Содержание

1 Часть А.

1.1 Функции алгебры логики. Полиномы Жегалкина. Быстрый алгоритм построения полинома Жегалкина функции алгебры логики (с обоснованием).

Опр. *Пусть* $E_2 = \{0, 1\}$. *Функцией алгебры логики называется произвольное отображение из* E_2^n *в* E_2, $n \geq 1$. *Множество всех функций алгебры логики, зависящих от* n *переменных, обозначит* $P_2^{(n)}$, *а множество всех функций алгебры логики —* $P_2 = \bigcup\limits_{n \geq 1} P_2^{(n)}$.

Опр. *Элементарная конъюнкция, не содержащая отрицаний переменных, называется монотонной ЭК, или мономом, или одночленом.*

Опр. *Полиномом Жегалкина длины* l, $l \geq 1$, *назовем сумму по модулю два* l *различных монотонных ЭК. Полиномом Жегалкина длины* 0 *назовем константу* 0.

Теорема. *Каждая функция* $f(x_1, \ldots, x_n) \in P_2$ *может быть единственным образом представлена в виде полинома Жегалкина* P_f.

Д-во. Существование. Применим полиномиальное разложение функции $f(x_1, \ldots, x_n)$ по всем n переменным:

$$f(x_1, \ldots, x_n) = \bigoplus_{\sigma \in E_2^n} x_1^{\sigma_1} \cdot \cdots \cdot x_n^{\sigma_n} \cdot f(\sigma).$$

Затем пользуясь тождеством $x^\sigma = x \oplus \sigma \oplus 1$ везде в правой части заменим выражение $x_i^{\sigma_i}$ на выражение $x_i \oplus \sigma_i \oplus 1$. Далее по правилам коммутативности и ассоциативности & и $\oplus$ и дистрибутивности вида $x \cdot (y \oplus z) = x \cdot y \oplus z \cdot z$ перемножим все скобки. После этого приведем подобные слагаемые по правилам $x \oplus x = x$, $x \oplus 0 = x$. В итоге получим полином Жигалкина, который представляет исходную функцию f.
Единственность. Покажем, что число полиномов Жегалкина над переменными $x_1, \ldots, x_n$ совпадает с числом функций из $P_2^{(n)}$. Монотонных элементарных конъюнкций над переменными $x_1, \ldots x_n$ всего найдется 2^n, т.к. каждая переменная x_i, $i = \overline{1, n}$, может либо входить, либо не входить в такую монотонную ЭК. Далее, полиномов Жегалкина над переменными $x_1, \ldots, x_n$ всего найдется 2^{2^n}, т.к. каждая из 2^n монотонных ЭК может либо входить, либо не входить в такой полином Жегалкина. Значит, учитывая то, что каждая функция f из $P_2^{(n)}$ может быть представлена в виде полинома Жегалкина, это представление единственно. □

Набору $\alpha \in E_2^n$, ≥ 2, взаимно однозначно сопоставим монотонную ЭК над переменными $x_1, \ldots, x_n$:

$$x^\alpha = \begin{cases} 1 & , \alpha = (0, \ldots, 0) \\ \prod\limits_{\alpha_i = 1} x_i & , \alpha \neq (0, \ldots, 0) \end{cases}.$$

Если α пробегает по всем возможным наборам из E_2^n, то x^α перечисляет все возможные монотонные ЭК над $x_1, \ldots, x_n$.

Пусть $c_f(\alpha)$ обозначает коэффициент при мономе $x^\alpha,\ \alpha \in E_2^n$, в полиноме Жегалкина функции $f \in P_2^{(n)}$. Тогда

$$f(x_1, \ldots, x_n) = \bigoplus_{\alpha \in E_2^n} c_f(\alpha) \cdot x^\alpha.$$

Для нахождения полинома Жегалкина функции f нужно найти коэффициенты $c_f(\alpha)$ для всех $\alpha \in E_2^n$.

Вычисление коэффициентов при $n = 1$.
Если $f(x) \in P_2^{(1)}$, то

$$\begin{aligned} f(x) &= \overline{x} \cdot f(0) \oplus x \cdot f(1) = (x \oplus 1) \cdot f(0) \oplus x \cdot f(1) = \\ &= x \cdot f(0) \oplus f(0) \oplus x \cdot f(1) = (f(0) \oplus f(1)) \cdot x \oplus f(0). \end{aligned}$$

Поэтому $c_f(0) = f(0),\ c_f(1) = f(0) \oplus f(1)$.

Теорема. *Если $n \geq 1$, $f(y, x_1, \ldots, x_n) \in P_2^{(n+1)}$, $f_a(x_1, \ldots, x_n) = f(a, x_1, \ldots, x_n)$, где $a \in E_2$, то для каждого $\alpha \in E_2^n$ верны равенства:*

$$\begin{aligned} c_f(0, \alpha_1, \ldots, \alpha_n) &= c_{f_0}(\alpha), \\ c_f(1, \alpha_1, \ldots, \alpha_n) &= c_{f_0}(\alpha) \oplus c_{f_1}(\alpha). \end{aligned}$$

Д-во. Применим полиномиальное разложение функции $f(y, x_1, \ldots, x_n)$ по переменной y:

$$\begin{aligned} f(y, x_1, \ldots, x_n) &= \overline{y} \cdot f(0, x_1, \ldots, x_n) \oplus y \cdot f(1, x_1, \ldots, x_n) = \\ &= \overline{y} \cdot f_0 \oplus y \cdot f_1 = (y \oplus 1) \cdot f_0 \oplus y \cdot f_1 = \\ &= y \cdot f_0 \oplus f_0 \oplus y \cdot f_1 = y(f_0 \oplus f_1) \oplus f_0. \end{aligned}$$

Но

$$\begin{aligned} f_0 &= \bigoplus_{\alpha \in E_2^n} c_{f_0}(\alpha) \cdot x^\alpha, \\ f_1 &= \bigoplus_{\alpha \in E_2^n} c_{f_1}(\alpha) \cdot x^\alpha. \end{aligned}$$

Поэтому:

$$f = y \cdot \left(\bigoplus_{\alpha \in E_2^n} c_{f_0}(\alpha) \cdot x^\alpha \oplus \bigoplus_{\alpha \in E_2^n} c_{f_1}(\alpha) \cdot x^\alpha \right) \oplus \bigoplus_{\alpha \in E_2^n} c_{f_0}(\alpha) \cdot x^\alpha.$$

Значит,

$$f = \bigoplus_{\alpha \in E_2^n} (c_{f_0}(\alpha) \oplus c_{f_1}(\alpha)) \cdot y \cdot x^\alpha \oplus \bigoplus_{\alpha \in E_2^n} c_{f_0}(\alpha) \cdot x^\alpha.$$

Перепишем следующим образом:

$$f = \bigoplus_{(1,\alpha) \in E_2^{n+1}} (c_{f_0}(\alpha) \oplus c_{f_1}(\alpha)) \cdot (y^1 \cdot x^\alpha) \oplus \bigoplus_{(0,\alpha) \in E_2^{n+1}} c_{f_0}(\alpha) \cdot (y^0 \cdot x^\alpha).$$

Из полученного находим:

$$c_f(0, \alpha_1, \ldots, \alpha_n) = c_{f_0}(\alpha),$$
$$c_f(1, \alpha_1, \ldots, \alpha_n) = c_{f_0}(\alpha) \oplus c_{f_1}(\alpha).$$

□

Пользуясь формулами предыдущей теоремы, найдем Жегалкина функции $f(x_1, x_2, x_3)$:

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

На шаге 1 вычисляем коэффициенты полиномов Жегалкина всех подфункций $f_\sigma(x_3),\ \sigma \in E_2^2$, функции $f(x_1, x_2, x_3)$ по переменным x_1, x_2.

x_1	x_2	x_3	f	1
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	0

На шаге 2, пользуясь полученными значениями на шаге 1, вычисляем коэффициенты полиномов Жегалкина всех подфункций $f_\delta(x_2, x_3),\ \delta \in E_2^1$, функции $f(x_1, x_2, x_3)$ по переменной x_1:

x_1	x_2	x_3	f	1	2
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	0	1

Наконец, на шаге 3, пользуясь полученными значениями на шаге 2, вычисляем коэффициенты полиномов Жегалкина функции $f(x_1, x_2, x_3)$:

x_1	x_2	x_3	f	1	2	3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	0	1	0

Получаем:

$$f(x_1, x_2, x_3) = x_2x_3 \oplus x_1x_3 \oplus x_1x_2.$$

1.2 Функции алгебры логики. Двойственность. Самодвойственные функции. Замкнутость класса самодвойственных функций.

Опр. *Функция $f^*(x_1, \ldots, x_n) \in P_2$ называется двойственной к функции $f(x_1, \ldots, x_n) \in P_2$, если*

$$f^*(x_1, \ldots, x_n) = \overline{f(\overline{x}_1, \ldots, \overline{x}_n)}.$$

Отметим, что для любой функции $f \in P_2$ верно $(f^*)^* = f$.

Опр. *Функция $f(x_1, \ldots, x_n) \in P_2$ называется самодвойственной, если $f^*(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$. Заметим, что данное определение эквивалентно тому, что функция f на всех парах противоположных наборов принимает противоположные значения, т.е. $\forall \alpha \in E_2^n : f(\overline{\alpha}) = \overline{f(\alpha)}$. Множество всех самодвойственных функций обозначим S.*

Лемма. *Пусть $A \subseteq P_2$ и $I \subseteq A$. Если для любых функций $f_0(y_1, \ldots, y_m) \in A$, $f_i(x_1, \ldots, x_n) \in A$, $i = \overline{1, m}$, причем функции f_i могут зависеть несущественно от некоторых своих переменных, верно*

$$f_0(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)) \in A,$$

то множество A является замкнутым классом.

Д-во. Рассмотрим произвольную функцию $f \in [A]$. Она выражается некоторой формулой F над множеством A. Докажем индукцией по числу d вхождений в формулу F обозначений функций из $A \setminus I$, что $f \in A$.
1. Базис индукции: $d = 0$. Если $F = x_i$, то $f \in A$ по условию утверждения.
2. Индуктивный переход: пусть любая функция, которая может быть вырождена формулой не более чем с d_0 вхождениями обозначений функции из $A \setminus I$, содержится в A. Рассмотрим функцию $f(x_1, \ldots, x_n) \in [A]$, которая выражается формулой F с $d_0 + 1$ вхождениями обозначений функций из $A \setminus I$. Тогда $f = f_0(F_1, \ldots, F_m)$, где $f_0 \in A \setminus I$, F_i — формулы не более чем с d_0 вхождениями функций из $A \setminus I$. По предположению индукции $f_{F_i} \in A \implies$

$$f(x_1, \ldots, x_n) = f_0(f_{F_1}(x_1, \ldots, x_n), \ldots, f_{F_m}(x_1, \ldots, x_n))..$$

Далее, по условию утверждения $f \in A$. □

Теорема. *Множество S является замкнутым классом.*

Д-во. Применим лемму о замкнутом классе. Пусть $f_0(y_1, \ldots, y_m) \in S,\ f_i(x_1, \ldots, x_n) \in S,\ i = \overline{1, m}$. Рассмотрим функцию

$$f(x_1, \ldots, x_n) = f_0(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)).$$

Получаем:

$$\begin{aligned}\overline{f(\overline{x}_1, \ldots, \overline{x}_n)} &= \overline{\overline{f_0}(f_1(\overline{x}_1, \ldots, \overline{x}_n), \ldots, f_m(\overline{x}_1, \ldots, \overline{x}_n))} = \\ &= \overline{\overline{f_0}(\overline{f_1(x_1, \ldots, x_n)}, \ldots, \overline{f_m(x_1, \ldots, x_n)})} = \\ &= f_0(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)) = \\ &= f(x_1, \ldots, x_n).\end{aligned}$$

Значит, $f \in S$. □

1.3 Функции алгебры логики. Монотонные функции. Замкнутость класса монотонных функций.

Пусть $\alpha, \beta \in E_2^n$. Будем говорить, что $\alpha \leq \beta$, если $\alpha_i \leq \beta_i,\ i = \overline{1, n}$.

Опр. *Функция $f(x_1, \ldots, x_n) \in P_2$ называется монотонной, если для любых наборов $\alpha, \beta \in E_2^n$ из $\alpha \leq \beta$ следует $f(\alpha) \leq f(\beta)$. Множество всех монотонных функций обозначим M.*

Теорема. *Множество M является замкнутым классом.*

Д-во. Применим лемму о замкнутом классе. Пусть $f_0(y_1, \ldots, y_m) \in M,\ f_i(x_1, \ldots, x_n) \in M,\ i = \overline{1, m}$. Рассмотрим функцию

$$f(x_1, \ldots, x_n) = f_0(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))..$$

Пусть $\alpha, \beta \in E_2^n$ и $\alpha \leq \beta$. Тогда:

$$\begin{aligned}f(\alpha) &= f_0(f_1(\alpha), \ldots, f_m(\alpha)) = f_0(\gamma). \\ f(\beta) &= f_0(f_1(\beta), \ldots, f_m(\beta)) = f_0(\delta).\end{aligned}$$

Заметим, что $\gamma \leq \delta \implies f(\alpha) \leq f(\beta) \implies f \in M$. □

1.4 Функции алгебры логики. Линейные функции. Лемма о нелинейной функции.

Опр. *Функция $f(x_1, \ldots, x_n) \in P_2$ называется линейной, если она может быть представлена в виде:*

$$f(x_1, \ldots, x_n) = c_0 \oplus c_1 x_1 \oplus \cdots \oplus c_n x_n,$$

где коэффициенты $c_0, c_1, \ldots, c_n \in E_2$. Множество всех линейных функций обозначим L.

Лемма. *Если $f \notin L$, то, подставляя вместо ее переменных функции $0,\ 1,\ x,\ \overline{x},\ y,\ \overline{y}$ можно получить функцию $x \cdot y$ или функцию $\overline{x \cdot y}$.*

Д-во. Если $f(x_1, \ldots, x_n) \notin L$, то в ее полиноме Жегалкина найдется слагаемое ранга, не меньше двух. Не ограничивая общности, пусть в полиноме Жегалкина функции f содержится слагаемое $x_1 \cdot \cdots \cdot x_k$, где $k \geq 2$. Представим полином Жегалкина функции f в виде:

$$\begin{aligned} f(x_1, \ldots, x_n) &= x_1x_2 \cdot g_1(x_3, \ldots, x_n) \oplus x_1 \cdot g_2(x_3, \ldots, x_n) \oplus \\ &= \oplus x_2 \cdot g_3(x_3, \ldots, x_n) \oplus g_4(x_3, \ldots, x_n), \end{aligned}$$

где $g_1, \ldots, g_4 \in P_2$, причем $g_1 \neq 0$. Значит, найдется такой набор $\alpha \in E_2^{n-2}$, что $g_1(\alpha) = 1$. Пусть $g_2(\alpha) = a,\ g_3(\alpha) = b,\ g_4(\alpha) = c$, где $a, b, c \in E_2$. Тогда

$$f(x_1, x_2, \alpha_1, \ldots, \alpha_{n-2}) = x_1x_2 \oplus ax_1 \oplus bx_2 \oplus c.$$

Положим:

$$\varphi(x, y) = f(x \oplus b, y \oplus a, \alpha_1, \ldots, \alpha_{n-2})..$$

Получаем:

$$\begin{aligned} \varphi(x, y) &= (x \oplus b)(y \oplus a) \oplus a(x \oplus b) \oplus b(y \oplus a) \oplus c = \\ &= (xy \oplus ax \oplus by \oplus ab) \oplus (ax \oplus ab) \oplus (by \oplus ab) \oplus c = \\ &= xy \oplus (ab \oplus c). \end{aligned}$$

□

1.5 Функции алгебры логики. Полнота. Теорема Поста о полноте системы функций алгебры логики.

Теорема (Поста)**.** *Пусть $A \subseteq P_2$. Множество A является полной системой тогда и только тогда, когда A не содержится ни в одном из классов $T_0,\ T_1,\ L,\ S,\ M$.*

Д-во. ($\Longrightarrow$) От противного. Пусть A является полной системой, но содержится в одном из классов $T_0,\ T_1,\ L,\ S,\ M$, пусть, например, $A \subseteq T_0$. Тогда $[A] \subseteq [T_0] = T_0 \neq P_2$. Противоречие. Значит A не содержится ни в одном из классов $T_0,\ T_1,\ L,\ S,\ M$.
($\Longleftarrow$) Пусть A не содержится ни в одном из классов $T_0,\ T_1,\ L,\ S,\ M$. Докажем, что в этом случае A — полная система.
A не содержится в классах $T_0,\ T_1,\ L,\ S,\ M \implies$ в A найдутся такие функции f_0, f_1, f_l, f_s, f_m, что $f_0 \notin T_0,\ f_1 \notin T_1,\ f_l \notin L,\ f_s \notin S,\ f_m \notin M$. Отметим что функции f_0, f_1, f_l, f_s, f_m не обязательно различны. Покажем, что функциями над A можно выразить все функции из полной системы $\{0,\ 1,\ \overline{x},\ x,\ x \cdot y\}$.
1. Построение констант 0 и 1.
Рассмотрим функции f_0 и f_1. Положим $\varphi_0(x) = f_0(x, \ldots, x),\ \varphi_1(x) = f_1(x, \ldots, x)$. Тогда:

x	φ_0	φ_1
0	1	b
1	a	0

Теперь, если $a = 1$ и $b = 0$, то $\varphi_0(x) = 1,\ \varphi_1(x) = 0$.
Если же $a = 0$ или $b = 1$, то получена функция $\overline{x}$. Тогда по лемме о несамодвойственной функции из $f_s \in S$, подставляя вместо ее переменных функции $x,\ \overline{x}$, получаем некоторую константу $c \in E_2$, а затем $\overline{c} \in E_2$.
2. Построение отрицания $\overline{x}$.
По лемме о немонотонной функции из f_m, подставляя вместо ее переменных функции $0,\ 1,\ x$, получаем отрицание $\overline{x}$.
3. Построение конъюнкции.
По лемме о нелинейной функции из f_l подставляя вместо ее переменных функции $0,\ 1,\ x,\ \overline{x},\ y,\ \overline{y}$ и, возможно навешивая отрицание над функцией, получаем конъюнкцию $x \cdot y$. □

1.6 Функции алгебры логики. Предполные классы. Теорема о предполных классах.

Опр. *Пусть $A \subseteq P_2$. Множество A называется предполным классом, если*

1. $[A] \neq P_2$;
2. $\forall f \in P_2 \setminus A : [A \cup \{f\}] = P_2$.

