть дискретное время $t = 0, 1, 2 \dots$ и в каждый момент времени $t = 1, 2 \dots$ на все входы поступают 0 или 1. Будем считать, что функциональные элементы срабатывают моментально в соответствии с приписанными им функциями из базиса Б, а элемент «задержка» работает так, как автомат «задержка».

Лекция 19. Автоматы. Часть 2.

Моделирование автомата СФЭ и СФЭЗ

Теорема. Любая СФЭЗ осуществляет автоматное отображение.

Доказательство. Пусть дана СФЭЗ, в которой r элементов задержки. Вершине v_i приписана задержка з. по определению задержка приписана вершине, в которую входит ровно 1 дуга из вершины w_i. Уберем из схемы дугу и задержку, тогда v_i станет истоком. припишем вершине v_i переменную $q_i{}'$, а вершину w_i пометим как выход и обозначим ее через q_i (рис.19.1). Если в вершине w_i уже был выход, то формально появится второй, хотя фактически они будут совпадать между собой.

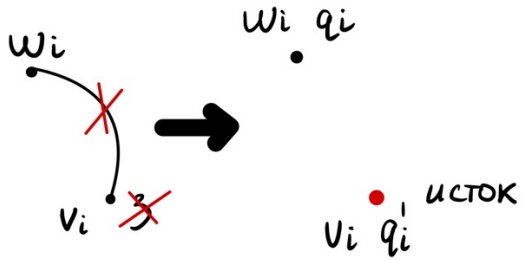

Рис. 19.1. Удаление задержки из схемы

Так поступаем с $\forall i = \overline{1,r}$. После этого преобразования в ориентированном графе не останется ориентированных циклов, так как по определению любой ориентированный цикл в СФЭЗ должен проходить через некоторую из вершин v_i, которым приписана задержка, а тогда и через дугу (w_i, v_i), которую мы выкинули. Тогда оставшийся ориентированный граф с пометками соответствует определению СФЭ.

Пусть в исходной СФЭЗ были входные переменные $x_1, \dots, x_n$, и выходы пусть были названы переменными $y_1, \dots, y_m$. Тогда в полученной СФЭ будут входные переменные $x_1, \dots, x_n, q_1', \dots, q_r'$ и выходные переменные $y_1, \dots, y_m, q_1, \dots, q_r$. Тогда по определению функционирования СФЭ для некоторых функция алгебры логики f_i, g_j выполняется:

$$\begin{cases} y_i = f_i(x_1, \dots, x_n, q_1', \dots, q_r'), \ i = \overline{1,m} \\ q_j = g_j(x_1, \dots, x_n, q_1', \dots, q_r'), \ j = \overline{1,r} \end{cases}$$

$$\begin{cases} y_i = f_i\big(x_1(t), \dots, x_n(t), q_1'(t), \dots, q_r'(t)\big), \ i = \overline{1,m} \\ q_j = g_j\big(x_1(t), \dots, x_n(t), q_1'(t), \dots, q_r'(t)\big), \ j = \overline{1,r} \end{cases}$$

В СФЭЗ для $\forall j\ q_j(t) = q_j(t-1)$ и тогда

$$\begin{cases} y_i = f_i\big(x_1(t), \dots, x_n(t), q_1(t-1), \dots, q_r(t-1)\big),\ i = \overline{1,m} \\ q_j = g_j\big(x_1(t), \dots, x_n(t), q_1(t-1), \dots, q_r(t-1)\big),\ j = \overline{1,r} \\ q_j(0) = 0 \end{cases} (*)$$ - канонические уравнения

для СФЭЗ.

Введем переменные:

$X = (x_1, \dots, x_n)$ принимает значения из E_2^n

$Y = (y_1, \dots, y_m)$ принимает значения из E_2^m

$Q = (q_1, \dots, q_k)$ принимает значения из E_2^k

Тогда систему (*) можно записать в виде (для некоторых функций F, G, не зависящих явно от t):

$$\begin{cases} Y(t) = F\big(X(t), Q(t-1)\big) \\ Q(t) = G\big(X(t), Q(t-1)\big) \\ Q(0) = (0,0,\dots,0) = \tilde{0} \end{cases}$$

Тогда отображение задается автоматом $(E_2^n, E_2^m, E_2^r, G, F, \tilde{0})$ ■.

Опр. Пусть заданы 2 автомата: $D_1 = (A_1, B_1, Q_1, G_1, F_1, q_{01})$ и $D_2 = (A_2, B_2, Q_2, G_2, F_2, q_{02})$. Пусть заданы отображения: $K_1: A_1 \to A_2$, $K_2: B_1 \to B_2$ причем разные элементы отображаются в разные. Будем говорить, что автомат D_2 *моделирует* автомат D_1 при отображениях K_1 и K_2, если для любой входной последовательности $a_1a_2a_3 \dots \in A^\infty$ выполняется:

Если D_1 отображает $a_1a_2a_3 \dots$ в $b_1b_2b_3 \dots \in B^\infty$, то D_2 отображает $K_1(a_1)K_1(a_2)K_1(a_3) \dots$ в $K_2(a_1)K_2(a_2)K_2(a_3) \dots$

В частности, СФЭЗ (как автомат) может моделировать другой автомат.

Теорема. Для любого автомата $D = (A, B, Q, G, F, q_0)$ существует СФЭЗ, которая моделирует D в базисе Б $= \{\&, \vee, -, з\}$

Доказательство. Пусть дан автомат $D = (A, B, Q, G, F, q_0)$ выберем натуральные числа n, m, r так, что $|A| \le 2^n, |B| \le 2^m, |Q| \le 2^r$.

Рассмотрим отображения (кодирования):

$K_1: A \to E_2^n$
$K_2: B \to E_2^m$
$K_3: Q \to E_2^r$

причем разные элементы отображается в разные и дополнительно потребуем, чтобы $K_3(q_0) = 0 \dots 0 = \tilde{0}$.

Рассмотрим функции G' и F' такие, что $G'\big(K_1(a), K_3(q)\big) = K_3\big(G(a,q)\big)$, $F'\big(K_1(a), K_3(q)\big) = K_2\big(F(a,q)\big)$ для всех $a \in A, q \in Q$.

Тогда автомат $(E_2^n, E_2^m, E_2^r, G', F', \tilde{0})$ моделирует автомат D.

$G'\big(K_1(a), K_3(q)\big) = K_3\big(G(a,q)\big)$, $F'\big(K_1(a), K_3(q)\big) = K_2\big(F(a,q)\big)$ определяют функции G' и F' не полностью - только на тех наборах, которые являются кодами, и на наборах, которые не соответствуют никаким кодам $K_1(a), K_3(q)$, доопределим их произвольно. Получили автомат $D_1 = (E_2^n, E_2^m, E_2^r, G', F', \tilde{0})$, моделирующий автомат D и G' и F' – всюду определенные функции алгебры логики.

Его канонические уравнения:

$$\begin{cases} Y(t) = F'\big(X(t), Q(t-1)\big) \\ Q(t) = G'\big(X(t), Q(t-1)\big) \quad (**) \\ Q(0) = (0,0,\dots,0) = \tilde{0} \end{cases}$$

Так как X принимает значения в E_2^n, то X можно рассматривать как $X = (x_1, \dots, x_n)$. Аналогично можно считать, что $Y = (y_1, \dots, y_m), Q = (q_0, \dots, q_r)$, где все x_i, y_j, q_l – булевские переменные.

Тогда (**) можно переписать в виде:

$$\begin{cases} y_i = f_i\big(x_1(t), \dots, x_n(t), q_1(t-1), \dots, q_r(t-1)\big), \ i = \overline{1,m} \\ q_j = g_j\big(x_1(t), \dots, x_n(t), q_1(t-1), \dots, q_r(t-1)\big), \ j = \overline{1,r} \\ q_j(0) = 0 \end{cases}$$

для некоторых функций алгебры логики f_i, g_j.

Далее по этой системе будем строить соответствующую СФЭЗ. Для этого сначала построим СФЭ в базисе $\{\vee, \&, -\}$ с $n + r$ входами и $m + r$ выходами, реализующую набор функций алгебры логики:

$$\begin{cases} y_i = f_i(x_1, \dots, x_n, q'_1, \dots, q'_r), \ i = \overline{1,m} \\ q_j = g_j(x_1, \dots, x_n, q'_1, \dots, q'_r), \ j = \overline{1,r} \end{cases}$$

Получим СФЭ (рис. 19.2):

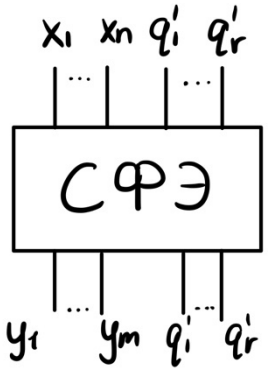

Рис. 19.2. СФЭ

Пусть в этой СФЭ входная переменная q'_j приписана вершине v_j, а выходная переменная q_j – вершине w_j. Добавим дугу (w_j, v_j) и сопоставим вершине v_j элемент задержки. Проделав это для всех пар q'_j, q_j $(j = \overline{1,r})$ получим СФЭЗ (рис. 19.3), функционирование которой описывается каноническими уравнениями:

$$\begin{cases} y_i = f_i\big(x_1(t), \dots, x_n(t), q_1(t-1), \dots, q_r(t-1)\big), & i = \overline{1,m} \\ q_j = g_j\big(x_1(t), \dots, x_n(t), q_1(t-1), \dots, q_r(t-1)\big), & j = \overline{1,r} \\ q_j(0) = 0 \end{cases}$$

Получим СФЭЗ, которая реализует автомат D_1, а значит моделирует автомат D ■.