Утверждение. *Любой предполный класс является замкнутым классом.*

Д-во. От противного. Пусть $A \subseteq P_2$ — предполный класс, но $[A] \neq A$. Тогда $\exists f \in [A] \setminus A$. Получаем: $[A] = [A \cup \{f\}] = P_2$. Противоречие. Значит $[A] = A$. □

Теорема. *В P_2 найдется всего пять предполных классов: $T_0,\ T_1,\ L,\ S,\ M$.*

Д-во. Сначала покажем, что каждый из классов $T_0,\ T_1,\ L,\ S,\ M$ не содержится ни в каком из этих классов. Для этого построим таблицу, в которой строки и столбцы соответствуют этим классам, а на пересечении строки и столбца указана функция, принадлежащая классу, которым обозначена эта строка, и не принадлежащая классу, которым обозначен этот столбец:

	T_0	T_1	L	S	M
T_0	–	0	$x \cdot y$	0	$x \oplus y$
T_1	1	–	$x \cdot y$	1	$x \sim y$
L	$\overline{x}$	$\overline{x}$	–	0	$\overline{x}$
S	$\overline{x}$	$\overline{x}$	$m(x,y,z)$	–	$\overline{x}$
M	1	0	$x \cdot y$	0	–

где $m(x, y, z) = xy \oplus xz \oplus yz$.
Теперь докажем, что каждый из классов $T_0,\ T_1,\ L,\ S,\ M$ является предполным. Например, рассмотрим класс T_0. Тогда:

1. $[T_0] = T_0 \neq P_2$;
2. если $f \notin T_0$, то по теореме Поста: $[T_0 \cup \{f\}] = P_2$.

Значит T_0 — предполный класс. Аналогично проводятся рассуждения для остальных классов. Наконец, докажем от противного, что других предполных классов нет. Пусть $A \subseteq P_2$ — пердполный класс, причем $A \neq T_0,\ T_1,\ L,\ S,\ M$. Значит либо A не содержится ни в одном из этих классов, либо строго содержится в каком-то из них.
Если A не содержится ни в одном из этих классов, то по теореме Поста $[A] = P_2$. Получаем противоречие с п.1 определения предполного класса.
Пусть A строго содержится в каком-то из этих классов, например, пусть $A \subseteq T_0,\ A \neq T_0$. Тогда $\exists f \in T_0 \setminus A$, откуда $[A \cup \{f\}] \subseteq T_0 \neq P_2$. Получаем противоречие с п.2 определения предполного класса.
Значит других предполных классов нет. □

1.7 Деревья. Теорема о равносильных определениях дерева.

Опр. *Деревом называется связный граф без циклов.*

Теорема. *Пусть $G = (V, E)$ — граф с p вершинами и q ребрами. Тогда следующие утверждения равносильны:*

1. *G — дерево;*
2. *G — связный граф и $q = p - 1$;*
3. *G — граф без циклов и $q = p - 1$;*
4. *G — граф без циклов, но при соединении любой пары несмежных вершин ребром появляется цикл;*
5. *G — связный граф, но при удалении любого ребра остается несвязный граф.*

Д-во. (1 $\implies$ 2) G — без циклов, поэтому по соотношению для G между числом вершин p, числом ребер q и числом компонент связности $s = 1$ получаем: $1 = s = p - q$. Значит, $q = p - 1$.
(2 $\implies$ 3) Если в связном графе G найдется цикл, то удалим из G некоторое ребро e из цикла. Останется связный граф G'. По соотношению для G' между числом вершин p, числом ребер $q - 1$ и числом компонент связности $s' = 1$ получаем: $s' \geq p - (q - 1) = (p - q) + 1 = 2$ — противоречие. Значит, G без циклов.
(3 $\implies$ 4) По соотношению для G между числом вершин p, числом ребер q и числом компонент связности s получаем: $s = p - q = 1$, т. е. G связный. Значит, при соединении в G любой пары несмежных вершин ребром появится цикл.
(4 $\implies$ 5) Если G не связный, то при соединении двух вершин из разных компонент связности цикл не появится. Значит, G связный. Пусть при удалении из G некоторого ребра e остался связный граф G'. Тогда G получается из связного графа G' добавлением нового ребра e. Поэтому в G найдется цикл — противоречие. Значит, при удалении из G любого ребра останется несвязный граф.
(5 $\implies$ 1) Если в G найдется цикл, то удалим из G любое ребро из цикла. Останется связный граф — противоречие. Значит, G без циклов. □

1.8 Остовные деревья. Алгоритм построения кратчайшего остовного дерева в связном графе (с обоснованием).

Опр. *Остовным деревом графа называется его остовный подграф, являющийся деревом.*

Значит, D — остовное дерево графа G, если выполняются три свойства:
1) D — остовный подграф, т.е. содержит все вершины графа G;
2) D — связный граф;
3) D — граф без циклов.

Опр. *Граф $G = (V, E)$ называется взвешенным, если задана функция весов $w : E \to \mathbb{R}_+$, которая ставит в соответствие каждому ребру $e \in E$ неотрицательное действительное число $w(e)$, называемое весом этого ребра e.*

Опр. *Пусть $G = (V, E)$ — взвешенный связный граф с функцией весов w и $D = (V, E')$ — его остовное дерево, $E' \subseteq E$. Тогда весом $w(D)$ дерева D называется сумма весов всех его ребер, т.е. $w(D) = \sum\limits_{e \in E'} w(e)$.*

Опр. *Остовное дерево D^* связного графа G называется кратчайшим, если его вес $w(D^*)$ является наименьшим среди весов всех остовных деревьев графа G.*

Алгоритм построения кратчайшего остовного дерева.
1. Положить $H_1 = (V_1, E_1)$, где $V_1 = \{v\},\ v \in V$ — произвольная вершина, $E_1 = \emptyset$.
2. Для всех $i = 1, \ldots, p-1$ повторить: выбрать произвольное ребро $e_i \in E$ наименьшего веса в множестве $\{e = (v, w) \in E : v \in V_i, w \in V \setminus V_i\}$, и положить $H_{i+1} = (V_i \cup \{w_i\}, E_i \cup \{e_i\})$.
3. Положить $D^* = H_p$.

Теорема. *Предложенный выше алгоритм для заданного графа и для заданной функции весов w находит какое-то кратчайшее дерево D^* графа G.*

Д-во. Пусть алгоритм строит граф $D^* = (V, E^*),\ E^* \subseteq E$.
1. Сначала покажем, что граф D^* — остовное дерево графа G. Действительно, начинаем с дерева H_1, состоящего из одной вершины. На каждом шаге добавляем одну вершину и одно ребро, получая дерево $H_{i+1},\ i = 1, \ldots, p-1$. Дерево $H_p = D^*$ содержит p вершин, поэтому является остовным.
2. Теперь покажем, что граф D^* — кратчайшее остовное дерево графа G. Пусть $D' = (V, E'),\ E' \subseteq E$ — какое-то кратчайшее остовное дерево графа G. Пусть при построении дерева D^* ребра добавлялись в следующем порядке: $e_1, e_2, \ldots, e_{p-1}$.
Пусть k — такое число, $1 \le k \le p$, что ребра $e_1, \ldots, e_{k-1}$ принадлежит дереву D', а ребро e_k не принадлежит дереву D'. Если $k = p$, то $D^* = D'$ и все доказано.
Пусть $k < p$. Рассмотрим дерево $H_k = (V_k, E_k),\ E_k = \{e_1, \ldots, e_{k-1}\} \subseteq E$, полученное при применении алгоритма. Пусть $e_k = (v_k, w_k)$, где $v_k \in V_k,\ w_k \in V \setminus V_k$. Заметим, что H_k является поддеревом дерева D'. В дереве D' найдется простая (v_k, w_k)-цепь P. Пусть $e' = (v', w') \in E'$ — первое такое ребро этой цепи при движении от вершины v_k к вершине w_k, что $v' \in V_k,\ w' \in V \setminus V_k$. При этом $e' \ne e_k$. Отметим, что $w(e_k) \le w(e')$, т.к. иначе при применении алгоритма к дереву H_k было бы добавлено ребро e', а не ребро e_k.
Рассмотрим подграф $G' = D' + e_k$. Граф G' содержит ровно один цикл $C = v_k P_1 v' e' w' P_2 w_k e_k v_k$,

где P_1 и P_2 — части, на которых ребро e' разбивает цепь P, причем $v_k P_1 v'$ — проста цепь в дереве H_k. Значит, подграф $H = G' - e'$ является связным и не содержит циклов.
Итак, H — остовный связный подграф без циклов. Значит, H — остовное дерево графа G. Кроме того, $w(H) = w(D') - w(e') + w(e_k) \leq w(D')$. Но D' — кратчайшее дерево, поэтому $w(H) = w(D')$.
Таким образом, построили кратчайшее остовное дерево H, у которого с остовным деревом D^* совпадает уже не менее k первых добавляемых по алгоритму ребер. Повторим рассуждения, положив $D' = H$. Через конечное число повторов получим, что D^* — кратчайшее остовное дерево графа G. □

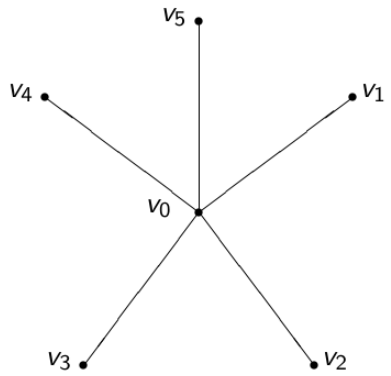
1.9 Раскраски вершин графов. Теорема о раскраске вершин планарных графов в 5 цветов.

Опр. *Раскраска вершин графа $G = (V, E)$ в k цветов — отображение $\rho : V \to \{1, 2, \ldots, k\}$, в котором из $(v, w) \in E$ следует $\rho(v) \neq \rho(w)$. Т.е. любые смежные вершины обязаны получить разные цвета.*

Опр. *Хроматическое число $\chi(G)$ графа G — наименьшее число цветов, в которое можно раскрасить его вершины. Для любого графа $G = (V, E)$ верно соотношение $\chi(G) \leq |V|$.*

Теорема. *Вершины любого планарного графа G можно раскрасить не более чем в пять цветов.*

Д-во. Проведем индукцию по числу p вершин в графе G.
Базис индукции. При $p = 1$ утверждение верно.
Индуктивный переход. Пусть вершины любого планарного графа менее чем с p вершинами можно раскрасить в 5 цветов. Рассмотрим планарный граф $G = (V, E)$, где $|V| = p$. Пусть задана его укладка на плоскости $\Phi(G)$. По доказанному свойству в графе G найдется вершина $v_0 \in V$, что $d_G(v_0) \leq 5$.
Пусть $v_1, \ldots, v_m \in V$ — все смежные с v_0 вершины в графе G, $m \leq 5$, и пусть в укладке $\Phi(G)$ ребра $(v_0, v_1), \ldots, (v_0, v_m)$ расположены по часовой стрелке в этом порядке.



Рассмотрим планарный граф $G' = G - v_0$. Для него верно предположение индукции, поэтому найдется раскраска ρ его вершин в 5 цветов. Перенесем эту раскраску ρ на вершины графа G, при этом вершина v_0 останется неокрашенной. Покажем, что вершину v_0 можно покрасить,

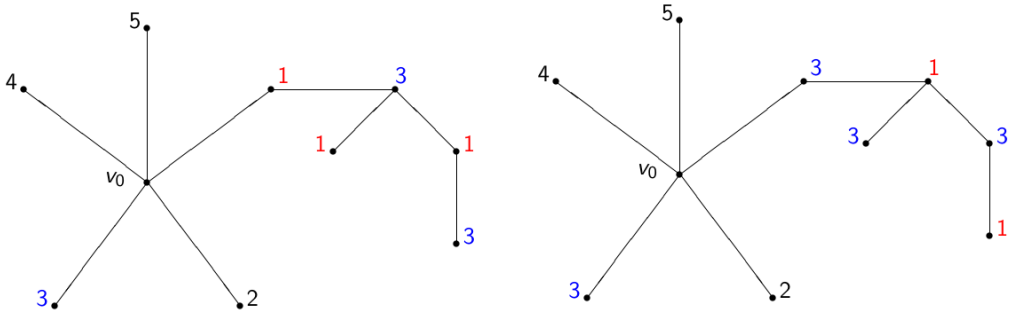
не добавляя новый.

1. Если $m \leq 4$ или $m = 5$, но среди цветов вершин $v_1, \ldots, v_m$ не встречается какой-то цвет, то припишем вершине v_0 цвет, отсутствующий в вершинах $v_1, \ldots, v_m$ цвет.

2. Пусть теперь $m = 5$ и среди цветов вершин $v_1, \ldots, v_m$ встречаются все 5 цветов, причем вершина v_i окрашена в цвет i, $i = \overline{1,5}$.

Пусть $A_{1,3}(v_1)$ — множество всех тех вершин графа G, в которые найдутся пути из вершины v_1 по вершинам только цветов 1 и 3.

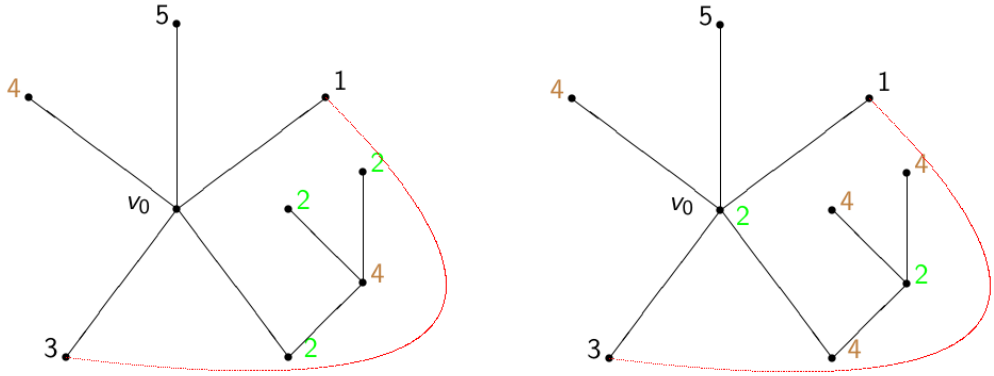
2.1 Если $v_3 \notin A_{1,3}(v_1)$, то все вершины из $A_{1,3}$ перекрасим: если вершина окрашена в цвет 1, то ее покрасим в цвет 3; если вершина окрашена в цвет 3, то ее покрасим в цвет 1.



Тогда вершина v_1 приобретет цвет 3. А значит, вершине v_0 можно приписать цвет 1.

2.2 Пусть $v_3 \in A_{1,3}$. Это означает, что в графе G найдется цикл C, содержащий вершину v_0, и все другие вершины цикла C окрашены только в цвета 1 или 3, причем вершины v_2 и v_4 лежат по разные стороны от этого цикла.

Пусть $A_{2,4}(v_2)$ — множество всех тех вершин графа G, в которые найдутся пути из вершины v_2 по вершинам только цветов 2 и 4. Теперь $v_4 \notin A_{2,4}(v_2)$, и все вершины из $A_{2,4}(v2)$ перекрасим: если вершина окрашена в цвет 2, то ее покрасим в цвет 4; если вершина окрашена в цвет 4, то ее покрасим в цвет 2. Тогда вершина v_2 приобретет цвет 4. А значит, вершине v_0 можно приписать цвет 2.



□

1.10 Алфавитные коды. Однозначность (разделимость) алфавитного кода. Алгоритм Маркова распознавания однозначности алфавитного кода (с обоснованием).

Опр. *Пусть заданы два алфавита A и B. Алфавит A назовем исходным, алфавит B — кодирующим. Кодированием (из A в B) называется произвольное отображение $\varphi : A^* \to B^*$. При кодировании φ любое слово $\alpha \in A^*$ называется сообщением, а слово $\beta = \varphi(\alpha) \in B^*$ — его кодом.*

Опр. *Кодирование $\varphi : A^* \to B^*$ называется однозначным (или разделимым), если для любых слов $\alpha_1, \alpha_2 \in A^*$ из $\alpha_1 \neq \alpha_2$ следует $\varphi(\alpha_1) \neq \varphi(\alpha_2)$.*

Опр. *Если $\varphi : A^* \to B^*$ — кодирование, то множество кодов всех слов из A^* назовем кодом C_φ, т.е. $C_\varphi = \{\varphi(\alpha) : \alpha \in A^*\} \subseteq B^*$.*

Опр. *Пусть $A = \{a_1, \ldots, a_r\}$ — исходный алфавит, $B = \{b_1, \ldots, b_q\}$ — кодирующий алфавит. Кодирование $\varphi : A^* \to B^*$ называется алфавитным, если оно описывается следующей схемой:*

1. *заданы различные непустые коды букв алфавита A:*

$$\varphi(a_1) = B_1 \in B^*,$$
$$\ldots$$
$$\varphi(a_r) = B_r \in B^*.$$

2. *слова в алфавите A кодируются побуквенно, т.е. если $\alpha \in A$, $\alpha = a_{i_1} a_{i_2} \ldots a_{i_m}$, где $m \geq 2$, то*

$$\varphi(\alpha) = \varphi(a_{i_1})\varphi(a_{i_2}) \ldots \varphi(a_{i_m}) = B_{i_1} B_{i_2} \ldots B_{i_m}.$$

Алфавитный код C_φ назовем однозначны (или разделимым), если кодирование φ — разделимо.

Пусть $C_\varphi = \{B_1, \ldots, B_r\} \subseteq B^*$ — алфавитный код. Построим орграф $G_\varphi = (V_\varphi, E_\varphi)$ для кода C_φ.

1. Множество вершин $V_\varphi \subseteq B^*$ состоит из пустого слова Λ и всех тех слов в алфавите B, котопый являются собственным префиксом некоторого кодоваого слова и одновременно собственным суффиксом некоторого кодового слова и не являются никаким кодовым словом, т.е.

$$V_\varphi = \{\lambda\} \cup \{\beta \in B^* : 1)\ \exists B_i \in C_\varphi : B_i = \beta\beta',\ \beta' \neq \Lambda;$$
$$2)\ \exists B_j \in C_\varphi : B_j = \beta''\beta,\ \beta'' \neq \Lambda;$$
$$3)\ \beta \neq B_k,\ k = \overline{1, r}\ \}.$$

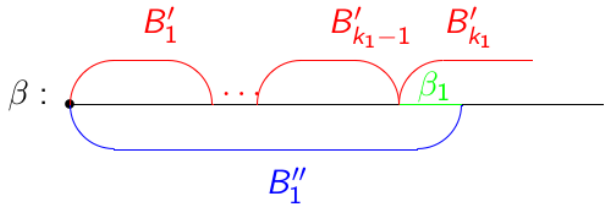
2. Опишем множество дуг E_φ: если $\beta', \beta'' \in V_\varphi$, то $(\beta', \beta'') \in E_\varphi$, если найдется такое кодовое слово B_i и такая последовательность D кодовых слов $B_{i_1}, \ldots, B_{i_k}$, что

$$B_i = \beta' B_{i_1} \ldots B_{i_k} \beta'',$$

причем если $\beta' = \beta'' = \Lambda$, то $k \geq 2$; если $\beta' \neq \Lambda$ или $\beta'' \neq \Lambda$, то $k \geq 1$; если $\beta'\beta'' \neq \Lambda$, то $k \geq 0$. При этом дуге $(\beta', \beta'') \in E_\varphi$ приписываем пометку $D = B_{i_1} \ldots B_{i_k}$.