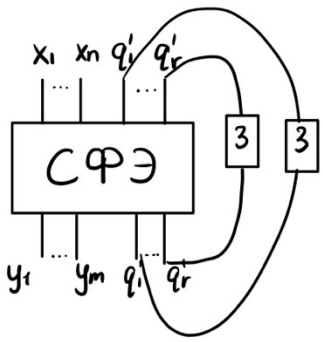

Рис. 19.3. СФЭЗ

Эксперименты с автоматами. Теорема Мура.

Рассмотрим автомат $D = (A, B, Q, G, F)$. Расширим функции G и F на $A^* \times Q$.

Пусть $\bar{a} \in A^*$ (слово в алфавите A) и пусть $q \in Q$.

Если начать вычисления автоматом D с состояния q и подать (по тактам) все слово $\bar{a} = (a_1 a_2 \dots a_k)$, то атвомат будет переходить на каждом такте из состояния в состояние

(по функции G) и в конце окажется в некотором состоянии $q' \in Q$, при этом на выходе образуется слово $\bar{b} \in B^*$длины k. Положим $F(\bar{a}, q) = \bar{b}$, $G(\bar{a}, q) = q'$.

Опр. Два состояния q_i и q_j в автомате $D = (A, B, Q, G, F)$ называются *отличимыми*, если существует входное слово $\bar{a} \in A^*$ такое, что $F(\bar{a}, q_i) \neq F(\bar{a}, q_j)$. В этом случае $\bar{a}$ называют экспериментом, отличающим q_i и q_j, а длину $\bar{a}$ называют *длиной эксперимента.*

Лекция 20. Теорема Мура

Рассмотрим автоматы без начального состояния. Наличие начального состояния означает что автомат *инициальный*. Ранее было дано определение что два состояния отличимы.

Теорема Мура

Опр. Два состояния автомата отличимы, если существует такое входное слово, подающее на автомат при начале работы находящийся в одном состоянии, будет одно выходное слово, а если подать это же начальное слово на автомат, находящийся в другом состоянии, будет другое выходное слово. При подаче одного и то же входного слова на автомат с разными начальными состояниями мы получим разные выходные слова. Тогда начальные состояния автомата называются отличимыми. Начальное слово в этом контексте называется экспериментом.

Теорема Мура. Пусть в автомате $\mathrm{D} = (\mathrm{A}, \mathrm{B}, \mathrm{Q}, \mathrm{G}, \mathrm{F})$ Число состояний $|\mathrm{Q}| = \mathrm{r}$, и пусть состояние q_i и q_j из Q отличны. Теорема утверждает, что тогда существует эксперимент (входное слово), который отличает q_i и q_j и имеет длину $\leq r - 1$, то есть достаточно просмотреть входные слова до длины $r - 1$. Если до этой длины всё совпало, то состояния неотличимы в принципе. Если состояния различимы на длине k, то они будут различимы на длине $k + 1$.

Доказательство.

Лемма. Пусть в автомате $\mathrm{D} = (\mathrm{A}, \mathrm{B}, \mathrm{Q}, \mathrm{G}, \mathrm{F})$ состояния q_u и q_v отличимы экспериментом длины p и не отличимы никаким экспериментом меньшей длины, то есть p самая маленькая длина входящего слова, при котором появляются различные состояния q_u и q_v. Утверждается, что тогда для любого натурального k: $1 \leq k \leq \mathrm{p}$ существуют два состояния в множестве Q, которые отличимы экспериментом длины k и не отличимы никаким более коротким экспериментом.

Доказательство. По условию $\exists$ входное слово $\bar{a} = a_1 \dots a_p \in A^*$ длины p, такое что выходное слово

$F(\bar{a}, q_u) \neq F(\bar{a}, q_v)$. Пусть $F(\bar{a}, q_u) = \bar{b} = b_1 b_2 \dots b_p$ и $F(\bar{a}, q_v) = \bar{c} = c_1 c_2 \dots c_p \Rightarrow \bar{b} \neq \bar{c}$

Причём входные слова $< p$ не отличают по условию q_u и q_v, то $\bar{b}$ и $\bar{c}$ различаются только в последнем символе ($b_p \neq c_p$). Пусть зафиксировано k: $1 \leq k \leq \mathrm{p}$ Разобьём все слова $\bar{a}, \bar{b}, \bar{c}$ на две части: $p - k$ и p. $\bar{a} = \overline{a_1}\,\overline{a_2}$, $\bar{b} = \overline{b_1}\,\overline{b_2}$, $\bar{c} = \overline{c_1}\,\overline{c_2}$ и $\text{дл}(\overline{a_2}) = \text{дл}(\overline{b_2}) = \text{дл}(\overline{c_2}) = k$ т.е. с конца отрезаем k символов причём $\overline{a_1}, \overline{b_1}, \overline{c_1}$ могут быть пустыми.