Теорема. *Алфавитный код C_φ является разделимым тогда и только тогда, когда в графе G_φ отсутствуют ориентированные циклы (в том числе, и петли), проходящие через вершину Λ.*

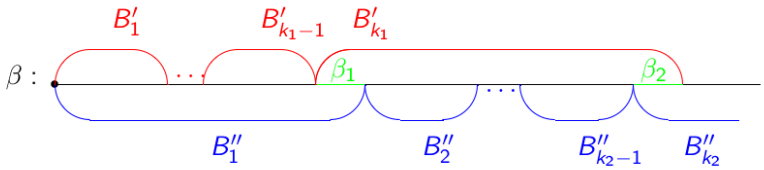
Д-во. Пусть $C_\varphi = \{B_1, \dots, B_r\} \subseteq B^*$ — алфавитный код и G_φ — граф для кода C_φ.
1. Пусть код C_φ не является разделимым. Значит, найдется слово $\beta \in B^*$ наименьшей длины, которое допускает не менее двух декодирований. Пусть $\beta = B'_1 B'_2 \dots B'_{t_1}$ — разбиение кода β на кодовые слова в 1-м декодировании и $\beta = B''_1 B''_2 \dots B''_{t_2}$ — разбиение слова β на кодовые слова во 2-м декодировании.
Обозначим: $l'_i = |B'_i|,\ i = \overline{1, t_1}$ и $l''_i = |B''_i|, i = \overline{1, t_2}$. Пусть, для определенности, $l''_1 > l'_1$.



Найдем такое число k_1, что

$$\sum_{i=1}^{k_1-1} l'_i < l''_1, \quad \sum_{i=1}^{k_1} l'_i > l''_1.$$

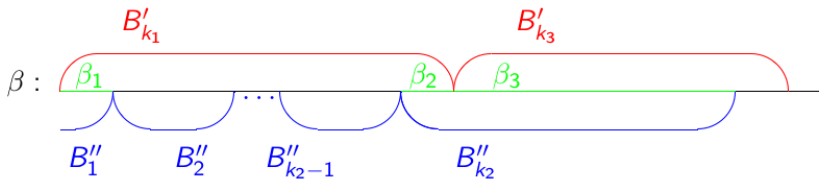
Заметим, что равенства здесь быть не может, т.к. в этом случае слово β можно было бы уменьшить, что не так. Тогда $B''_1 = B'_1 \dots B'_{k_1-1}\beta_1$ для некоторого слова $\beta_1 \in B^*,\ \beta_1 \neq \Lambda$. Отметим, что слово β_1 является собственным префиксом кодового слова B'_{k_1} и собственным суффиксом кодового слова B''_1, а также не является никаким кодовым словом.
Значит, в графе G_φ присутствует дуга $e_1 = (\Lambda, \beta_1) \in E_\varphi$, которой приписана пометка $D_1 = B'_1 \dots B'_{k_1-1}$.



Теперь найдем такое число k_2, что

$$|\beta_1| + \sum_{i=2}^{k_2-1} l''_i < l'_{k_1}, \quad |\beta_1| + \sum_{i=1}^{k_2} l''_i > l'_{k_1}.$$

Снова равенства быть не может, т.к. в этом случае слово β можно было бы уменьшить, что не так. Тогда $B'_{k_1} = \beta_1 B''_2 \dots B_{k_2-1}\beta_2$ для некоторого слова $\beta_2 \in B^*,\ \beta_2 \neq \lambda$. Слово β_2 является собственным префиксом кодового слова B''_{k_2} и собственным суффиксом кодового слова B'_{k_1}, а также не является никаким кодовым словом.
Значит, в графе G_φ присутствует дуга $e_2 = (\beta_1, \beta_2) \in E_\varphi$, которой приписана пометка $D_2 = B''_2 \dots B''_{k_2-1}$.



Далее найдем такое число k_3, что

$$|\beta_2| + \sum_{i=k_1+1}^{k_3-1} l'_i < l''_{k_2}, \quad |\beta_2| + \sum_{i=k_1+1}^{k_3} l'_i > l''_{k_2}.$$

Равенства быть не может, т.к. в этом случае слово β можно было бы уменьшить, что не так. Тогда $B''_{k_3} = \beta_2 B'_{k_1+1} \dots B'_{k_3-1}\beta_3$ для некоторого слова $\beta_3 \in B^*,\ \beta_3 \neq \Lambda$.
Значит, в графе G_φ присутствует дуга $e_3 = (\beta_2, \beta_3) \in E_\varphi$, которой приписана пометка $D_3 = B'_{k_1+1} \dots B'_{k_3-1}$. И т.д.
Через конечное число число таких шагов достигнем окончания слова β. Значит в графе G_φ присутствует дуга $e_{m+1} = (\beta_m, \Lambda) \in E_\varphi$ для некоторого слова $\beta_m \in B^*,\ \beta_m \neq \Lambda$. Этой дуге приписана пометка $B_{m+1} = B^o_{k_{m-1}+1} \dots B^o_{k_m-1}$, где $^o = \{',''\}$ в зависимости от четности числа m.
Таким образом, в графе G_φ найдется ориентированный замкнутый путь:

$$P = \Lambda, e_1, \beta_1, e_2, \beta_2, \dots, \beta_m, e_{m+1}, \Lambda,$$

в котором вершина Λ не встречается среди вершин $\beta_1, \dots, \beta_m$.
Из этого пути P можно выделить ориентированный цикл (в частности, петлю), проходящий через вершину Λ.

2. пусть теперь в графе G_φ найдется ориентированные цикл (в частности, петля)

$$P = \Lambda, e_1, \beta_1, e_2, \beta_2, \dots, \beta_m, e_{m+1}, \Lambda,$$

проходящий через вершину Λ. Пусть дуге e_i приписана пометка $D_i = B_{i_1}, \dots, B_{i_{k_i}}$, $i = \overline{1, m+1}$. Покажем, что слово

$$\beta = D_1\beta_1 D_2\beta_2 \dots \beta_m D_{m+1} \in B^*$$

допускает не менее двух декодирований. Пусть, для определенности, m — четное.
Первое декодирование:

$$D_1\beta_1 D_2\beta_2 D_3\beta_3 D_4 \dots D_m\beta_m D_{m+1}.$$

Второе декодирование:

$$D_1\beta_1 D_2\beta_2 D_3\beta_3 D_4\beta_4 \dots \beta_{m-1} D_m\beta_m D_{m+1}.$$

Случай нечетного m разбирается аналогично. Значит, код C_φ не является разделимым. $\square$

1.11 Алфавитные коды. Теорема Маркова об алфавитных кодах.

Теорема. *Пусть A — исходный алфавит, $C_\varphi = \{B_1, \dots, B_r\} \subseteq B^*$ — алфавитный код, где $|B_i| = l_i,\ i = \overline{1,r}$. Пусть $L = \sum\limits_{i=1}^{r} l_i$ и w обозначает наибольшее число кодовых слов (возможно, с повторами), соединение которых является подслово какого-то кодового слова. Тогда если код C_φ не является разделимым, то найдутся такие слова $\alpha_1, \alpha_2 \in A^*,\ \alpha_1 \neq \alpha_2,\ \varphi(\alpha_1) \neq \varphi(\alpha_2)$, что*

$$|\alpha_1|, |\alpha_2| \leq \left\lfloor \frac{(L - r + 2)(w + 1)}{2} \right\rfloor,$$

где $\lfloor a \rfloor$ обозначает целую часть числа a.

Д-во. Код C_φ не является разделимым, значит, в графе G_φ найдется ориентированный цикл (в частности, петля)

$$P = \Lambda, e_1, \beta_1, e_2, \beta_3, \dots, \beta_m, e_{m+1}, \Lambda,$$

проходящий через вершину Λ. Пусть дуге e_i приписана пометка $D_i = B_{i_1} \dots B_{i_{k_i}},\ i = \overline{1, m+1}$. Можно считать, что P — петля или простой цикл, поэтому слова $\beta_1, \dots, \beta_m$ — различны.
Каждое слово β_i является, в частности, собственным префиксом какого-то кодового слова, поэтому

$$m \leq \sum_{i=1}^{r} (l_i - 1) = L - r.$$

Рассмотрим слово

$$\beta = D_1 \beta_1 D_2 \beta_2 \dots \beta_m D_{m+1} \in B^*,$$

которое допускает не менее двух декодирований. Пусть $\alpha_1, \alpha_2 \in A^*,\ \alpha_1 \neq \alpha_2$ — два декодирования слова β, т.е. $\beta = \varphi(\alpha_1) = \varphi(\alpha_2)$. Слова $\beta_1, \dots, \beta_m$ разбивают слово β на $m + 1$ частей: $D_1, \dots, D_{m+1}$. Рассмотрим k пар частей:

$$(D_1, D_2),\ (D_3, D_4), \dots,\ (D_{2k-1}, D_\kappa),$$

где $k = \left\lfloor \frac{m+1}{2} \right\rfloor$. Для каждого $i = 1, \dots, k$ слова

$$\beta_i' = \overbrace{\beta_{2i-2} D_{2i-1} \beta_{2i-1}}^{1} \overbrace{D_{2i}}^{\leq w},$$
$$\beta_i'' = \underbrace{D_{2i-1}}_{\leq w} \underbrace{\beta_{2i-1} D_{2i} \beta_{2i}}_{1},$$

разбиваются не более, чем на $w + 1$, кодовых слов. Значит, каждая пара (D_{2i-1}, D_{2i}) вносит не более чем $w + 1$ кодовых слов в каждое из декодирований слова β.
Если $m + 1$ — нечетно, то останется еще последовательность D_{m+1}, которая также вносит не более $w + 1$ кодовых слов в каждое из декодирований слова β. Значит

$$|\alpha_1|, |\alpha_2| \leq \frac{m+2}{2}(w+1) \leq \frac{(L - r + 2)(w + 1)}{2}.$$

Из того, что $|\alpha_1|, |\alpha_2|$ — целые числа, получаем утверждение теоремы. □

1.12 Алфавитные коды. Неравенство Макмиллана.

Теорема (Неравенство Макимллана). *Пусть $C_\varphi = \{B_1, \dots, B_r\}$ — алфавитный код в кодирующем алфавите B, $|B| = q$, и $|B_i| = l_i$, $i = \overline{1,r}$. Если код C_φ — разделим, то верно неравенство:*

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

Д-во. Пусть $n \ge 1$. Рассмотрим выражение:

$$\left(\sum_{i=1}^{r} \frac{1}{q^{l_i}} \right)^n.$$

Получаем:

$$\left(\sum_{i=1}^{r} \frac{1}{q^{l_i}} \right)^n = \left(\sum_{i_1=1}^{r} \frac{1}{q^{l_{i_1}}} \right) \cdot \left(\sum_{i_2=1}^{r} \frac{1}{q^{l_{i_2}}} \right) \cdot \dots \cdot \left(\sum_{i_n=1}^{r} \frac{1}{q^{l_{i_n}}} \right) =$$

$$= \sum_{i_1=1}^{r} \sum_{i_2=1}^{r} \cdots \sum_{i_n=1}^{r} \frac{1}{q^{l_{i_1}+l_{i_2}+\dots+l_{i_n}}} = \sum_{k=1}^{n \cdot l_{\max}} \frac{c_k}{q^k},$$

где $l_{\max} = \max\limits_{1 \le i \le r} l_i$ и c_k равно числу таких наборов $(i_1, \dots, i_n)$, что $l_{i_1} + \dots + l_{i_k} = k$.

Лемма. *Если C_φ — разделимый алфавитный код, то $c_k \le q^k$.*

Д-во. Каждому набору $(i_1, \dots, i_n)$ соответствует слово $\alpha = a_{i_1} \dots a_{i_n} \in A^*$. Далее:

$$\varphi(\alpha) = \varphi(a_{i_1} \dots a_{i_n}) = B_{i_1} \dots B_{i_n} = \beta,$$

причем $|\beta| = l_{i_1} + \dots + l_{i_n} = k$. Но код C_φ — разделим, поэтому если $\beta \in B^*$, то найдется не более одного такого слова $\alpha \in A^*$, что $\varphi(\alpha) = \beta$, Поэтому любому слову $\beta \in B^*$, $|\beta| = k$, соответствует не более одного такого слова $\alpha = a_{i_1} \dots a_{i_n} \in A^*$, что $\beta = \varphi(\alpha)$. А значит, число таких наборов $(i_1, \dots, i_n)$, что $l_{i_1} + \dots + l_{i_n} = k$, не превосходит числа слов длины k в алфавите B, т.е. $c_k \le q^k$. □

Итак,

$$\left(\sum_{i=1}^{r} \frac{1}{q^{l_i}} \right)^n = \sum_{k=1}^{n \cdot l_{\max}} \frac{c_k}{q_k} \le \sum_{k=1}^{n \cdot l_{\max}} 1 \le n \cdot l_{\max},$$

или

$$\sum_{i=1}^{r} \frac{1}{q_{l_i}} \le \sqrt[n]{n \cdot l_{\max}}.$$

Данное неравенство выполняется для любого $n \ge 1$. Переходя к пределу при $n \to \infty$, получаем:

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

□

1.13 Алфавитные коды. Префиксные коды. Существование префиксного кода с заданными длинами кодовых слов.

Опр. *Алфавитный код $C = \{B_1, \dots, B_r\} \subseteq B^*$ называется префиксным, если никакое его кодовое слово не является префиксом никакого другого его кодового слова.*

Утверждение. *Любой префиксный алфавитный код разделим.*

Д-во. Пусть $C = \{B_1, \dots, B_r\} \subseteq B^*$ — префиксный алфавитный код. Рассмотрим произвольное слово $\beta \in B^*$.
Пусть $\beta = B_{i_1}\beta_1$ для некоторого кодового слова $B_{i_1} \in C$ и некоторого слова $\beta_1 \in B^*$. Отметим, что кодовое слово B_{i_1} (если оно существует) находится однозначно. Далее повторим рассуждения слова $\beta_1 \in B^*$. В итоге либо однозначно декодируем слово β, либо на каком-то шаге не найдем подходящее кодовое слово, а значит, слово β не допускает декодирования. $\square$

Теорема. *Пусть $q, l_1, \dots, l_r$ — такие натуральные числа, что выполняется неравенство:*

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

Тогда существует такое префиксный код $C = \{B_1, \dots, B_r\}$ в любом кодирующем алфавите из q букв, что $|B_i| = l_i,\ i = \overline{1, r}$.

Д-во. Пусть $m_1, \dots, m_k$ — все различные числа среди чисел $l_1, \dots, l_r$, причем чисел m_i среди $l_1, \dots, l_r$ ровно r_i. Отметим, что $r_1 + \dots + r_k = r$. Значит

$$\sum_{j=1}^{k} \frac{r_j}{q^{m_j}} \le 1.$$

Пусть для определенности $m_1 < m_2 < \dots < m_k$. Тогда выполняется система неравенств:

$$\begin{cases} \frac{r_1}{q^{m_1}} \le 1, \\ \frac{r_1}{q^{m_1}} + \frac{r_2}{q^{m_2}} \le 1, \\ \frac{r_1}{q^{m_1}} + \frac{r_2}{q^{m_2}} + \frac{r_3}{q^{m_3}} \le 1, \\ \dots \\ \frac{r_1}{q^{m_1}} + \frac{r_2}{q^{m_2}} + \dots + \frac{r_k}{q^{m_k}} \le 1, \end{cases}$$

откуда

$$\begin{cases} r_1 \le q^{m_1}, \\ r_2 \le q^{m_2} - r_1 q^{m_2 - m_1}, \\ r_3 \le q^{m_3} - r_2 q^{m_3 - m_2} - r_1 q^{m_3 - m_1}, \\ \dots \\ r_k \le q^{m_k} - r_{k-1} q^{m_k - m_{k-1}} - \dots - r_1 q^{m_k - m_1}. \end{cases}$$

Итак, $r_1 \le q^{m_1}$. Выберем r_1 различных слов $B_1, \dots, B_{r_1}$ длины m_1 в алфавите B. Всего различных слов длины m_1 в алфавите B найдется q^{m_1}. Из $r_1 \le q^{m_1}$ следует, что r_1 различных

слов длины m_1 в алфавите B можно найти. Из дальнейшего рассмотрения исключим все слова в алфавите B с префиксами $B_1, \ldots, B_{r_1}$.
Теперь, $r_2 \leq q^{m_2} - r_1 q^{m_2 - m_1}$. Выберем r_2 различных слов $B_{r_1+1}, \ldots, B_{r_1+r_2}$ длины m_2 в алфавите B, не начинающихся с $B_1, \ldots, B_{r_1}$. Всего различных слов длины m_2 в алфавите B найдется q^{m_2}. Из них содержат одно из слов $B_1, \ldots, B_{r_1}$ как префикс в точности $r_1 q^{m_2 - m_1}$ слов. Но $r_2 \leq q^{m_2} - r_1 q^{m_2 - m_1}$, поэтому r_2 различных слов длины m_2 с таким условием можно найти. Из дальнейшего рассмотрения исключим все слова в алфавите B с префиксами $B_1, \ldots, B_{r_1}, B_{r_1+1}, \ldots, B_{r_1+r_2}$.
Повторив эти рассуждения k раз, получим слова:

$$B_1, \ldots, B_{r_1}, B_{r_1+1}, \ldots, B_{r_1+r_2}, \ldots, B_{r_1+r_2+\cdots+r_k}.$$

По построению ни одно из этих слов не является префиксом никакого другого их этих слов. Поэтому эти слова образуют искомый префиксный (а значит, и разделимый) алфавитный код. □

1.14 Коды с минимальной избыточностью (оптимальные коды). Теорема редукции.

Опр. *Пусть $A = \{a_1, \ldots, a_r\}$ — исходный алфавит и B — кодирующий алфавит. Пусть $P = (p_1, \ldots, p_r)$ — набор частот появления букв исходного алфавита, где*

1. $p_i \in \mathbb{R}_+$;
2. $\sum\limits_{i=1}^{r} = 1$.

Пусть $C_\varphi = \{B_1, \ldots, B_r\}$ — алфавитный код, $|B_i| = l_i$. Стоимостью (или избыточностью) кода C_φ назовем величину

$$c(\varphi) = \sum_{i=1}^{r} p_i l_i.$$

Опр. *Однозначный код C_{φ^*} назовем оптимальным (или кодом с минимальной избыточностью) (при заданных A, B, B), если*

$$c(\varphi^*) = \inf_{\varphi} c(\varphi),$$

где инфинум берется по всем однозначным алфавитным кодам.