Введём два состояния. Пусть $G(\overline{a_1}, q_u)$ т.е. при состоянии q_u подаем не все входящее слово целиком, только первую его часть $\overline{a_1}$, тогда $G(\overline{a_1}, q_u) = q', (\overline{a_1}, q_v) = q''$ Что будет на выходе если $\Rightarrow F(\overline{a_2}, q') = \overline{b_2}$, $F(\overline{a_2}, q'') = \overline{c_2}$. Поскольку слова $\overline{b}$ и $\overline{c}$ отличаются только в последнем символе, $\overline{b_2} \neq \overline{c_2}$ т.к. $\overline{b_p} \neq \overline{c_p}$ то получаем входное слово $\overline{a_2}$ длины k которое отличает состояния q' и q''.

Было сказано, что для любого k найдутся два состояния которые отличимы длиной k.

Докажем от противного, что q' и q'' не отличимы экспериментом меньшей длиной k. Допустим, что q' и q'' отличимы экспериментом $\overline{a_3}$ с дл$(\overline{a_3}) < k$.

Пусть $F(\overline{a_3}, q') = \overline{b_3}$ и $F(\overline{a_3}, q'') = \overline{c_3} \Rightarrow \overline{b_3} \neq \overline{c_3}$ Посмотрим, что получиться при применении слова $F(\overline{a_1}\ \overline{a_3}, q_u) = \overline{b_1} F(\overline{a_3}, q') = \overline{b_1}\ \overline{b_3}$ аналогично $F(\overline{a_1}\ \overline{a_3}, q_v) = \overline{c_1} F(\overline{a_3}, q'') = \overline{c_1}\ \overline{c_3}$ поскольку $\overline{b_3} \neq \overline{c_3} \Rightarrow \overline{b_1}\ \overline{b_3} \neq \overline{c_1}\ \overline{c_3} \Rightarrow$ слово $\overline{a_1}\ \overline{a_3}$ отличает состояния q_u и q_v и

дл $(\overline{a_1}\ \overline{a_3}) =$ дл $(\overline{a_1})$ + дл $(\overline{a_3}) = p - k +$ дл $(\overline{a_3}) < p$ Это противоречит условию в том, что для пары q_u и q_v p это кратчайшая длина эксперимента, отличающего эти состояния.

$\Rightarrow$ (от противного) не $\exists$ эксперимента различающего q' и q'' с длиной $< k$ ■.

Доказательство Теоремы.

По условию q_i и q_j отличимы. Пусть самый короткий отличающий их эксперимент имеет длину p. Нас интересует чему равна длина самого короткого эксперимента. Докажем, что $p \leq r - 1$

Рассмотрим бинарные отношения на множестве Q.

$R_m(q_s, q_t) \equiv q_s$ и q_t не отличимы никаким экспериментом длины m (а значит и более коротким экспериментом) $m = 0,1,2, \dots p$

$R_0 \equiv'$ истинно$' \Rightarrow$ Для любого m $R_m -$ отношение эквивалентности

Рассмотрим разбиение множества Q на классы эквивалентности относительно R_m:

$Q_1^m, Q_2^m, Q_3^m, \dots, Q_{s(m)}^m \Rightarrow$ s$(0) = 1$, то есть один класс эквивалентности (все эквивалентны)

Если два состояния попали в один класс, это означает что они неотличимы экспериментом длиной m. Если два состояния попали в разные классы, это означает что они отличимы экспериментом длиной m.

Посмотрим, что происходит при переходе от m к $m + 1$. Если два состояния отличимы экспериментом длины m, то они отличимы и экспериментом длины $m + 1$. $\Rightarrow$ состояния из разных классов для R_m остаются в разных классах и для R_{m+1} Может произойти, что

один класс распадётся на несколько классов, но склеивания не произойдёт. Вопрос: распадётся или нет?

По Лемме: для $\forall\, m: 0 \le m \le p-1$ $\exists$ 2 состояния отличимой экспериментом длиной $m+1$ и не отличимы более коротким экспериментом $\Rightarrow$ при переходе от R_m к R_{m+1} по крайней мере один из классов эквивалентности распадётся на ≥ 2 класса $\Rightarrow s(m+1) > s(m)$ при $m = 0,1,2,\dots,p-1$

$1 = s(0) < s(1) < s(2) < \cdots < s(p-1) < s(p) \le r$ и все $s(m) -$ натуральные числа

$\Rightarrow p+1 \le r \;\; \Rightarrow p \le r-1$ ■.