Лемма. *Пусть $B = \{0, 1\}$ — кодирующий алфавит, заданы два исходных алфавита A, A' и соответствующие наборы частот P, P' и алфавитные коды C_φ, $C_{\varphi'}$:*

$$\begin{aligned} A &= \{a_1, \ldots, a_{r-1}, a_r\}, & A' &= \{a_1, \ldots, a_{r-1}, a', a''\}, \\ P &= (p_1, \ldots, p_{r-1}, p_r), & P' &= (p_1, \ldots, p_{r-1}, p', p''), \\ C_\varphi &= \{B_1, \ldots, B_{r-1}, B_r\}, & C_{\varphi'} &= \{B_1, \ldots, B_{r-1}, B_r 0, B_r 1\}, \end{aligned}$$

где $r \geq 2$. Тогда если один из этих кодом префиксный, то и другой префиксный, причем $c(\varphi') = c(\varphi) + p_r$.

Теорема. *Пусть $B = \{0, 1\}$ — кодирующий алфавит, заданы два исходных алфавита A, A' и соответствующие наборы частот P, P' и алфавитные коды C_φ, $C_{\varphi'}$:*

$$A = \{a_1, \ldots, a_{r-1}, a_r\}, \qquad A' = \{a_1, \ldots, a_{r-1}, a', a''\},$$
$$P = (p_1, \ldots, p_{r-1}, p_r), \qquad P' = (p_1, \ldots, p_{r-1}, p', p''),$$
$$C_\varphi = \{B_1, \ldots, B_{r-1}, B_r\}, \qquad C_{\varphi'} = \{B_1, \ldots, B_{r-1}, B_r0, B_r1\},$$

где $r \geq 2$. Тогда:

1. *если $C_{\varphi'}$ — оптимальный префиксный код, то и C_φ — оптимальный префиксный код;*
2. *Если C_φ — оптимальный префиксный код и*

$$p_1 \geq p_2 \cdots \geq p_{r-1} \geq p' \geq p'',$$

то и $C_{\varphi'}$ — оптимальный префиксный код.

Д-во. **1.** Пусть $C_{\varphi'}$ — оптимальный префиксный код. Предположим, что код C_φ не является оптимальным. Значит, найдется оптимальный префиксный код $C_{\varphi_1} = \{D_1, \ldots, D_{r-1}, D_r\}$. Отметим, что $c(\varphi_1) < c(\varphi)$. Рассмотрим префиксный код $C_{\varphi_1'} = \{D_1, \ldots, D_{r-1}, D_r0, D_r1\}$. Получаем:

$$c(\varphi_1') - c(\varphi') = (c(\varphi_1) + p_r) - (c(\varphi) + p_r) = c(\varphi_1) - c(\varphi) < 0.$$

Значит, $c(\varphi_1') < c(\varphi')$, чего не может быть, т.к. $C_{\varphi'}$ — оптимальный код. Следовательно, код C_φ — оптимальный.
2. Пусть теперь C_φ — оптимальный префиксный код и $p_1 \geq p_2 \geq \cdots \geq p_{r-1} \geq p' \geq p''$. Предположим, что код $C_{\varphi'}$ — не является оптимальным.
Значит, найдется оптимальный префиксный код $C_{\varphi_1''}$, имеющий вид $\{D_1, \ldots, D_{r-1}, D_r0, D_r1\}$. Отметим, что $c(\varphi_1') < c(\varphi)$. Рассмотрим префиксный код $C_{\varphi_1} = \{D_1, \ldots, D_{r-1}, D_r\}$. Получаем:

$$c(\varphi_1) - c(\varphi) = (C(\varphi_1') - p_r) - (c(\varphi') - p_r) = c(\varphi_1') - c(\varphi') < 0.$$

Значит, $c(\varphi_1) < c(\varphi)$, чего не может быть, т.к. C_φ — оптимальный код. Следовательно, код $C_{\varphi'}$ — оптимальный. □

1.15 Коды, обнаруживающие и исправляющие ошибки. Критерии кодов, обнаруживающих и исправляющих t ошибок замещения. Функция $M_t(n)$, ее оценки.

Опр. *Код $C_\varphi \subseteq B^*$ назовем обнаруживающим t ошибок, если для любого слова $\beta \in C_\varphi$ выполняется следующие условие: если в слове β произойдет не более чем t ошибок замещения и при этом оно перейдет в слово β', то по неправильному слову β' можно установить, что ошибки были.*

Опр. *Код $C_\varphi \subseteq B^*$ назовем исправляющем t ошибок, если для любого слова $\beta \in C_\varphi$ выполняется следующие условие: если в слове β произойдет не более t ошибок замещения и при этом оно перейдет в слово β', то по неправильному слову β' можно:*

1. *установить, что ошибки были;*

2. *в случае, когда ошибки были, восстановить правильное слово* β.

Теорема. *Пусть* $B = \{0,1\}$ *и* $C \subseteq B^n$ *— равномерный код,* $n \geq 1$. *Код* C *обнаруживает* t *ошибок замещения тогда и только тогда, когда* $d_C \geq t+1$.

Д-во. Пусть $\beta \in C$, в слове β произошло не более t ошибок замещения, и оно перешло в слово $\beta' \in B^n$. Значит $\beta' \in S_t(\beta)$. Тогда можно установить, что ошибки были, в том и только в том случае, когда β' не совпадает ни с каким кодовым словом из C, не равных слову β. Другими словами, когда никакому шару радиуса t с центром в кодовом слове из C не принадлежит никакое другое кодовое слово из C. Т.е. когда $d_C \geq t+1$. □

Теорема. *Пусть* $B = \{0,1\}$ *и* $C \subseteq B^n$ *— равномерный код,* $n \geq 1$. *Код* C *исправляет* t *ошибок замещения тогда и только тогда, когда* $d_C \geq 2t+1$.

Д-во. Пусть $\beta \in C$, в слове β произошло не более t ошибок замещения, и оно перешло в слово $\beta' \in B^n$. Значит, $\beta' \in S_t(\beta)$. Тогда можно установить, что ошибки были и, кроме того, их исправить, в том и только в том случае, когда β' не совпадает ни с каким словом из B^n, в которое может перейти некоторое кодовое слово из C, не равное слову β, при условии, что в нем произойдет не более t ошибок замещения. Другими словами, когда никакие два шара радиуса t с центрами в различных кодовых словах из C не пересекаются. Т.е. когда $d_C \geq 2t+1$. □

Опр. *Пусть* $M_t(n)$ *обозначает наибольшее число кодовых слов в коде* $C \subseteq B^n$, *исправляющем* t *ошибок замещения.*

Теорема. *При* $t \geq 1,\ n \geq 1$ *справедливы следующие неравенства:*

$$\frac{2^n}{S_{2t}(n)} \leq M_t(n) \leq \frac{2^n}{S_t(n)},$$

где $S_r(n)$ *обозначает число наборов в шаре радиуса* r *из* B^n.

Д-во. Верхняя оценка. Пусть $C \subseteq B^n$ — код, исправляющий t ошибок. Тогда никакие два шара радиуса t с центрами в различных кодовых словах из C не пересекаются. Поэтому

$$|C| \leq \frac{|B^n|}{S_t(n)} = \frac{2^n}{S_t(n)}.$$

Нижняя оценка. По индукции построим код $C \subseteq B^n$, исправляющий t ошибок, в котором не менее $\frac{2^n}{S_{2t}(n)}$ слов.
Базис индукции. Пусть $C_1 = \{\beta_1\}$, где β_1 — произвольное слово из B^n. Заметим, что код C_1 исправляет t ошибок.
Индуктивный переход. Пусть уже построен код $C_k = \{\beta_1, \ldots, \beta_k\} \subseteq B^n$, исправляющий t ошибок. Попытаемся к нему так добавить еще одно слово из B^n, чтобы получился код, исправляющий t ошибок.
Каждое слово β_i запрещает добавлять все слова из шара $S_{2t}(\beta_i),\ i = \overline{1,k}$, т.е. Каждое слово из C_k запрещает $S_{2t}(n)$ слов из B^n. Поэтому все слова из C_k запрещают не более $k \cdot S_{2t}(n)$ слов

из B^n. Значит, если $k \cdot S_{2t}(n) < |B^n| = 2^n$, то еще хотя бы одно новое слово можно добавить к коду C_k, чтобы получить код C_{k+1} исправляющий t ошибок.
Пусть построен код $C_m = \{\beta_1, \ldots, \beta_m\} \subseteq B^n$, исправляющий t ошибок, $m \geq 1$, для которого выполняется неравенство $m \cdot S_{2t}(n) \geq |B^n| = 2^n$. Тогда положим $C = C_m$, $|C| = m$. Тогда выполняется условие: $m \cdot S_{2t}(n) \geq 2^n$, а значит

$$|C| = m \geq \frac{2^n}{S_{2t}(n)}.$$

□

1.16 Коды, исправляющие одну ошибку. Коды Хэмминга. Оценка функции $M_1(n)$.

Рассмотрим один вид кодов, исправляющих одну ошибку. Они основаны на свойствах представления натуральных чисел в позиционной системе счисления. Такие коды называются кодами Хэмминга.

Пусть $n \geq 3$, $2^{k-1} < n < 2^k$, где $k \in \mathbb{N}$, и $N_n = \{1, 2, \ldots, n\}$. Любое число $s \in N_n$ можно представить в двоичной системе счисления с k разрядами: $s_{k-1} \ldots s_1 s_0$, где $s_{k-1}, \ldots, s_1, s_0 \in B$ и $s = \sum\limits_{i=0}^{k-1} s_i \cdot 2^i$. Для каждого $i = \overline{1, k-1}$ положим:

$$D_i = \{s \in N_n : s_i = 1\}.$$

Другими словами, в D_i содержатся все натуральные числа, не превосходящие n, в двоичном представлении которых i-й разряд равен 1.

Утверждение. *Пусть $D_0, D_1, \ldots, D_{k-1}$ — введенные выше множества. Тогда:*

1. *$2^i \in D_i$ и $2^j \notin D_i$ при $j \neq i$;*
2. *$\min\limits_{s \in D_i} s = 2^i$.*

Опр. *Пусть $n \geq 3$ и $2^{k-1} < n < 2^k$, где $k \in \mathbb{N}$. Множество $H \subseteq B^n$ называется кодом Хэмминга порядка n, если для любого набора $\beta \in H$ верна система уравнений:*

$$\begin{cases} \bigoplus\limits_{j \in D_0} \beta_j = 0, \\ \bigoplus\limits_{j \in D_1} \beta_j = 0, \\ \ldots \\ \bigoplus\limits_{j \in D_{k-1}} \beta_j = 0, \end{cases}$$

и, кроме того, H содержит все наборы из B^n, для которых эта система верна.

Теорема. *Пусть $n \geq 3$ и $2^{k-1} < n < 2^k$. Код Хэмминга порядка n содержит 2^{n-k} кодов и исправляет одну ошибку.*

Д-во. Пусть $H \subseteq B^n$ — код Хэмминга.

1. Сначала найдем число слов в коде H. Перепишем систему из определения кода Хэмминга в виде:

$$\begin{cases} \beta_{2^0} = \bigoplus\limits_{j \in D_0, j \neq 2^0} \beta_j, \\ \beta_{2^1} = \bigoplus\limits_{j \in D_1, j \neq 2^1} \beta_j, \\ \dots \\ \beta_{2^{k-1}} = \bigoplus\limits_{j \in D_{k-1}, j \neq 2^{k-1}} \beta_j. \end{cases}$$

Отметим, что $\beta_{2^0}, \beta_{2^1}, \dots, \beta_{2^{k-1}}$ в правых частях не встречаются. Поэтому если заданы $\beta_j \in B$ при $j \in N_n \setminus \{2^0, 2^1, \dots, 2^{k-1}\}$, то $\beta_{2^0}, \beta_{2^1}, \dots, \beta_{2^{k-1}}$ однозначно определяются этой системой. Число возможностей задать $\beta_j \in B$ при $j \in N_n \setminus \{2^0, 2^1, \dots, 2^{k-1}\}$ равно 2^{n-k}. Каждая из них определяет одно слово из H, а все они — слова из H. Значит, $|H| = 2^{n-k}$.

2. Теперь покажем, что код H исправляет одну ошибку замещения. Пусть $\beta \in H$, в слове β произошла ошибка в s-м разряде и оно перешло в слово $\beta' \in B^n$. Отметим, что $\beta'_i = \begin{cases} \beta_i, i \neq s, \\ \overline{\beta}_i, i = s. \end{cases}$

Пусть в двоичной системе счисления число s, $1 \leq s \leq n$, записывается как $s_{k-1} \dots s_1 s_0$, где $s_{k-1}, \dots, s_1, s_0 \in B$ и $s = \sum\limits_{i=1}^{k-1} s_i \cdot 2^i$. Для каждого $i = \overline{0, k-1}$ рассмотрим проверочную сумму: $\bigoplus\limits_{j \in D_i} \beta'_j$. Возможны два случая.

1) Если $s_i = 0$, то $s \notin D_i$. Поэтому $\beta'_j = \beta_j$ для всех $j \in D_i$. Получем:

$$\bigoplus_{j \in D_i} \beta'_j = \bigoplus_{j \in D_i} \beta_j = 0.$$

Значит, в этом случае верно:

$$\bigoplus_{j \in D_i} \beta'_j = s_i.$$

2) Если $s_i = 1$, то $s \in D_i$. Поэтому $\beta'_j = \beta_j$ для всех $j \in D_i,\ j \neq s,\ \beta'_s = \overline{\beta}_s = \beta_s \oplus 1$.

$$\bigoplus_{j \in D_i} \beta'_j = \left(\bigoplus_{j \in D_i, j \neq s} \beta_j \right) \oplus \beta'_s = \left(\bigoplus_{j \in D_i, i \neq s} \beta_j \right) \oplus (\beta_s \oplus 1) =$$
$$= \bigoplus_{j \in D_i} \beta_j \oplus 1 = 1.$$

Значит, и в этом случае верно:

$$\bigoplus_{j \in D_i} \beta'_j = s_i.$$

Значит, по неправильному слову β' можно найти все $s_0, s_1, \dots, s_{k-1}$ и разряд $s = \sum\limits_{i=0}^{k-1} s_i \cdot 2^i$, в котором произошла ошибка в слове β. Теперь если $s = 0$, то ошибки не было, а если $s \neq 0$, то можно восстановить правильное слово β. $\square$

Теорема. *При $n \geq 1$ справедливы следующие неравенства:*

$$\frac{2^n}{2n} \leq M_1(n) \leq \frac{2^n}{n+1}.$$

Д-во. **1.** *Верхняя оценка.* Известно, что

$$M_1(n) \leq \frac{2^n}{S_1(n)}.$$

Заметим, что $S_1(n) = n + 1$. Поэтому

$$M_1(n) \leq \frac{2^n}{n+1}.$$

2. *Нижняя оценка.* Если $n \leq 2$, то оценка верна. Поэтому пусть $n \geq 3$.
Сначала пусть $2^{k-1} < n < 2^k$, тогда $k = \lceil \log_2 n \rceil$.
Рассмотрим код Хэмминга порядка H порядка n. Он содержит 2^{n-k} слов и исправляет одну ошибку. Значит,

$$M_1(n) \geq |H| = 2^{n-k}.$$

Получаем:

$$M_1(n) \geq 2^{n-k} = \frac{2^n}{2^{\lceil \log_2 n \rceil}} \geq \frac{2^n}{2^{\log_2 n + 1}} = \frac{2^n}{2n}.$$

Теперь пусть $n = 2^k$, тогда $k = \log_2 n$. Рассмотрим код Хэмминга H порядка $(n-1)$ и построим по нему код $C \subseteq B^n$, добавив к каждому кодовому слову из H в конце 0.
Код C содержит 2^{n-1-k} слов и исправляет одну ошибку. Значит,

$$M_1(n) \geq |C| = 2^{n-1-k} = \frac{2^n}{2^{1+\log_2 n}} = \frac{2^n}{2n}.$$

□

1.17 Схемы из функциональных элементов и элементов задержки (СФЭЗ). Автоматность осуществляемых ими отображений.

Схемой из функциональных элементов с задержками (СФЭЗ) $S(x_1(t), \ldots, x_n(t); y_1(t), \ldots, y_m(t))$ в базисе $B = \{x\&y,\ x \vee y,\ \overline{x}\} \cup \{z\}$ называется

1. ориентированный граф $G = (V, E)$ с возможными ориентированными циклами, причем в графе G полустепень захода любой его вершины не превосходит двух;

2. любая вершина графа G с полустепенью захода, равной нулю, называется входной (или входом) и ей приписывается какая-то входная переменная $x_i(t)$;

3. любой вершине графа G с полустепенью захода, равной единице, приписывается либо единичная задржка z, либо отрицание $\overline{x}$;

4. в любом ориентированном цикле графа G должна быть хотя бы одна вершина с приписанной ей единичной задержкой;

5. любой вершине графа G с полустепенью захода, равной двум, приписывается либо конъюнкция $\&$, либо дизъюнкция $\vee$;

6. некоторый (в том числе и входные) вершины графа G называются выходными (или выходами) и им приписываются (различные) выходные переменный $y_1(t), \ldots, y_m(t)$.

Теорема. *Каждая СФЭЗ $S(x_1(t), \ldots, x_n(t); y_1(t), \ldots, y_m(t))$ осуществляет автоматное отображение входов $x_1(t), \ldots, x_n(t)$ в выходы $y_1(t), \ldots, y_m(t)$.*

Д-во. Рассмотрим граф $G = (V, E)$ СФЭЗ S. Пусть $v_1, \ldots, v_k \in V$ — все вершины, которым приписана единичная задержка z. Рассмотрим вершину v_i. В графе G в нее ведет одна луга из вершины, которую обозначим w_i. Удалим эту дугу (w_i, v_i) из графа G. Вершине w_i припишем новую выходную вершину $q_i(t)$. Вершина v_i станет входной, ей припишем новую входную переменную $p_i(t)$. Заметим, что т.к. в любом ориентированном цикле графа G хотя бы одной вершине была приписана z, выполнив такое преобразование для вершин $v_1, \ldots, v_k$, мы разорвем все ориентированные циклы.
В итоге получаем СФЭ (без задержек) S'. На ее выходах $y_j(t),\ q_i(t)$ соответственно вычисляются некоторые функции алгебры логики $F_j,\ G_i \in P_2$:

$$\begin{aligned} y_j(t) &= F_j(x_1(t), \ldots, x_n(t), p_1(t), \ldots, p_k(t)), \quad 1 \le j \le m, \\ q_i(t) &= G_i(x_1(t), \ldots, x_n(y), p_1(t), \ldots, p_k(t)), \quad 1 \le i \le k. \end{aligned}$$

По определению функции единичной задержки верно $p_i(t) = q_i(t-1),\ q_i(0) = 0$. Поэтому

$$\begin{cases} y_j(t) = F_j(x_1(t), \ldots, x_n(t), q_1(t-1), \ldots, q_k(t-1)), & 1 \le j \le m, \\ q_i(t) = G_i(x_1(t), \ldots, x_n(y), q_1(t-1), \ldots, q_k(t-1)), & \\ q_i(0) = 0, & 1 \le i \le k. \end{cases}$$

Получили каноническое уравнение, а значит, отображение — автоматное. □

1.18 Схемы из функциональных элементов и элементов задержки (СФЭЗ). Моделирование автоматной функции схемой из функциональных элементов и элементов задержки.

Теорема. *Каждый конечный автомат $\mathcal{A} = (A, B, Q, \varphi, \psi, q_*)$ (а значит и автоматная функция $f_{\mathcal{A}}$) может быть представлен СФЭЗ в базисе $B = \{x\&y,\ x \vee y,\ \overline{x}\} \cup \{z\}$ при некотором кодировании элементов из множеств $A,\ B,\ Q$ наборами из нулей и единиц.*

Д-во. Пусть $|A| = s,\ |B| = t,\ |Q| = r$. Закодируем взаимно однозначно:
1) элементы множества A наборами $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$, где $n = \lceil \log_2 s \rceil$;
2) элементы множества B — наборами $(y_1, \ldots, y_m) \in \{0,1\}^m$, где $m = \lceil \log_2 t \rceil$;
3) элементы множества Q — наборами $(q_1, \ldots, q_k) \in \{0,1\}^k$, где $k = \lceil \log_2 r \rceil$, причем начальное состояние q_* закодируем нулевым набором $(0, \ldots, 0)$.
Автомат $\mathcal{A}$ можно задать каноническими уравнениями:

$$\begin{cases} y(t) = \varphi(x(t), q(t-1)), \\ q(t) = \psi(x(t), q(t-1)), \\ q(0) = q_*. \end{cases}$$

Перепишем эти уравнения для кодов элементов из множеств $A,\ B,\ Q$. При этом функции φ и ψ преобразуют в наборы функций алгебры логики $(F_1, \dots, F_m)$ и $(G_1, \dots, G_k)$:

$$\begin{cases} y_j(t) = F_j(x_1(t), \dots, x_n(t), q_1(t-1), \dots, q_k(t-1)), & 1 \le j \le m, \\ q_i(t) = G_i(x_1(t), \dots, x_n(y), q_1(t-1), \dots, q_k(t-1)), & \\ q_i(0) = 0, & 1 \le i \le k. \end{cases} \quad (1)$$

Теперь построим СФЭ (без задержек) S' в базисе $B_0 = \{x\&y,\ x \vee y,\ \overline{x}\}$, вычисляющую на выходах $y_j(t),\ q_i(t)$ соответственно функции алгебры логики $F_j,\ G_i$:

$$\begin{cases} y_j(t) = F_j(x_1(t), \dots, x_n(t), q_1(t-1), \dots, q_k(t-1)), & 1 \le j \le m, \\ q_i(t) = G_i(x_1(t), \dots, x_n(y), q_1(t-1), \dots, q_k(t-1)), & 1 \le i \le k. \end{cases}.$$

Затем соединим в схеме S' выход $q_i(t)$ с входом $q_i(t-1)$ через единичную задержку z для всех $i = \overline{1,k}$. Получим СФЭЗ S, осуществляющую автоматное отображение в соответствии с каноническими уравнениями (1). □

1.19 Конечные автоматы. Отличимость состояний конечного автомата. Теорема Мура. Достижимость оценки теоремы Мура.

Опр. *Конечным автоматом $\mathcal{A}$ называется набор $(A, B, W, \varphi, \psi, q_*)$, в котором:*

1. *A — входной алфавит (являющийся конечным непустым множеством);*
2. *B — выходной алфавит (являющийся конечным непустым множеством);*
3. *Q — множество состояний (являющийся конечным непустым множеством);*
4. *$\alpha : A \times Q \to B$ — функция выходов;*
5. *$\psi : A \times Q \to Q$ — функция переходов;*
6. *$q_* \in Q$ — начальное состояние.*

Пусть $\mathcal{A} = (A, B, Q, \varphi, \psi)$ — конечный автомат (без начального состояния). По функциям φ и ψ определим функции

$$\overline{\varphi} : A^* \times Q \to B^* \quad \text{и} \quad \overline{\psi} : A^* \times Q \to Q.$$

Для всех $a \in A,\ \alpha \in A^*$, где $|\alpha| = m \ge 2$, и $q \in Q$ положим:

$$\begin{aligned} &\overline{\varphi}(\Lambda, q) = \Lambda, \\ &\overline{\varphi}(a, q) = \varphi(a, q), \\ &\overline{\varphi}(\alpha, q) = \varphi(\alpha(1), q)\overline{\varphi}(\alpha(2) \dots \varphi(m), \psi(\alpha(1), q)); \end{aligned}$$

$$\begin{aligned} &\overline{\psi}(\Lambda, q) = q, \\ &\overline{\psi}(a, q) = \psi(a, q), \\ &\overline{\psi}(\alpha, q) = \overline{\psi}(\alpha(2) \dots \alpha(m), \psi(\alpha(1), q)). \end{aligned}$$

Если $\mathcal{A} = (A, B, Q, \varphi, \psi)$, то
1) $\overline{\varphi}(\alpha, q)$ — слово $\beta \in B^*$, в которое автомат $\mathcal{A}$ преобразует слово $\alpha \in A^*$ из состояния $q \in Q$;
2) $\overline{\psi}(\alpha, q)$ — состояние $q' \in Q$, в которое автомат A переходит при преобразовании слова $\alpha \in A^*$ из состояния $q \in Q$.

Опр. *Слово $\alpha \in A^*$ отличает состояния $q' \in Q$ и $q'' \in Q$, если*

$$\overline{\varphi}(\alpha, q') \neq \overline{\varphi}(\alpha, q'').$$

Опр. *Два состояния $q' \in Q$ и $q'' \in Q$ называются отличимыми, если найдется слово $\alpha \in A^*$, которое их отличает. В обратном случае состояния $q' \in Q$ и $q'' \in Q$ называются неотличимыми, или эквивалентными.*

Лемма. *Пусть $\mathcal{A} = (A, B, Q, \varphi, \psi)$ — конечный автомат без начального состояния и состояния $q' \in Q$ и $q'' \in Q$ отличимы каким-то словом длины m и не отличимы никаким словом меньшей длины. Тогда для каждого $k = \overline{1, m}$ найдутся состояния $q'_k \in Q$ и $q''_k \in Q$, который отличимы каким-то словом длины k и не отличимы никаким словом меньшей длины.*

Д-во. Пусть $\mathcal{A} = (A, B, Q, \varphi, \psi)$ — конечный автомат и состояния $q' \in Q$ и $q'' \in Q$ отличимы словом $\alpha \in A^*$ длины и не отличимы никаким словом меньшей длины. Для каждого $k = \overline{1, m}$ определим состояния $q'_k,\ q''_k$:

$$q'_k = \overline{\psi}(\alpha(1) \dots \alpha(m-k), q') \in Q,$$
$$q''_k = \overline{\psi}(\alpha(1) \dots \alpha(m-k), q'') \in Q.$$

1. Состояния q'_k и q''_k отличимы словом $\alpha_k = \alpha(m-k+1) \dots \alpha(m)$ длины k.
2. Докажем от обратного, что состояния q'_k и q''_k не отличимы никаким словом меньшей длины. Пусть найдется слово $\alpha_0 \in A^*$ длины $k_0 < k$, отличающие состояния q'_k и q''_k. Но тогда состояния q' и q'' отличимы словом

$$\alpha_1 = \alpha(1) \dots \alpha(m-k)\alpha_0$$

длины $(m-k) + k_0 < m$, что противоречит условию. Следовательно, состояния q'_k и q''_k не отличимы никаким словом длины, меньшей k. □

Теорема (Мура). *Пусть $\mathcal{A} = (A, B, Q, \varphi, \psi)$ — конечный автомат с r состояниями ($|Q| = r$). Если состояния q' и q'' отличимы, то они отличимы некоторым словом длины, не большей $(r-1)$.*

Д-во. Пусть $Q = \{q_1, \dots, q_r\}$. Для каждого $k = 0, 1, \dots$ рассмотрим следующее отношение R_k на множестве Q: для $q_i, q_j \in Q$ верно $q_i R_k q_j$, если состояния q_i и q_j не отличимы никаким словом длины, меньшей или равной k. Полагаем, что $q_i R_0 q_j$ для всех $q_i, q_j \in Q$.
Докажем, что для каждого $k = 0, 1, \dots$ R_k — отношение эквивалентности на Q.
1. *Рефлексивность*: qR_kq для каждого $q \in Q$.
2. *Симметричность*: если $q_i R_k q_j$, то $q_j R_k q_s$.
3. *Транзитивность*: пусть $q_i R_k q_j$ и $q_j R_k q_s$, т.е. для каждого такого $\alpha \in A^*$, что $|\alpha| \leq k$, верно

$$\overline{\varphi}(\alpha, q_i) = \overline{\varphi}(\alpha, q_j),$$
$$\overline{\varphi}(\alpha, q_j) = \overline{\varphi}(\alpha, q_s).$$

Тогда для каждого такого $a \in A^*$, что $|\alpha| \le k$, верно и $\overline{\varphi}(\alpha, q_i) = \overline{\varphi}(\alpha, q_s)$, т.е. $q_i R_k q_s$.
Следовательно, R_k — отношение эквивалентности на Q.
Пусть $r_k = |Q/R_k|$ — число классов эквивалентности по отношению R_k на множестве Q. Заметим, что $r_0 = 1$.
По условию состояния $q' \in Q$ и $q'' \in Q$ — отличимы. Пусть $\alpha \in A^*$ — слово наименьшей длины, отличающее состояния q' и q''. Пусть $|\alpha| = m$. По лемме для каждого $k = \overline{1, m}$, найдутся состояния $q'_k \in Q$ и $q''_k \in Q$, который отличимы каким-то словом длины k и не отличимы никаким словом меньшей длины.
Посмотрим, как устроены фактор-множества Q/R_{k-1} и Q/R_k и как соотносятся между собой числа r_{k-1} и r_k при $1 \le k \le m$. Заметим, что если $q_i \overline{R}_{k-1} q_j$, то $q_i \overline{R}_k q_j$. Т.е. если состояния q_i и q_j отличимы каким-то словом длины, не большей $(k-1)$, то состояния q_i и q_j отличимы и каким-то словом длины, не большей k. Поэтому $r_{k-1} \le r_k$.
Рассмотрим состояния q'_k и q''_k. Они не отличимы никаким словом длины, меньшей k. Значит, по отношению R_{k-1} они находятся в одном классе эквивалентности. Но они отличимы каким-то словом длины k. Значит, по отношению R_k они находятся в разных классах эквивалентности. Следовательно, при переходе от фактор-множества Q/R_{k-1} к фактор-множеству Q/R_k хотя бы один класс эквивалентности по отношению R_{k-1} разбивается хотя бы на два класса эквивалентности по отношению R_k. Поэтому $r_{k-1} < r_k$.
Отметим, что т.к. $|Q| = r$, для всех k верно $r_k \le r$. Получаем возрастающую последовательность чисел:

$$1 = r_0 < r_1 < r_2 < \cdots < r_m \le r.$$

Следовательно, $m \le r - 1$. □

Достижимость оценки теоремы Мура. Для каждого $r \ge 2$ приведем пример конечного автомата $\mathcal{A} = (A, B, Q, \varphi, \psi)$ с r состояниями ($|Q| = r$), в котором найдутся два состояния, отличимые словом длины $r - 1$, но не отличимые никаким словом меньшей длины.
Рассмотрим автомат, который дает на выход последовательность $\underbrace{0 \ldots 0}_{r-1} 10 \ldots 01 \ldots$. Понятно, что для этого автомата нужно r различных состояний, где для $i = \overline{1, r-1}$ в состоянии q_i независимо от $x(t)$ записывается 0 и совершается переход в состояние q_{i+1}, а в состоянии q_r записывается r и совершается переход в q_r.
Тогда состояния q_1 и q_2 неотличимы никаким словом длины, не превосходящей $r - 2$, т.к. на выходе будет $\underbrace{0 \ldots 0}_{r-2}$. А вот на слове α длины $r - 2$: $\overline{\varphi}(\alpha, q_1) = 0 \ldots 00$, $\overline{\varphi}(\alpha, q_2) = 0 \ldots 01$.

1.20 Схемы из функциональных элементов. Сумматор, верхняя оценка его сложности.

Опр. *Схемой из функциональных элементов (СФЭ)* $S(x_1, \ldots, x_n; y_1, \ldots, y_m)$ *в базисе* $B_0 = \{x\&y,\ x \vee y,\ \overline{x}\}$ *называется*

1. *ориентированный граф* $G = (V, E)$ *без ориентированных циклов, причем в графе* G *полустепень захода любой его вершины не превосходит двух;*

2. *любая вершина графа* G *с полустепенью захода, равной нулю, называется входной (или входом) и ей приписывается какая-то входная переменная* x_i*;*

3. *любая вершина графа G с полустепенью захода, не равной нулю, называется внутренней;*

4. *любой вершине графа G с полустепенью захода, равной единице, приписывается отрицание $\overline{x}$;*

5. *любой вершине графа G с полустепенью захода, равной двум, приписывается либо конъюнкция $\&$, либо дизъюнкция $\vee$;*

6. *некоторый (входные или внутренние) вершины графа G называются выходными (или выходами) и им приписываются (различные) входные переменные $y_1, \ldots, y_m$.*

Опр. *Сложностью $L(S)$ СФЭ S называется число ее внутренних вершин.*

Если $(x_1, \ldots, x_n) \in E_2^n$, где $E_2 = \{0, 1\}$, то положим

$$(x_1, \ldots, x_n)_2 = \sum_{i=1}^{n} x_i \cdot 2^{n-i}.$$

Опр. *Сумматором S_n порядка $n \geq 1$, называется такая СФЭ с $2n$ входами $x_1, \ldots, x_n$, $y_1, \ldots, y_n$ и $n+1$ выходами $z_0, z_1, \ldots, z_n$, что*

$$(z_0, z_1, \ldots, z_n)_2 = (x_1, \ldots, x_n)_2 + (y_1, \ldots, y_n)_2.$$

Сумматор S_n также называется n-разрядным сумматором.

Построим одноразрядный сумматор $S_1(x, y; z_0, z_1)$. Найдем функции $z_0(x, y)$ и $z_1(x, y)$:

x	y	z_0	z_1
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$z_0 = x \cdot y, \quad z_1 = (x \vee y) \cdot (\overline{x} \vee \overline{y}) = (x \vee y) \cdot (\overline{x \cdot y}).$$

Значит, в базисе B_0 можно построить сумматор S_1 со сложностью 4.

С какой сложностью можно построить сумматор S_n, $n \geq 1$? Вспомним алгоритм сложения n-разрядных чисел "в столбик". При сложении каждого разряда i (кроме младшего) складывают x_i, y_i и разряд переноса p_i. При этом получается двухразрядное число $q_i z_i$, где q_i — старший, а z_i — младший разряды. Теперь z_i является разрядом i суммы этих n-разрядных чисел, а q_i — разрядом переноса в следующем, более старшем разряде.
Назовем ячейкой сумматора S СФЭ с тремя входами x, y, p и двуся выходами q, z, которая вычисляет описанное выше преобразование входов в выходы, а именно,

$$q = x \cdot y \vee x \cdot p \vee y \cdot p, \quad z = x \oplus y \oplus p.$$

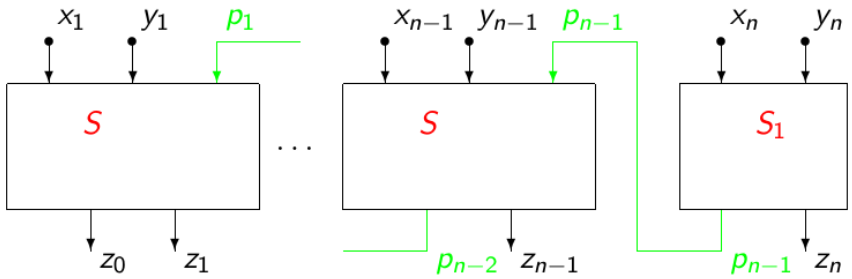
Отметим, что

$$q = x \cdot y \vee p \cdot (x \oplus y), \quad z = (x \oplus y) \oplus p.$$

Значит, что в базисе B_0 можно построить ячейку сумматора S со сложностью 9.

Теорема. *В базисе* $B_0 = \{x \cdot y,\ x \vee y,\ \overline{x}\}$ *можно построить сумматор* S_n *со сложностью* $9n - 5$.

Д-во. Применим алгоритм сложения n-разрядных чисел $(x_1, \ldots, x_n)_2$ и $(y_1, \ldots, y_n)_2$ "в столбик".
Сначала возьмем одноразрядный сумматор S_1 и припишем его входам x_n и y_n. Младший разряд выхода этого одноразрядного сумматора S_1 назовем выходом z_n, а старший разряд его выхода обозначим p_{n-1}.
Далее для каждого $i = \overline{n-1, 1}$ повторим следующие рассуждения. Возьмем новую ячейку сумматора S, придадим ей номер i и двум ее входам припишем x_i, y_i, а на третий вход направим p_i. Младший разряд выхода этой ячейки сумматора S с номером i назовем выходном z_i, а старший разряд ее выхода обозначим p_{i-1} при $i \geq 2$ и назовем выходом z_0 при $i = 1$.



Полученная в итоге СФЭ является n-разрядным сумматором S_n. оценим его сложность:

$$L(S_n) \leq (n-1)L(S) + L(S_1) \leq 9(n-1).$$

□

1.21 Схемы из функциональных элементов. Вычитатель, верхняя оценка его сложности.

Опр. *Вычитателем* W_n *порядка* $n \geq 1$ *называется такая СФЭ с* $2n$ *входами* $x_1, \ldots, x_n$, $y_1, \ldots, y_n$ *и* n *выходами* $u_1, \ldots, u_n$, *что*

$$(u_1, \ldots, u_n)_2 = (x_1, \ldots, x_n)_2 - (y_1, \ldots, y_n)_2,$$

если $(x_1, \ldots, x_n)_2 \geq (y_1, \ldots, y_n)_2$. *Если первое из этих чисел меньше второго, то входы неправильные, и не важно, что вычисляется на выходах.*

Построим одноразрядный вычитатель $W_1(x, y; u)$. Найдем функцию $u(x, y)$:

x	y	u
0	0	0
0	1	–
1	0	1
1	1	0

Например:

$$u = x \cdot \overline{y}.$$

Значит, в базисе B_0 можно построить вычитатель W_1 со сложностью 2.