Фактически мы рассмотрели ситуацию, когда подошли к черному ящику и мы знаем, что автомат находится или в Q_i или Q_j состоянии. Вопрос можно ли распознать в каком именно состоянии находится автомат? У нас есть понятия отличимости состояний при эксперименте с конечной длиной мы можем определить состояние.

Теперь другая постановка. Мы подходим к черному ящику. Автомат полностью известен, известна его диаграмма или уравнение. Неизвестно в каком состоянии он находится. Вопрос: можем ли мы подать такой эксперимент, который бы определил начальное состояние автомата в момент, когда мы подошли к черному ящику. На этот вопрос отвечает следующая теорема.

Теорема. $\exists$ автомат с тремя состояниями, в котором каждая пара состояний отличима, но не существует единого эксперимента, который бы однозначно определял начальное состояние автомата.

Рассмотрим пример в качестве **доказательства.** Возьмём автомат (рис. 20.1) с бинарным входом и бинарным выходом. $A = B = \{0,1\}$

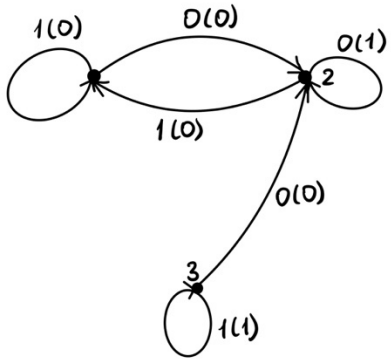

Рис. 20.1. Диаграмма автомата

На входы автомата нужно подать эксперимент чтобы на выходе состояния изменились

Входы	Эксперимент	Прим
1 и 2	0	Состояния отличимы по входу, если подадим эксперимент равный 0, то выходы будут разными.
1 и 3	1	
2 и 3	0	

У нас есть «черный ящик», в котором реализован этот автомат и всё что мы можем это подавать на вход какое-то одно слово и по выходу должны определять начальные состояния автомата.

Это простой эксперимент, не рокадный, мы не можем изменить начальные состояния автомата. При рокадном эксперименте у нас были бы три автомата с одинаковыми начальными условиями, и мы на каждый автомат подавали бы разный эксперимент. Мы же должны за один эксперимент определить начальные состояния автомата. Нам нужно придумать такое входное слово, если автомат находился в одном состоянии, то выход был один, а если в другом состоянии, то выход был бы другой, если в третьем состоянии – выход третий. Слово должно определять всю тройку входов сразу и это слово изменяет состояние всех трёх выходов.

Рассмотрим два варианта.

Если входное слово начинается с 0, то этот эксперимент не отличает состояние 1 и 3

Если входное слово начинается с 1, то этот эксперимент не отличает состояние 1 и 2

Нет такого слова чтобы выходные слова были разные ■.

Схемный сумматор порядка n

Опр. *Сумматором порядка* n называется схема (СФЭ) с $2n$ входами $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n$ и $n+1$ выходами $z_0, z_1, .., z_n$ такая, что при любых значениях (булевских) входных переменных $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n$ выполняется: $(z_0, z_1, .., z_n)_2 = (x_1, x_2, \dots, x_n)_2 + (y_1, y_2, \dots, y_n)_2$

Теорема. $\exists$ схема функциональных элементов (СФЭ) в стандартном базисе $\{\vee, \&, -\}$ которая является сумматором и имеет сложность (число функциональных элементов) $\leq 9n - 5$

Доказательство. Сложим в столбик два двоичных числа и определим алгоритм сложения

q_i	q_0	q_1	q_2		q_n
x_i		x_1	x_2	…	x_n
	+				
y_i		y_1	y_2	…	y_n
	z_0	z_1	z_2	…	z_n

где q – разряд переноса

$z_i = q_i \oplus x_i \oplus y_i$ — это сложение по модулю 2

$q_{i-1} = m(x_i, y_i, q_i) = x_i y_i \vee x_i q_i \vee y_i q_i$ при условии $1 \le i \le n-1$

$z_n = x_n \oplus y_n$, $z_0 = q_0$

$q_{i-1} = x_n \cdot y_n$

Рассмотрим схему полусумматора (рис. 20.2):

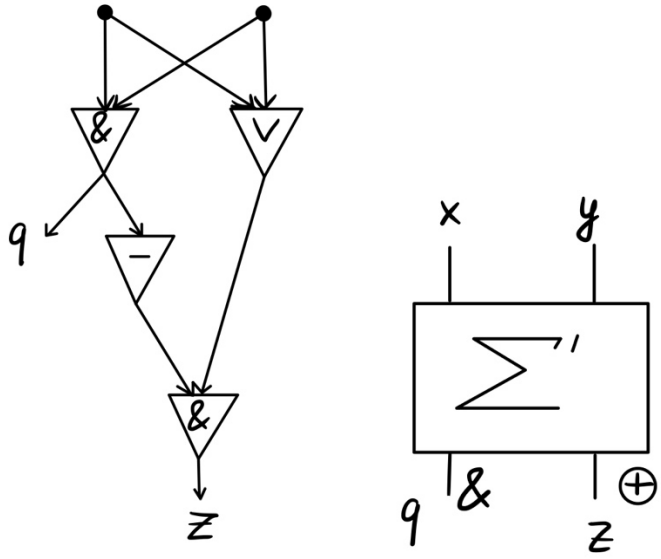

Рис. 20.2. Полусумматор

Построим схему ячейки сумматора (рис. 20.3). Для этого возьмем 2 полусумматора и конъюнкцию:

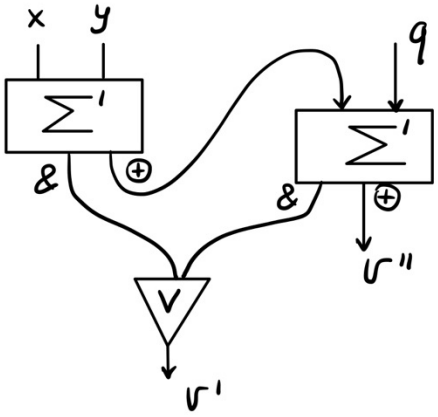

Рис. 20.3. Ячейка сумматора

Реализуем функции:

$V'' = x \oplus y \oplus q$

$V' = xy \vee (x \oplus y)q = xy \vee (x \vee y)q = xy \vee xq \vee yq = m(x, y, q)$

Ячейку сумматора Σ_1 схематично изобразим так – рис. 20.4:

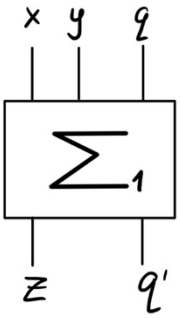

Рис. 20.4. Схема ячейки сумматора

$L(\Sigma_1) = 9$ – сложность ячейки сумматора (9 функциональных элементов).

Тогда сумматор порядка N можно построить следующим образом (рис. 20.5): как и при сложении, справа-налево соединим полусумматоры и ячейки сумматоры. На входе у полусумматора x_nи y_n, на выходе - z_n и перенос, который уходит в следующий разряд, то есть на вход следующей ячейки сумматора и так далее.

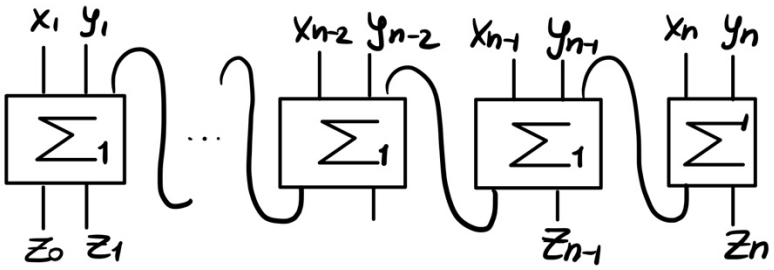

Рис. 20.5. Схема сумматора порядка N

Сложность сумматора S_n - $L(S_n) = (n-1)9 + 4 = 9n - 5$ ■.

Лекция 21. Умножитель порядка N.

Вычитатель порядка N

Опр. *Вычитателем порядка* n называется схема (СФЭ) с $2n$ входами $x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n$ и n выходами $z_1, z_2, \dots, z_n$ такими, что для всех значений входов такихз, что $(x_1 x_2 \dots x_n)_2 \geq (y_1 y_2 \dots y_n)_2$ выполняется:

$$(z_1 z_2 \dots z_n)_2 = (x_1 x_2 \dots x_n)_2 - (y_1 y_2 \dots y_n)_2$$

Теорема. $\exists$ СФЭ в базисе $\{\vee, \&, -\}$, которая является вычитателем и имеет сложность (число функциональных элементов) $\leq 11n - 5$.

Доказательство.

Обозначим $(x_1 x_2 \dots x_n)_2 = X, (y_1 y_2 \dots y_n)_2 = Y$.

Тогда $X - Y = (2^n - 1) - \big((2^n - 1) - X\big) + Y)$ $(*)$.

Утверждение. $2^n - 1 - (x_1 x_2 \dots x_n)_2 = (\bar{x}_1 \bar{x}_2 \dots \bar{x}_n)_2$.