Лемма. *Если* $x_1, \ldots, x_n \in E_2$, *то*

$$(x_1, \ldots, x_n)_2 + (\overline{x}_1, \ldots, \overline{x}_n) = 2^n - 1.$$

Теорема. *В базисе* $B_0 = \{x \cdot y,\ x \vee y,\ \overline{x}\}$ *можно построить вычитатель* W_n *со сложностью* $11n - 5$.

Д-во. Построим вычитатель W_n в соответствии с тождеством:

$$(u_1, \ldots, u_n)_2 = 2^n - 1 - ((y_1, \ldots, y_n)_2 + (2^n - 1 - (x_1, \ldots, x_n)_2)).$$

При построении применим вспомогательную лемму.
Оценим сложность полученного вычитателя W_n:

$$L(W_n) \leq 2n + L(S_n) \leq 2n + 9n - 5 = 11n - 5.$$

□

1.22 Схемы из функциональных элементов (СФЭ). Умножитель. Метод Карацубы построения умножителя, верхняя оценка его сложности.

Заметим, что

$$0 \leq (x_1, \ldots, x_n)_2 < 2^n,$$
$$0 \leq (y_1, \ldots, y_n)_2 < 2^n,$$

поэтому

$$0 \leq (x_1, \ldots, x_n)_2 \cdot (y_1, \ldots, y_n)_2 < 2^{2n}.$$

Опр. *Умножителем* M_n *порядка* $n \geq 1$ *называется такая СФЭ с* $2n$ *входами* $x_1, \ldots, x_n$, $y_1, \ldots, y_n$ *и* $2n$ *выходами* $z_1, \ldots, x_n$, *что*

$$(z_1, \ldots, z_{2n})_2 = (x_1, \ldots, x_n)_2 \cdot (y_1, \ldots, y_n)_2.$$

Умножитель Mn также называется n-разрядным умножителем.

Построим одноразрядный умножитель $M_1(x, y; z)$. Найдем функцию $z(x, y)$:

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

$$z = x \cdot y.$$

Значит, в базисе B_0 можно построить умножитель M_1 со сложностью 1.

Пусть M'_n обозначает СФЭ с $n+1$ входами $x_1, \ldots, x_n, y$ и n выходами $z_1, \ldots, z_n$, которая вычисляет умножение n-разрадного числа $(x_1, \ldots, x_n)_2$ на разряд y, т.е.

$$(z_1, \ldots, z_n)_2 = (x_1, \ldots, x_n) \cdot y.$$

Лемма. *В базисе $B_0 = \{x \cdot y,\ x \vee y,\ \overline{x}\}$ модно построить схему M'_n со сложностью n.*

Д-во. Действительно, достаточно заметить, что $z_i = x_i \cdot y$ для всех $i = \overline{1, n}$. □

Пусть $M''_{n,m}$ обозначает СФЭ с n входами $x_1, \ldots, x_n$ и $n+m$ выходами $z_1, \ldots, z_{n+m}$, которая вычисляет умножение n-разрядного числа $(x_1, \ldots, x_n)$ на число 2^m, т.е.

$$(z_1, \ldots, z_{n+m})_2 = (x_1, \ldots, x_n)_2 \cdot 2^m.$$

Лемма. *В базисе $B_0 = \{x \cdot y,\ x \vee y,\ \overline{x}\}$ можно построить схему $M''_{n,m}$ с константой сложностью.*

Д-во. Действительно, достаточно заметить, что $z_i = x_i$ для всех $i = \overline{1, n}$, а $z_{n+1} = \cdots = z_{n+m} = 0$. Поэтому сложность схемы можно оценить сложностью вычисления константы 0, а эта сложность — константна. □

Лемма. *В базисе $B_0 = \{x \cdot y,\ x \vee y,\ \overline{x}\}$ для каждого $n \geq 1$ и любого умножителя M_n можно построить такой умножитель M_{n+1}, что*

$$L(M_{n+1}) \leq L(M_n) + C_1 n,$$

где $C_1 \geq 0$ — некоторое действительное число, не зависящее от n.

Д-во. Пусть $n \geq 1$. Рассмотрим произвольный умножитель M_n.
Пусть на входы умножителя M_{n+1} подают числа:

$$x = (x_0, x_1, \ldots, x_n)_2, \quad y = (y_0, y_1, \ldots, y_n)_2.$$

Введем обозначения:

$$x' = (x_1, \ldots, x_n)_2, \quad y' = (y_1, \ldots, y_n)_2.$$

Тогда:

$$\begin{aligned} x &= (x_0, x_1, \ldots, x_n)_2 = x_0 \cdot 2^n + (x_1, \ldots, x_n)_2 = x_0 \cdot 2^n + x', \\ y &= (y_0, y_1, \ldots, y_n)_2 = y_0 \cdot 2^n + (y_1, \ldots, y_n)_2 = y_0 \cdot 2^n + y'. \end{aligned}$$

Получаем:

$$\begin{aligned} x \cdot y &= (x_0 \cdot 2^n + x')(y_0 \cdot 2^n + y') = \\ &= x_0 \cdot y_0 \cdot 2^{2n} + (x_0 \cdot y' + x' \cdot y_0) \cdot 2^n + x' \cdot y'. \end{aligned}$$

Значит, для умножения $(n+1)$-разрядных чисел можно умножить n-разрядные числа и выполнить дополнительные вычисления. При этом сложность этих дополнительных вычислений не превосходит $C_1 \cdot n$, где $C_1 > 0$ — действительное число, не зависящее от n. □

Лемма (Основная)**.** *В базисе* $B_0 = \{x \cdot y,\ x \vee y,\ \overline{x}\}$ *для каждого* $n \geq 1$ *и любого умножителя* M_n *можно построить такой умножитель* M_{2n}*, что*

$$L(M_{2n}) \leq 3L(M_n) + C_2 n,$$

где $C_2 < 0$ *— некоторое действительно число, не зависящее от* n*.*

Д-во. Пусть $n \geq 1$. Рассмотрим произвольный умножитель M_n.
Пусть на вход умножителя M_{2n} подаются числа:

$$x = (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n})_2, \quad y = (y_1, \ldots, y_n, y_{n+1}, \ldots, y_{2n})_2.$$

Введем обозначения:

$$\begin{aligned} x' &= (x_1, \ldots, x_n)_2, & x'' &= (x_{n+1}, \ldots, x_{2n})_2, \\ y' &= (y_1, \ldots, y_n)_2, & y'' &= (y_{n+1}, \ldots, y_{2n})_2. \end{aligned}$$

Тогда:

$$\begin{aligned} x &= x' \cdot 2^n + x'', \\ y &= y' \cdot 2^n + y''. \end{aligned}$$

Получаем:

$$\begin{aligned} x \cdot y &= (x' \cdot 2^n + x'')(y' \cdot 2^n + y'') = \\ &= x' \cdot y' \cdot 2^{2n} + (x' \cdot y'' + x'' \cdot y') \cdot 2^n + x'' \cdot y''. \end{aligned}$$

Рассмотрим тождество:

$$x' \cdot y'' + x'' \cdot y' = (x' + x'') \cdot (y' + y'') - x' \cdot y' - x'' \cdot y''.$$

Значит

$$\begin{aligned} x \cdot y = {} & x' \cdot y' \cdot 2^{2n} + \\ & + ((x' + x'') \cdot (y' + y'') - x' \cdot y' - x'' \cdot y'') \cdot 2^n + \\ & + x'' \cdot y'' \end{aligned}$$

Значит, для умножения $2n$-разрядных чисел можно трижды умножить n-разрядные числа и выполнить дополнительные вычисления. При этом сложность этих дополнительных вычислений не превосходит $C_2 \cdot n$, где $C_2 > 0$ — действительное число, не зависящее от n. □

Теорема (Карацубы)**.** *В базисе* $B_0 = \{x \cdot y,\ x \vee y,\ \overline{x}\}$ *можно построить умножитель* M_n *со сложностью* $O(n^{\log_2 3})$*.*

Д-во. **1.** Сначала рассмотрим случай $n = 2k$, где $k \in \mathbb{N}$. Получаем:

$$\begin{aligned} L(M_{2^k}) &\leq 3L(M_{2^{k-1}}) + C_2 \cdot 2^{k-1} \leq \\ &\leq 3(3L(M_{2^{k-2}}) + C_2 \cdot 2^{k-1}) + C_2 \cdot 2^{k-1} = \\ &= 3^2 L(M_{2^{k-2}}) + C_2(3 \cdot 2^{k-2} + 2^{k-1}) \leq \cdots \leq \\ &\leq 3^k L(M_{2^0}) + C_2(3^{k-1} + \cdots + 3 \cdot 2^{k-2} + 2^{k-1}). \end{aligned}$$

Заметим, что $L(M_1) = 1$. Кроме того,

$$3^{k-1} + \cdots + 3 \cdot 2^{k-2} + 2^{k-1} = 3^{k-1} \cdot \left(1 + \cdots + \left(\frac{2}{3}\right)^{k-1}\right) \leq$$
$$\leq 3^{k-1} \frac{1}{1 - \frac{2}{3}} = 3^k.$$

Поэтому:

$$L(M_{2^k}) \leq 3^k + C_2 3^k \leq C_3 \cdot 3^k,$$

где $C_3 = C_2 + 1 > 0$ — некоторое действительное число. Но $n = 2^k$, значит,

$$L(M_n) \leq C_3 \cdot 3^k = C_3 \cdot 2^{k \log_2 3} = C_3 \cdot n^{\log_2 3} = O(n^{\log_2 3}).$$

2. Теперь рассмотрим случай $2^{k-1} < n < 2^k$, где $k \in \mathbb{N}$.
Добавим к n-разрядным числам нули слева, чтобы получились 2^k-разрядные числа. Тогда:

$$L(M_n) \leq L(M_{2^k}) \leq C_3 \cdot 2^{k \log_2 3} = (C_3 \cdot 2^{\log_2 3}) \cdot 2^{(k-1) \log_2 3} \leq C \cdot n^{\log_2 3},$$

где $C = C_3 \cdot 2^{\log_2 3} > 0$ — некоторое действительное число. Значит

$$L(M_n) \leq C \cdot n^{\log_2 3} = O(n^{\log_2 3}).$$

□

Известен алгоритм Шенхаге-Штрассена, который n-разрядные числа позволяет умножать со сложностью $O(n \cdot \log n \cdot \log \log n)$.

2 Часть Б.

2.1 Функции алгебры логики. Существенность переменных. Формулы. Тождества.

Опр. *Пусть $E_2 = \{0, 1\}$. Функцией алгебры логики называется произвольное отображение из E_2^n в E_2, $n \geq 1$. Множество всех функций алгебры логики, зависящих от n переменных, обозначит $P_2^{(n)}$, а множество всех функций алгебры логики — $P_2 = \bigcup\limits_{n \geq 1} P_2^{(n)}$.*

Опр. *Переменная x_i называется существенной для функции $f(x_1, \ldots, x_n) \in P_2$, если найдутся такие элементы $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n \in E_2$, что*

$$f(a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n) \neq f(a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n).$$

Переменная, не являющаяся существенной, называется несущественной, или фиктивной. Как правило, мы будем рассматривать функции с точностью до несущественных переменных. Т.е. будем считать, что несущественные переменные можно добавлять и убирать.

Опр. *Формула над множеством A определяется по индукции.*
Базис индукции. Если f — обозначение m-местной функции из A и $x_1, \ldots, x_m$ — переменные (из X), причем не обязательно различные, то выражение $f(x_1, \ldots, x_m)$ — формула.
Индуктивный переход. Если f — обозначение m-местной функции из A и $F_1, \ldots, F_m$ — уже построенные формулы или переменные (не обязательно различные), то выражение $f(F_1, \ldots, F_m)$ — формула.

Опр. *Формулы F_1 и F_2 называются эквивалентными, если они определяют равные функции, т.е. функции f_{F_1} и f_{F_2} равны. Обозначение эквивалентных формул: $F_1 = F_2$; при этом равенство $F_1 = F_2$ называется тождеством.*

Верны следующие тождества:

- коммутативность связок $\cdot, \vee, \oplus, \sim, /, \downarrow$;
- ассоциативность связок $\cdot, \vee, \oplus$;
- дистрибутивность видов

$$(x \vee y) \cdot z = x \cdot z \vee y \cdot z;$$
$$(x \cdot y) \vee z = (x \vee z) \cdot (y \vee z);$$
$$(x \oplus y) \cdot z = x \cdot z \oplus y \cdot z.$$

2.2 Функции алгебры логики. Теорема о разложении функции алгебры логики по переменным. Теорема о совершенной дизъюнктивной нормальной форме (ДНФ). Теорема о совершенной конъюнктивной нормальной форме (КНФ).

Опр. *Если $f(x_1, \ldots, x_n) \in P_2^{(n)}$ и $\sigma \in E_2^k$, $1 \le k \le n$, то положим*

$$f_\sigma(x_{k+1}, \ldots, x_n) = f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n).$$

Функция f_σ называется σ-подфункцией функции f по k первым переменным.

Если $\sigma \in E_2$, то введем обозначение: $x^\sigma = \begin{cases} x, & \sigma = 1 \\ \overline{x}, & \sigma = 0 \end{cases}$. Отметим, что $x^\sigma = 1$ в том и только в том случае, когда $x = \sigma$.

Теорема. *При $1 \le k \le n$ каждая функция $f(x_1, \ldots, x_n) \in P_2$ может быть представлена в виде:*

$$f(x_1, \ldots, x_n) = \bigvee_{\sigma \in E_2^k} x_1^{\sigma_1} \cdot \ldots \cdot x_k^{\sigma_k} \cdot f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n).$$

Д-во. Рассмотрим произвольный набор $\alpha \in E_2^n$ и подставим его в левую часть равенства из утверждения. Получаем:

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

Рассмотрим набор $\beta \in E_2^k$, где $\beta_i = \alpha_i,\ i = \overline{1,k}$. Набор σ пробегает все наборы множества E_2^k, а набор β — какой-то набор из E_2^k.

1. Если $\sigma \neq \beta$, то найдется такое $i,\ 1 \leq i \leq k$, что $\sigma_i \neq \alpha_i$. Значит, $\alpha_i^{\sigma_i} = 0$, откуда в этом случае

$$\alpha_1^{\sigma_1} \cdot \dots \cdot \alpha_{i-1}^{\sigma_{i-1}} \cdot 0 \cdot \alpha_{i+1}^{\sigma_{i+1}} \cdot \dots \cdot \alpha_k^{\sigma_k} \cdot f(\sigma_1, \dots, \sigma_k, \alpha_{k+1}, \dots, \alpha_n) = 0.$$

2. Если $\sigma = \beta$, то для всех $i,\ i = \overline{1,k}$, верно $\sigma_i = \alpha_i$, а значит, $\alpha_i^{\sigma_i} = 1$. Поэтому в этом случае

$$\alpha_1^{\sigma_1} \cdot \dots \cdot \alpha_k^{\sigma_k} f(\sigma_1, \dots, \sigma_k, \alpha_{k+1}, \dots, \alpha_n) = f(\alpha).$$

Следовательно,

$$f(\alpha) = 0 \vee \dots \vee 0 \vee f(\alpha) \vee 0 \vee \dots \vee 0 = f(\alpha).$$

□

Опр. *Выражение (формула) вида*

$$x_{i_1}^{\sigma_1} \cdot \dots \cdot x_{i_k}^{\sigma_k},$$

где $x_{i_1}, \dots, x_{i_k}$ — различные переменные и $\sigma_1, \dots, \sigma_k \in E_n$, называется элементарной конъюнкцией (ЭК) ранга $k,\ k \geq 1$.

Опр. *Дизъюнктивной нормальной формой (ДНФ) длины $l, l \geq 1$, назовем дизъюнкцию l различных ЭК. ДНФ длины 0 назовем константу 0. Если каждая переменная содержит все переменные этой ДНФ, то такая ДНФ называется совершенной.*

Теорема. *Каждая функция $f(x_1, \dots, x_n) \in P_2,\ F \neq 0$, может быть представлена в виде совершенной ДНФ D_f, а именно:*

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

Д-во. Прямое следствие теоремы о дизъюнктивном разложении. □

Теорема. *При $1 \leq k \leq n$ каждая функция $f(x_1, \dots, x_n) \in P_2$ может быть представлена в виде:*

$$f(x_1, \dots, x_n) = \bigwedge_{\sigma \in E_2^k} (x_1^{\overline{\sigma}_1} \vee \dots \vee x_k^{\overline{\sigma}_k} \vee f(\sigma_1, \dots, \sigma_k, x_{k+1}, \dots, x_n)).$$

Д-во. Аналогично доказательству теоремы о дизъюнктивном разложении. □

Опр. *Выражение (формула) вида*

$$x_{i_1}^{\sigma_1} \vee \dots \vee x_{i_k}^{\sigma_k},$$

где $x_{i_1}, \dots, x_{i_k}$ — различные переменные и $\sigma_1, \dots, \sigma_k \in E_2$ называется элементарной дизъюнкцией (ЭД) ранга $k,\ k \geq 1$.

Опр. *Конъюнктивной нормальной формой (КНФ) длины $l,\ l \geq 1$, назовем конъюнкцию l различных ЭД. КНФ длины 0 назовем константу 1. Если каждая ЭД в КНФ содержит все переменные этой КНФ, то такая КНФ называется совершенной.*

Теорема. *Каждая функция $f(x_1, \dots, x_n) \in P_2,\ f \neq 0$, может быть представлена в виде совершенной КНФ K_f, а именно:*

$$f(x_1, \dots, x_n) = \bigwedge_{\sigma \in E_2^n : f(\sigma) = 0} (x_1^{\overline{\sigma}_1} \vee \dots \vee x_n^{\overline{\sigma}_n}).$$

Д-во. Прямое следствие из теоремы о конъюнктивном разложении. □

2.3 Функции алгебры логики. Полные системы. Примеры полных систем (с доказательством полноты).

Опр. *Пусть $A \subseteq P_2$. Множество A называется полной системой, если формулами над A можно выразить любую функцию алгебры логики.*

Утверждение. *Система $A = \{x \cdot y,\ x \vee y,\ \overline{x}\}$ является полной.*

Д-во. Рассмотрим произвольную функцию $f \in P_2$.
1. Если $f = 0$, то $f = \overline{x} \cdot x$.
2. Если $f \neq 0$, то представим f ее совершенной ДНФ. □

Утверждение. *Следующие множества являются полными системами:*

1. $A = \{\overline{x},\ x \cdot y\}$;
2. $A = \{\overline{x},\ x \vee y\}$;
3. $A = \{x/y\}$;
4. $A = \{x \downarrow y\}$.

Д-во. Система $B = \{x \cdot y,\ x \vee y,\ \overline{x}\}$ полная. Выразим все ее функции через функции систем их условия.

1. $x \vee y = \overline{\overline{x} \cdot \overline{y}}$.
2. $x \cdot y = \overline{\overline{x} \vee \overline{y}}$.
3. $\overline{x} = x/x,\ x \cdot y = \overline{x/y}$.
4. $\overline{x} = x \downarrow x,\ x \vee y = \overline{x \downarrow y}$.

□

2.4 Функции алгебры логики. Теорема Жегалкина о выразимости функции алгебры логики полиномом Жегалкина.

Опр. *Элементарная конъюнкция, не содержащая отрицаний переменных, называется монотонной (ЭК), или мономом, или одночленом.*

Опр. *Полиномом Жегалкина длины l, $l \geq 1$, назовем сумму по модулю два l различных монотонных ЭК. Полиномом Жегалкина длины 0 назовем константу 0.*

Теорема. *Каждая функция $f(x_1, \ldots, x_n) \in P_2$ может быть единственным образом представлена в виде полинома Жегалкина P_f.*

Д-во. Существование. Применим полиномиальное разложение функции $f(x_1, \ldots, x_n)$ по всем n переменным:

$$f(x_1, \ldots, x_n) = \bigoplus_{\sigma \in E_2^n} x_1^{\sigma_1} \cdot \cdots \cdot x_n^{\sigma_n} \cdot f(\sigma).$$

Затем пользуясь тождеством $x^\sigma = x \oplus \sigma \oplus 1$ везде в правой части заменим выражение $x_i^{\sigma_i}$ на выражение $x_i \oplus \sigma_i \oplus 1$. Далее по правилам коммутативности и ассоциативности & и $\oplus$ и дистрибутивности вида $x \cdot (y \oplus z) = x \cdot y \oplus z \cdot z$ перемножим все скобки. После этого приведем подобные слагаемые по правилам $x \oplus x = x,\ x \oplus 0 = x$. В итоге получим полином Жигалкина, который представляет исходную функцию f.
Единственность. Покажем, что число полиномов Жегалкина над переменными $x_1, \ldots, x_n$ совпадает с числом функций из $P_2^{(n)}$. Монотонных элементарных конъюнкций над переменными $x_1, \ldots x_n$ всего найдется 2^n, т.к. каждая переменная $x_i,\ i = \overline{1, n}$, может либо входить, либо не входить в такую монотонную ЭК. Далее, полиномов Жегалкина над переменными $x_1, \ldots, x_n$ всего найдется 2^{2^n}, т.к. каждая из 2^n монотонных ЭК может либо входить, либо не входить в такой полином Жегалкина. Значит, учитывая то, что каждая функция f из $P_2^{(n)}$ может быть представлена в виде полинома Жегалкина, это представление единственно. □

2.5 Функции алгебры логики. Замыкание, замкнутый класс. Функции, сохраняющие константу, и линейные функции. Замкнутость классов функций, сохраняющих константу, и линейных функций.

Опр. *Пусть $A \subseteq P_2$. Замыканием $[A]$ множества A называется множество всех функций, который могут быть выражены формулами над A.*

Опр. *Замыкание множества A можно определить по-другому. Замыкание $[A]$ называется множество всех функций из P_2, которые можно получить из функций множества A приминением слудующих операций:*

1. *добавлением или удалением несущественных переменных;*
2. *подстановкой в функции из A вместо переменных других переменных (не обязательно различных);*
3. *подстановкой в функции из A вместо переменных функций из A или функций, который уже получены.*

Операции 1–3 *называем операциями суперпозиции.*

Утверждение. *Два приведенных определения замыкания $A,\ A \subseteq P_2$, равносильны.*

Для произвольных множеств $A, B \subseteq P_2$ верны следующие утверждений:

1. $[P_2] = P_2$;
2. $A \subseteq [A]$;
3. если $A \subseteq B$, то $[A] \subseteq [B]$;

4. $[[A]] = [A]$.

Опр. *Пусть $A \subseteq P_2$. Множество A называется замкнутым классом, если $[A] = A$.*

Утверждение. *Пусть $A \subseteq P_2$, A — замкнутый класс и $A \neq P_2$. Тогда для любого множества B, $B \subseteq P_2$, верно: если $B \subseteq A$, то B — не полная система.*

Д-во. Итак, $B \subseteq A$. По свойствам замыкания и из условия получаем: $[B] \subseteq [A] = A \neq P_2$. Значит $[B] \neq P_2$, т.е. B — не полная система. □

Опр. *Функция $f(x_1, \ldots, x_n) \in P_2$ сохраняет 0, если $f(0, \ldots, 0) = 0$. Множество всех функций, сохраняющих 0, обозначим T_0.*

Опр. *Пусть $e_i^n \subseteq P_2^{(n)}$ и $e_i^n(x_1, \ldots, x_n) = x_i$, $i = \overline{1,n}$, $n \geq 1$. Положим $I = \{e_i^n : i = \overline{1,n},\ n \geq 1\}$, т.е. I — множество всех функций конгруэнтных тождественной функции.*

Лемма. *Пусть $A \subseteq P_2$ и $I \subseteq A$. Если для любых функций $f_0(y_1, \ldots, y_m) \in A$, $f_i(x_1, \ldots, x_n) \in A$, $i = \overline{1,m}$, причем функции f_i могут зависеть несущественно от некоторых своих переменных, верно*

$$f_0(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)) \in A,$$

то множество A является замкнутым классом.

Д-во. Рассмотрим произвольную функцию $f \in [A]$. Она выражается некоторой формулой F над множеством A. Докажем индукцией по числу d вхождений в формулу F обозначений функций из $A \setminus I$, что $f \in A$.
1. Базис индукции: $d = 0$. Если $F = x_i$, то $f \in A$ по условию утверждения.
2. Индуктивный переход. Пусть любая функция, которая может быть вырождена формулой не более чем с d_0 вхождениями обозначений функции из $A \setminus I$, содержится в A. Рассмотрим функцию $f(x_1, \ldots, x_n) \in [A]$, которая выражается формулой F с $d_0 + 1$ вхождениями обозначений функций из $A \setminus I$. Тогда $F = F_0(F_1, \ldots, F_m)$, где $f_0 \in A \setminus I$, F_i — формулы не более чем с d_0 вхождениями функций из $A \setminus I$. По предположению индукции $f_{F_i} \in A \implies$

$$f(x_1, \ldots, x_n) = f_0(f_{F_1}(x_1, \ldots, x_n), \ldots, f_{F_m}(x_1, \ldots, x_n))..$$

Далее, по условию утверждения $f \in A$. □

Теорема. *Множество T_0 является замкнутым классом.*

Д-во. Применим лемму о замкнутом классе. Пусть $f_0(y_1, \ldots, y_m) \in T_0$, $f_i(x_1, \ldots, x_n) \in T_0$, $i = \overline{1,m}$. Рассмотрим функцию

$$f(x_1, \ldots, x_n) = f_0(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)).$$

Получаем:

$$f(0, \ldots, 0) = f_0(f_1(0, \ldots, 0), \ldots, f_m(0, \ldots, 0)) = f_0(0, \ldots, 0) = 0.$$

Значит, $f \in T_0$. □

Опр. *Функция $f(x_1, \ldots, x_n) \in P_2$ сохраняет 1, если $f(1, \ldots, 1) = 1$. Множество всех функций, сохраняющих 1, обозначим T_1.*

Теорема. *Множество T_1 является замкнутым классом.*

Д-во. Полностью аналогично доказательству предыдущего утверждения. □

Опр. *Функция $f(x_1, \ldots, x_n) \in P_2$ называется линейной, если она может быть представлена в виде:*

$$f(x_1, \ldots, x_n) = c_0 \oplus c_1 x_1 \oplus \cdots \oplus c_n x_n,$$

где коэффициенты $c_0, c_1, \ldots, c_n \in E_2$. Множество всех линейных функций обозначим L.

Теорема. *Множество L является замкнутым классом.*

Д-во. Достаточно заметить, что при подстановке вместо переменных линейной функции каких-то других линейных функций не могут появиться конъюнкции переменных в слагаемых. □

2.6 Функции алгебры логики. Самодвойственные функции. Лемма о несамодвойственной функции.

Опр. *Функция $f(x_1, \ldots, x_n) \in P_2$ называется самодвойственной, если $f^*(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$. Заметим, что данное определение эквивалентно тому, что функция f на всех парах противоположных наборов принимает противоположные значения, т.е. $\forall \alpha \in E_2^n : f(\overline{\alpha}) = \overline{f(\alpha)}$. Множество всех самодвойственных функций обозначим S.*

Лемма. *Если $f \notin S$, то, подставляя вместо ее переменных функции x, $\overline{x}$, можно получить функцию, равную константе.*

Д-во. Если $f(x_1, \ldots, x_n) \notin S$, то найдется такая пара противоположных наборов $\alpha, \overline{\alpha} \in E_2^n$, что $f(\alpha) = f(\overline{\alpha}) = c \in E_2$. Положим:

$$\varphi(x) = f(x \oplus \alpha_1, \ldots, x \oplus \alpha_n).$$

Получаем:

$$\varphi(0) = f(\alpha) = c,$$
$$\varphi(1) = f(\overline{\alpha}) = c.$$

Значит, $\varphi(x) = c$. □

2.7 Функции алгебры логики. Монотонные функции. Лемма о немонотонной функции.

Пусть $\alpha, \beta \in E_2^n$. Будем говорить, что $\alpha \leq \beta$, если $\alpha_i \leq \beta_i,\ i = \overline{1, n}$.

Опр. *Функция $f(x_1, \ldots, x_n) \in P_2$ называется монотонной, если для любых наборов $\alpha, \beta \in E_2^n$ из $\alpha \leq \beta$ следует $f(\alpha) \leq f(\beta)$. Множество всех монотонных функций обозначим M.*

Лемма. *Если $f \notin M$, то, подставляя вместо ее переменных функции 0, 1, x можно получить функцию $\overline{x}$.*

Д-во. Если $f(x_1, \ldots, x_n) \notin M$, то найдется такая пара наборов $\alpha, \beta \in E_2^n$, что $\alpha \leq \beta$, но $f(\alpha) > f(\beta)$. Значит, $f(\alpha) = 1$ и $f(\beta) = 0$. Не ограничивая общности, пусть $\alpha_i = 0,\ \beta_i = 1,\ i = \overline{1,k}$ и $\alpha_i = \beta_i,\ i = \overline{k+1,n}$. Положим:

$$\varphi(x) = f(\underbrace{x, \ldots, x}_{k}, \alpha_{k+1}, \ldots, \alpha_n)..$$

Получаем:

$$\varphi(0) = f(\underbrace{0, \ldots, 0}_{k}, \alpha_{k+1}, \ldots, \alpha_n) = f(\alpha) = 1,$$
$$\varphi(1) = f(\underbrace{1, \ldots, 1}_{k}, \alpha_{k+1}, \ldots, \alpha_n) = f(\beta) = 0.$$

Значит, $\varphi(x) = \overline{x}$. □

2.8 Функции алгебры логики. Базис. Теорема о числе функций в базисе в алгебре логики.

Опр. *Пусть $B \subseteq P_2$. Множество B называется базисом P_2, если*

1. $[B] = P_2$;
2. $\forall f \in B : [B \setminus \{f\}] \neq P_2$.

Теорема.

1. *Любой базис P_2 содержит не больше четырех функций.*
2. *Для любого числа $k,\ 1 \leq k \leq 4$, в P_2 найдется базис, содержащий ровно k функций.*

Д-во. **1.** Пусть $B \subseteq P_2$ — базис P_2. Тогда B — полная система. Значит, по теореме Поста в B найдутся следующие (не обязательно различные) функции: $f_0 \notin T_0,\ f_1 \notin T_1,\ f_l \notin L,\ f_s \notin S,\ f_m \notin M$. Система $\{f_0, f_1, f_l, f_s, f_m\}$ — полна, а B — неизбыточна, поэтому $B = \{f_0, f_1, f_l, f_s, f_m\}$. Значит, $|B| \leq 5$.
Рассмотрим функцию $f_0 \in B,\ f_0 \notin T_0$:

x_1	$\ldots$	x_n	f_0
0	$\ldots$	0	1
	$\ldots$		
1	$\ldots$	1	a

Теперь
1) если $a = 0$, то $f_0 \notin T_1, M$, а значит, $f_1 = f_m = f_0$ и $|B| \leq 3$;
2) если $a = 1$, то $f_0 \notin S$, а значит, $f_s = f_0$, и $|B| \leq 4$.
Следовательно, $|B| \leq 4$.
2. Для каждого числа $k,\ 1 \leq k \leq 4$, приведем примеры базисов B из k функций:

1. если $k = 1$, то, например, $B = \{x/y\}$ или $B = \{x \downarrow y\}$;

2. если $k = 2$, то, например, $B = \{\overline{x},\ x \cdot y\}$ или $B = \{\overline{x},\ x \vee y\}$;

3. если $k = 3$, то, например, $B = \{1,\ x \oplus y,\ x \cdot y\}$;

4. если $k = 4$, то, например, $B = \{0,\ 1,\ x \oplus y \oplus z,\ x \cdot y\}$.

□

2.9 Графы. Изоморфизм графов. Связность. Формула Эйлера для степеней вершин. Теорема о соотношении между числом вершин, ребер и компонент связности в графе.

Опр. *(Неориентированным) графом G называется пара (V, E), где V — непустое множество вершин; E — конечное множество ребер, примем каждому ребру $e \in E$ сопоставлена неупорядоченная пара вершин, т.е. $e = (v, w)$, где $v, w \in V$.*

Ребро $e = (v, v)$, где $v \in V$, называется петлей. Ребра $e_1 = (v, w)$ и $e_2 = (v, w)$, где $v, w \in V$ и $e_1 \neq e_2$, называются кратными ребрами. Граф, в котором допускаются и петли, и кратные ребра иногда называется псевдографом. Граф без петель, но, возможно, с кратными ребрами называется мультиграфом. Граф без петель и кратных ребер называется простым, или обыкновенным графом.

Опр. *Два графа без петель и кратных ребер $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называются изоморфными, если найдется взаимно однозначное отображение $\varphi : V_1 \to V_2$, сохраняющее ребра, т.е. для любых вершин $v, w \in V_1$ выполняется соотношение:*

$$(v, w) \in E_1 \Leftrightarrow (\varphi(v), \varphi(w)) \in E_2.$$

Опр. *Степенью $d_G(v)$ вершины $v \in V$ в графе $G = (V, E)$ называется число исходящих из нее ребер (причем петля вносит двойной вклад в степень вершины).*

Утверждение. *Пусть $G = (V, E)$ — граф без петель и кратных ребер. Тогда*

1. $\sum\limits_{v \in V} d_G(v) = 2 \cdot |E|$;

2. *в графе G число вершин, имеющих нечетную степень четно.*

Д-во. **1.** Рассмотрим сумму в левой части равенства. Т.к. любое ребро графа имеет ровно два конца, каждое ребро в этой сумме будет подсчитано ровно два раза. Получаем выражение в правой части равенства.
2. Свойство непосредственно следует из равенства п.1. □

Опр. *Граф $G = (V, E)$ называется связным, если для каждой пары вершин графа G найдется путь, соединяющий эти вершины (а значит, и простая цепь, соединяющая эти вершины). Максимальный (по включению) связный подграф графа G называется его компонентой связности.*

Теорема. *Пусть $G = (V, E)$ — граф без петель и кратных ребер с p вершинами, q ребер и s компонентами связности. Тогда*

1. $s \geq p - q$;

2. если в графе G *отсутствуют циклы, то* $s = p - q$.

Д-во. **1.** Рассмотрим переход от графа $G_i = (V, E_i)$ к графу $G_{i+1} + e$, где $E_i \subseteq E,\ e \in E \setminus E_i$. Пусть в графах G_i, G_{i+1} соответственно $s_i,\ s_{i+1}$ компонент связности. Тогда если ребро e соединяет вершины из одной компоненты связности графа G_i, то $s_{i+1} = s_i$; и если ребро e соединяет вершины из разных компонент связности графа G_i, то $s_{i+1} = s_i - 1$. Поэтому $s_{i+1} \geq s_i - 1$. Граф G можно получить из графа $G_0 = (V, \emptyset)$ с p компонентами связности добавлением всех ребер множества E. Поэтому $s \geq p - q$.
2. Если же в графе G нет циклов, то в предыдущих рассуждениях верно $s_{i+1} = s_i - 1$. Поэтому $s = p - q$. □

2.10 Деревья. Корневые деревья, упорядоченные корневые деревья. Верхняя оценка числа деревьев с заданным числом ребер.

Опр. *Деревом называется связный граф без циклов.*

Опр. *Корневым деревом называется пара* $(D; v_0)$, *где* $D = (V, E)$ — *дерево,* $v_0 \in V$ — *выделенная вершина, называемая корнем. При изоморфизме корневых деревьев корень обязан переходить в корень. Висячая вершина корневого дерева, не являющаяся корнем, называется листом.*

Опр. *Пусть* $(D; v_0)$ — *корневое дерево и* $D_1, \ldots, D_m$ — *все его поддеревья. Корневое дерево* D *называется упорядоченным, если задан порядок его поддеревьев, а каждое его поддерево* $D_i, i = \overline{1, m}$, *также является упорядоченным корневым деревом.*

Теорема. *Для числа* $\delta''(q)$ *неизоморфных упорядоченных корневых деревьев с* q *ребрами справедлива оценка:* $\delta''(q) \leq 4^q$.

Д-во. Пусть (D, v_0) — упорядоченное корневое дерево с q ребрами. Обойдем дерево D в глубину из вершины $v_0 \in V$ по порядку его поддеревьев. При таком обходе по каждому ребру пройдем два раза: первый раз при обходе в соответствующее поддерево, второй раз при возвращении из него.
По этому обходу построим код дерева D — набор $k(D)$ из нулей и единиц длины $2q$. Сначала этот код не заполнен. При проходе по очередному ребру заполняем в коде $k(D)$ первый незаполненный разряд по следующим правилам:
1) если по ребру переходим в поддерево, то в код $k(D)$ пишем ноль;
2) если по ребру возвращаемся из поддерева, то в код $k(D)$ пишем единицу.
Тогда различным упорядоченным корневым деревьям соответствуют разные коды. Поэтому $\delta''(q)$ не превосходит числа наборов из нулей и единиц длины $2q$, т.е.

$$\delta''(q) \leq 2^{2q} = 4^q.$$

□

2.11 Геометрическое представление графов. Теорема о геометрическом представлении графов в трехмерном пространстве.