Доказательство. Сложим столбиком $(x_1 x_2 \dots x_n)_2 + (\bar{x}_1 \bar{x}_2 \dots \bar{x}_n)_2$:

$$\begin{array}{ll} & \bar{x}_1 \bar{x}_2 \dots \bar{x}_n \\ + & x_1 x_2 \dots x_n \qquad \blacksquare. \\ & (11 \dots 11)_2 = 2^n - 1 \end{array}$$

Построим вычитатель по (*) – рис. 21.1:

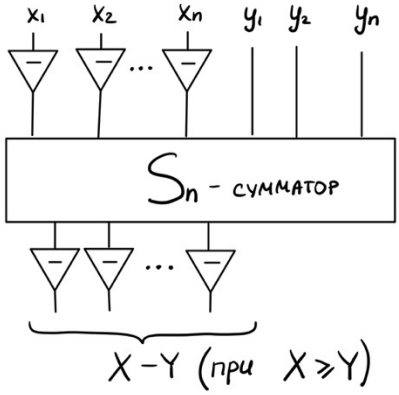

Рис. 21.1. Схема вычитателя порядка N

По условию входы, на которых СФЭ вычитателя правильно работает, удовлетворяют условию: $X \geq Y$.

Тогда $(2^n - X) + Y = 2^n - 1 - \underbrace{(X - Y)}_{\geq 0} \leq 2^n - 1$.

Сложность вычитателя порядка n $L = L(S_n) + 2n \leq 9n - 5 + 2n = 11n - 5$ ■.

Умножитель порядка N

Опр. *Умножителем порядка* n называется схема (СФЭ) с с $2n$ входами $x_1, x_2, \dots, x_n, y_1, y_2, .., y_n$ и $2n$ выходов $z_1, z_2, \dots, z_{2n}$ такая, что для ∀ значений входов выполняется $(z_1 z_2 \dots z_{2n})_2 = (x_1 x_2 \dots x_n)_2 \cdot (y_1 y_2 \dots y_n)_2$.

Выходов $2n$, так как

$0 \leq X \leq 2^n - 1 \leq 2^n$

$0 \leq Y \leq 2^n - 1 \leq 2^n$

$0 \leq XY \leq 2^{2n}$

Алгоритм умножения в столбик

Алгоритм умножения в столбик требует $\geq n^2$ (по порядку).

Теорема (Карацуба А.А.). ∃ умножитель порядка n (СФЭ в базисе $\{\vee, \&, -\}$ со сложностью (числом функциональных элементов) $O(n^{\log_2 3})$.

Доказательство. Обозначим через $M(n) -$ минимальное число элементов умножителя порядка n в фиксированном базисе $\{\vee, \&, -\}$.

Лемма 1 (основная). ∃ $const\ c_1$ такая, что для $\forall n \in \mathbb{N}$ выполняется

$$M(2n) \leq 3M(n) + c_1 n$$

Доказательство Л1. Пусть даны 2 числа:

$$X = \left(\underbrace{x_1 x_2 \dots x_n}_{X_1} \underbrace{\dots x_{2n}}_{X_2}\right),\ Y = \left(\underbrace{y_1 y_2 \dots y_n}_{Y_1} \underbrace{\dots y_{2n}}_{Y_2}\right)$$

$$X_1 = (x_1 \dots x_n)_2, \quad X_2 = (x_{n+1} \dots x_{2n})_2;\ Y_1 = (y_1 \dots y_n)_2, \quad Y_2 = (y_{2n+1} \dots y_{2n})_2$$

$$X \cdot Y = (X_1 \cdot 2^n + X_2) \cdot (Y_1 \cdot 2^n + Y_2) = X_1 Y_1 2^{2n} + (X_1 Y_2 + X_2 Y_1) 2^n + X_2 Y_2 =$$
$$= X_1 Y_1 2^{2n} + \underbrace{[(X_1 + X_2)(Y_1 + Y_2) - X_1 Y_1}_{X_1 Y_2 + X_2 Y_1 \geq 0} - X_2 Y_2] + X_2 Y_2$$

Если строить СФЭ по этой формуле, то потребуется 3 умножителя (2 из них порядка n и 1 из них порядка $n+1$) и конечное число сумматоров и вычитателей порядка $\leq 4n$, так как $X \cdot Y \leq 2^{4n}$.

Поскольку у сумматора и вычитателя порядка p сложность $< 11p$, то $\exists\, const = c_2$ и суммарная сложность всех сумматоров и вычитателей $\leq c_2 n$.

Тогда $M(2n) \leq M(n) + M(n) + M(n+1) + c_2 n$.

Лемма 1.1. $\exists\, const\ c_2 > 0$ такая, что для $\forall n \in \mathbb{N}\ M(n+1) \leq M(n) + c_3 n$.

Доказательство Л 1.1. Пусть $X = \left(x_0\ \underbrace{x_1 \dots x_n}_{X_1}\right)_2$, $Y = \left(y_0\ \underbrace{y_1 \dots y_n}_{Y_1}\right)_2$ — два $n+1-$ разрядных числа. Положим $X_1 = (x_1 \dots x_n)_2$, $Y_1 = (y_1 \dots y_n)_2$.

Тогда $X \cdot Y = (x_0 \cdot 2^n + X_1)(y_0 \cdot 2^n + Y_1) = x_0 y_0 2^{2n} + (x_0 Y_1 + y_0 X_1)2^n + X_1 Y_1$.

Утверждение. Для умножения $n-$разрядного числа на 1-разрядное достаточно n элементов.

Доказательство. Умножая в столбик, получим

$$\begin{array}{r} x_1 \dots x_n \\ \times \qquad y \\ \hline z_1 \dots . z_n \end{array}, \quad z_i = x_i \cdot y_i,\ i = \overline{1,n}\ \blacksquare.$$

$M(n+1) \leq M(n) + 2n + 1 + \underbrace{c_4 n}_{\text{все сумматоры}} \leq M(n) + c_3 n$. Лемма 1.1 доказана ■.

Замечание. На старшие разряды сумматоров в Лемме 1 и Лемме 1.1 может потребоваться подавать нули (для старших разрядов, так как числа на входе могут быть меньше порядка сумматора). Для этого вначале строим схему (рис. 21.2) и ее выход подаем нужные разряды (входы). Сложность схем в таком случае возрастает на 2, но это не влияет на оценки в леммах 1 и 1.1 (просто изменим константу c_3).

Рис. 21.2. Схема тождественного 0

Продолжение доказательство леммы 1:

По лемме 1.1

$M(2n) \le M(n) + M(n) + M(n+1) + c_2 n \le 3M(n) + c_2 n + c_3 n = 3M(n) + c_1 n$

Лемма 1 доказана ■.

Докажем сначала оценку $O(n^{\log_2 3})$ из теоремы для под последовательности n вида $n = 2^k, k \in \mathbb{N}$.

При $n = 2^k$: $n^{\log_2 3} = (2^k)^{\log_2 3} = \left(2^{\log_2 3}\right)^k = 3^k$.

Лемма 2. $\exists\, const\ c_5$ такая, что для всех $n = 2^k, k \in \mathbb{N}$ выполняется $M(n) \le c_5 \cdot n^{\log_2 3}$.

Доказательство Л 2. Надо для всех $n = 2^k, k \in \mathbb{N}$ получить оценку $M(2^k) \le c_5 \cdot 3^k$. Введем функцию $f(k) = \frac{M(2^k)}{3^k}$.

Из Леммы 1 получаем:

$M(2^k) \le 3M(2^{k-1}) + c_1 2^{k-1} \ \ /: 3^k$

$\frac{M(2^k)}{3^k} \le \frac{M(2^{k-1})}{3^{k-1}} + \frac{c_1}{2}\left(\frac{2}{3}\right)^k$

$\Rightarrow f(k) \le f(k-1) + \frac{c_1}{2}\left(\frac{2}{3}\right)^k$ для $\forall k \in \mathbb{N}$

$$\Rightarrow f(k) \le f(k-2) + \frac{c_1}{2}\left(\frac{2}{3}\right)^{k-1} + \frac{c_1}{2}\left(\frac{2}{3}\right)^k \le f(k-3) + \frac{c_1}{2}\left(\frac{2}{3}\right)^{k-2} + \frac{c_1}{2}\left(\frac{2}{3}\right)^{k-1} + \frac{c_1}{2}\left(\frac{2}{3}\right)^k$$

$$\le \cdots \le f(1) + \frac{c_1}{2}\left[\underbrace{\left(\frac{2}{3}\right)^2 + \left(\frac{2}{3}\right)^3 + \cdots + \left(\frac{2}{3}\right)^k}_{\text{так как сходящийся ряд}}\right] \le c_5$$

$\Rightarrow f(k) \le c_5$ для $\forall k \in \mathbb{N}$

$\Rightarrow \frac{M(2^k)}{3^k} \le c_5 \ \Rightarrow M(2^k) \le c_5 3^k \ \Rightarrow M(n) \le c_5 n^{\log_2 3}$ лемма 2 доказана ■.

Продолжение доказательства теоремы Карацубы:

$\forall n \in \mathbb{N}\ \exists\, k \in \mathbb{N} \cup \{0\}$: $2^k - 1 \le n \le 2^k \ \Rightarrow M(n) \le M(2^k) \underbrace{+2}_{\text{из−за 0 для старших разрядов}} \le$

$\le \{\text{по лем. 2}\} \le c_5 3^k + 2 = 3c_5 3^{k-1} + 2 = 3c_5\left(2^{\log_2 3}\right)^{k-1} + 2 = 3c_5\left(\underbrace{2^{k-1}}_{<n}\right)^{\log_2 3} + 2 <$

$< 3c_5 n^{\log_2 3} + 2 \le c_1 n^{\log_2 3}$ ■.

ФАКУЛЬТЕТ
ВЫЧИСЛИТЕЛЬНОЙ
МАТЕМАТИКИ И
КИБЕРНЕТИКИ
МГУ ИМЕНИ
М.В. ЛОМОНОСОВА

teach-in

ЛЕКЦИИ УЧЕНЫХ МГУ