Опр. *Геометрическим представлением графа $G = (V, E)$ в пространстве $\mathbb{R}^n$ называется такое его отображение в $\mathbb{R}^n$, при котором:*

1. *каждой вершине $v \in V$ сопоставлена точка в $\mathbb{R}^n$, причем разным вершинам — разные точки;*
2. *каждому ребру $(v, w) \in E$ сопоставлена непрерывная кривая, соединяющая точки, соответствующие вершинам v и w, и не проходящая через точки, соответствующие другим вершинам;*
3. *кроме того, кривые, соответствующие различным ребрам, не пересекаются за исключением своих концов.*

Теорема. *Любой граф G допускает геометрическое представление в $\mathbb{R}^3$.*

Д-во. Пусть $G = (V, E)$, где $V = \{v_1, \ldots, v_p\}, E = \{e_1, \ldots, e_q\}$. Возьмем в $\mathbb{R}^3$ произвольную прямую l и отметим на ней p различных точек, которые обозначим $v_1, \ldots, v_p$. Сопоставим их вершинам графа G. Возьмем q различных плоскостей $\pi_1, \ldots, \pi_q$, содержащих прямую l. Ребру $e_i = (v_{i_1}, v_{i_2})$ графа G сопоставим кривую, соединяющую точки v_{i_1} и v_{i_2}, которую проведем в плоскости π_i, $i = 1, \ldots, q$. По построению кривые, сопоставленные ребрам, могут пересекаться только в концевых точках. Значит, получили геометрическое представление G в $\mathbb{R}^3$. □

2.12 Планарные графы. Формула Эйлера для планарных графов. Верхняя оценка числа ребер в планарном графе.

Опр. *Граф G называется планарным, если найдется его геометрическое представление на плоскости (т.е. в $\mathbb{R}^2$). В обратном случае граф G называется непланарным. Геометрическое представление планарного графа в R^2 назовем его укладкой на плоскости. Связные области плоскости, ограниченные ребрами планарного графа при его укладке на плоскости, называются гранями, неограниченная область называется также внешней гранью.*

Теорема (формула Эйлера для планарных графов)**.** *Если $G = (V, E)$ — связный планарный граф с p вершинами и q ребрами, то для каждой его укладки на плоскости верно равенство: $p - q + r = 2$, где r — число граней в этой укладке.*

Д-во. Проведем индукцию по q при заданном p.
Базис индукции. Если $q = p - 1$, то G — дерево. Каждое дерево — планарный граф с одной гранью. Поэтому формула верна.
Индуктивный переход. Рассмотрим связный планарный граф G с p вершинами и $q \geq p$ ребрами. Пусть задана его укладка на плоскости, в которой r граней.
В графе G найдется хотя бы один цикл, и пусть e — любое ребро какого-то его цикла. Тогда граф $G' = G - e$ — связный и планарный с p вершинами и $q - 1$ ребрами, и его укдладка на плоскости содержит $r - 1$ грань, т.к. при удалении ребра e из укладки графа G две грани соединяются в одну. Для графа G' верно предположение индукции, т.е. $p - (q - 1) + (r - 1) = 2 \implies p - q + r = 2$. □

Теорема. *Наибольшее число ребер в планарном графе (без петель и кратных ребер) с $p \geq 3$ вершинами равно $3p - 6$.*

Д-во. Можно рассматривать связные графы.
1. *Верхняя граница.* Пусть $G = (V, E)$ — связный планарный граф с p вершинами и q ребрами. Рассмотрим укладку графа G на плоскость, и пусть q_i — число ребер, встречающихся при обходе границы i-й грани в этой кладке, $i = \overline{1, r}$. Тогда $\sum\limits_{i=1}^{r} q_i = 2q$, т.к. каждое ребро:
1) либо разделяет две грани, а значит, считается при обходе границ этих двух граней;
2) либо лежит в одной грани, а значит, при обходе ее границы считается два раза.
Из связности графа и $p \geq 3$ получаем $q_i \geq 3$, откуда $3r \leq 2q$, или $r \leq \frac{2}{3}q$. По формуле Эйлера $r = q - p + 2$, поэтому $q - p + 2 \leq \frac{2}{3}q$, а значит

$$q \leq 3p - 6.$$

2. *Достижимость верхней оценки.* Построим графы, на которых достигается эта оценка. Это связные планарные графы, в которых любая грань (включая внешнюю) ограничена циклом длины три. Такие графы называются триангуляциями.
Если $p = 3$, то $G_p = K_3$. Пусть уже построен связный планарный граф G_p с p вершинами и $3p-6$ ребрами, каждая грань которого ограничена треугольником. Тогда граф G_{p+1} получается из G_p добавлением новой вершины внутри какой-то грани и ребер, соединяющих эту вершину с тремя вершинами границы этой грани. □

2.13 Графы K5 и K3,3. Непланарность графов K5 и K3,3. Теорема Понтрягина-Куратовского (доказательство в одну сторону).

Теорема. *Граф K_5 не является планарным.*

Д-во. От противного. Пусть граф K_5 планарен. Тогда для произвольной укладки на плоскость верно равенство: $p - q + r = 2$, где $p = 5$ — число вершин и $q = 10$ число ребер в графе, а r — число граней в этой укладке. Поэтому $r = 7$.
Пусть q_i — число ребер, встречающихся при обходе границы i-й грани в этой укладке. Тогда $\sum\limits_{i=1}^{r} q_i = 2q$. Но $q_i \geq 3$, поэтому $3r \leq 2q$, или $r \leq \frac{2}{3}q$. Получаем: $7 \leq \frac{2}{3} \cdot 10$ — противоречие. Значит, граф K_5 не является планарным. □

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

Д-во. От противного. Пусть граф $K_{3,3}$ планарен. Тогда для произвольной его укладки на плоскости верно равенство: $p-q+r = 2$, где $p = 6$ — число вершин и $q = 9$ число ребер в графе, а r — число граней в этой укладке. Поэтому $r = 5$. Пусть q_i — число ребер, встречающихся при обходе границы i-й грани в этой укладке. Тогда $\sum\limits_{i=1}^{r} q_i = 2q$. Но $q_i \geq 4$, поэтому $4r \leq 2q$, или $r \leq \frac{1}{2}q$. Получаем: $5 \leq \frac{9}{2}$ — противоречие. Значит, граф $K_{3,3}$ не является планарным. □

Опр. *Говорят, что граф $G' = (V', E')$ получен из графа $G = (V, E)$ подразбиением ребра $e = (v, w) \in E$, если*

$$V' = V \cup \{u\}, \text{ где } u \notin V;$$
$$E' = E \setminus \{(v, w)\} \cup \{(v, u), (u, w)\}.$$

Граф G' называется подразбиением графа G, если G' может быть получен из G конечным числом подразбиений ребер.

Опр. *Графы $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называются гомеоморфными, если найдутся их изоморфные подразбиения G_1' и G_2' соответственно.*

Теорема. *Граф $G = (V, E)$ планарен тогда и только тогда, когда в нем не найдется ни одного подграфа, гомеоморфного либо графу K_5, либо графу $K_{3,3}$.*

Д-во. ($\Longrightarrow$) От противного. Пусть граф G планарен, но в нем есть подграф, гомеоморфный, например, K_5. Тогда этот подграф не планарен, а значит не планарен и сам граф. Противоречие. Значит в G нет подргафа гомеоморфного K_5. □

2.14 Раскраски вершин графов. Теорема о раскраске вершин графа в 2 цвета (теорема Кенига).

Теорема (Кенига)**.** *Вершины графа G можно раскрасить в два цвета тогда и только тогда, когда в нем не найдется ни одного простого цикла нечетной длины.*

Д-во. **1.** Если в графе G найдется простой цикл нечетной длины, то вершины этого цикла в два цвета не раскрасить.
2. Пусть теперь в графе G отсутствуют простые циклы нечетной длины. Можно считать, что G — связный граф, иначе проведем рассуждения для каждой его компоненты связности.
Построим в графе G его остовное дерево D. Выберем произвольную вершину $v_0 \in V$. В дереве D для пары вершин v_0, w, где $w \in V$, существует ровно одна простая (v_0, w)-цепь P_w. Рассмотрим отображение $\rho : V \to \{1, 2\}$: $\rho(w) = 1$, если длина цепи P_w нечетна; $\rho(w) = 2$, если длина цепи P_w четна. Покажем, что ρ является раскраской вершин, т.е. в графе G нет ребер, оба конца которых окрашены в один и тот же цвет.
Предположим обратное: пусть $(u, w) \in E$ и $\rho(u) = \rho(w)$. Рассмотрит в графе G замкнутый путь $P = v_0 P_u u(u, w) w P_w v_0$. Длина цепи P нечетна, т.к. у длин цепей P_u, P_w в дереве D одинаковая четность. Но из указанного замкнутого пути P можно выделить простой цикл нечетной длины — противоречие. Значит, ρ — раскраска вершин графа G в два цвета. □

2.15 Коды с минимальной избыточностью (оптимальные коды). Три леммы о свойствах кодов с минимальной избыточностью.

Лемма. *Пусть заданы A, $|A| = r$, B и $P = (p_1, \ldots, p_r)$, причем $p_i > p_j$. Если $C_\varphi = \{B_1, \ldots, B_r\}$ — оптимальный код, то $|B_i| \le |B_j|$.*

Д-во. Пусть $l_1, \ldots, l_r$ — длины кодовых слов $B_1, \ldots, B_r$ и, для определенности, $i < j$. Докажем от обратного: предположим, что $l_i > l_j$.
Рассмотрим код $C_{\varphi'}$, где

$$C_{\varphi'} = \{B_1, \ldots, B_{i-1}, B_j, B_{i+1}, \ldots, B_{j-1}, B_i, B_{j+1}, \ldots, B_r\}.$$

Код $C_{\varphi'}$ получен из однозначного кода C_φ перестановкой кодовых слов B_i и B_j. Значит, код $C_{\varphi'}$ — также однозначен. Получаем:

$$c(\varphi') - c(\varphi) = p_i(l_j - l_i) + p_j(l_i - l_j) = (p_i - p_j)(l_j - l_i) < 0,$$

Значит, $c(\varphi') < c(\varphi)$, чего не может быть, т.к. C_φ — оптимальный код. Следовательно, $l_i < l_j$. □

Лемма. *Пусть заданы* $A,\ |A| = r \geq 2,\ B = \{0,1\}$ *и* $P = (p_1, \ldots, p_r)$. *Если* $C_\varphi = \{B_1, \ldots, B_r\}$ *— оптимальный префиксный код и* B_i *— кодовое слово с наибольшей длиной, причем* $B_i = B_i'b$, *где* $B_i' \in B^*,\ b \in B$, *то в коде* C_φ *найдется кодовое слово* $B_j = B_i'\overline{b}$.

Д-во. Если $r = 2$, то лемма верна. Пусть $r \geq 3$ и $l_1, \ldots, l_r$ — длины кодовых слов $B_1, \ldots, B_r$. Отметим, что $l_i \geq 2$. Докажем от противного: предположим, что слово $B_i'\overline{b}$ в коде C_φ не встречается.
Рассмотрим код $C_{\varphi'}$, где

$$C_{\varphi'} = \{B_1, \ldots, B_{i-1}, B_i', B_{i+1}, \ldots, B_r\}.$$

Код $C_{\varphi'}$ получен из префиксного кода $C_{\varphi'}$ удалением последней буквы из самого длинного кодового слова B_i. Значит, код $C_{\varphi'}$ также является префиксным. Получаем:

$$c(\varphi') - c(\varphi) = p_i(l_i - 1) - p_i l_i = -p_i < 0.$$

Значит, $c(\varphi') < c(\varphi)$, чего не может быть, т.к. C_φ — оптимальный код. Следовательно, в коде C_φ найдется кодовое слово $B_i'\overline{b}$. □

Лемма. *Пусть заданы* $A,\ |A| = r,\ r \geq 2,\ B = \{0,1\}$ *и* $P = (p_1, \ldots, p_r)$, *причем* $p_1 \geq p_2 \geq \cdots \geq p_{r-1} \geq p_r$. *Тогда найдется такой оптимальный префиксный код, что кодовые слова, сопоставленные буквам с частотами* p_{r-1} *и* p_r *являются самыми длинными и отличаются только последней буквой.*

Д-во. Пусть $C_\varphi = \{B_1, \ldots, B_r\}$ — какой-то оптимальный префиксный код и $l_1, \ldots, l_r$ — длины кодовых слов $B_1, \ldots, B_r$. Пусть B_i — кодовое слово с наибольшей длиной в коде C_φ, $B_i = B_i'b$, где $B_i' \in B^*,\ b \in B$. Тогда в коде C_φ найдется кодовое слово $B_j = B_i'\overline{b}$. Пусть для определенности $i < j$.
Рассмотрим код $C_{\varphi'}$, где

$$C_{\varphi'} = \{B_1, \ldots, B_{i-1}, B_{r-1}, B_{i+1}, \ldots, B_r, B_{j+1}, \ldots, B_{r-2}, B_i, B_j\}.$$

Код $C_{\varphi'}$ получен из префиксного кода C_φ перестановкой кодовых слов. Значит, код $C_{\varphi'}$ также является префиксным. Получаем:

$$\begin{aligned} c(\varphi') - c(\varphi) &= p_i(l_{r-1} - l_i) + p_j(l_r - l_j) + p_{r-1}(l_i - l_{r-1}) + p_r(l_j - l_r) = \\ &= (p_i - p_{r-1})(l_{r-1} - l_i) + (p_j - p_r)(l_r - l_j). \end{aligned}$$

Теперь если $p_i = p_{r-1}$, то $(p_i - p_{r-1})(l_{r-1} - l_i) = 0$.
Если же $p_i > p_{r-1}$, то верно $l_i \leq l_{r-1}$. Но l_i — наибольшая длина среди всех кодовых слов, поэтому $l_i = l_{r-1}$, откуда $(p_i - p_{r-1})(l_{r-1} - l_i) = 0$.
Аналогично устанавливаем, что $(p_j - p_r)(l_r - l_j) = 0$. Значит $c(\varphi') = c(\varphi)$. Но C_φ — оптимальный код, поэтому $C_{\varphi'}$ — также оптимальный код. Код $C_{\varphi'}$ — искомый. □

2.16 Коды с минимальной избыточностью (оптимальные коды). Алгоритм Хаффмена построения кода с минимальной избыточностью.

По теореме редукции задачу поиска оптимального кода можно свести к такой же задаче, но с исходным алфавитом с числом букв, меньшим на единицу, и с набором частот, получающимся из первоначального сложением двух наименьших частот. Так можно уменьшать число букв в исходных алфавитах до тех пор, пока не получим алфавит из двух букв. А для исходного алфавита из двух букв при любом наборе частот в кодирующем алфавите $B = \{0, 1\}$ оптимальным является код $C_\varphi = \{0, 1\}$.

Алгоритм построения оптимального кода в кодирующем алфавите $B = \{0, 1\}$.

Вход: набор частот $P = (p_1, \ldots, p_r),\ p_i \in \mathbb{R}_+,\ \sum\limits_{i=1}^{r} p_i = 1,\ r \geq 2$.

Выход: дерево D_{φ^*} какого-то оптимального префиксного кода $C_{\varphi^*} = \{B_1, \ldots, B_r\}$ для набора частот P.

Описание алгоритма.

1. Положить: $H_1 = (V_1, E_1)$, где $V_1 = \{u_1, \ldots, u_r\},\ E_1 = \emptyset$, и $p(u_i) = p_i$ для всех $i = 1, \ldots, r,\ W_1 = V_1$.

2. Для всех $k = 1, \ldots, r-1$: выбрать в множестве W_k две такие вершины w' и w'', что

$$p(w') \leq p(w),\ p(w'') \leq p(w)$$

для любой вершины $w \in W_k,\ w \neq w',\ w \neq w''$, положить: $H_{k+1} = (V_{k+1}, E_{k+1})$, где $V_{k+1} = V_k \cup \{v_k\},\ E_{k+1} = E_k \cup \{(v_k, w'), (v_k, w'')\}$, и

$$p(v_k) = p(w') + p(w''), \quad W_{k+1} = (W_k \cup \{v_k\}) \setminus \{w', w''\},$$

ребру (v_k, w') приписать 0, ребру (v_k, w'') приписать 1.

3. Положить: $D_{\varphi^*} = H_r$ с корнем v_{r-1}.

2.17 Коды, исправляющие одну ошибку. Алгоритмы кодирования, исправления ошибки и декодирования в коде Хэмминга.

При рассмотрении кодов Хэмминга обычно в словах разряды с номерами, являющимися степенями двойки, называют проверочными, а остальные разряды — информационными.

Пусть $n \geq 3,\ 2^{k-1} < n < 2^k$ и $m = n - k$. Если H — код Хэмминга порядка n, то H содержит 2^m слов и исправляет одну ошибку. Поэтому найдется такое разделимое кодирование $\varphi_H : A^m \to B^n$, что $C_{\varphi_H} = H$.

Опишем алгоритм кодирование в коде Хэмминга.

Вход: слово $\alpha \in A^m$, где $m = n - k,\ n \geq 3,\ 2^{k-1} < n < 2^k$.

Выход: слово $\beta = \varphi_H(\alpha) \in H$, где $\beta \in B^n$.

Описание алгоритма.

1. Заполнение информационных разрядов. Для всех $j \in N_n \setminus \{2^0, 2^1, \ldots, 2^{k-1}\}$ положить:

$$\beta_j = \alpha_{j - \lceil \log_2 j \rceil}.$$

2. Заполнение проверочных разрядов. Для всех $i = \overline{0, k-1}$ положить:

$$\beta_{2^i} = \bigoplus_{j \in D_i, j \neq 2^i} \beta_j.$$

Опишем алгоритм исправления ошибки к коде Хэмминга.
Вход: слово $\beta' \in B^n$, полученное из некоторого слова $\beta \in H$, в котором могла произойти одна ошибка замещения, где $n \geq 3,\ 2^{k-1} < n < 2^k$.
Выход: слово $\beta \in H$, где $\beta \in B^n$.
Описание алгоритма.
1. Вычисление проверочных сумм. Для всех $i = \overline{0, k-1}$ найти:

$$s_i = \bigoplus_{j \in D_i} \beta'_j,$$

затем положить: $s = \sum\limits_{i=0}^{k-1} s_i \cdot 2^i$.
2. Исправление ошибки. Если $s = 0$, то ошибки нет, положить: $\beta = \beta'$.
Если $s \neq 0$, то ошибка в s-м разряде, положить:

$$\beta_j = \beta'_j \text{ при } j = \overline{1, n}, j \neq s \ \text{ и } \ \beta_s = \overline{\beta}'_s.$$

Опишем алгоритм декодирования в коде Хэмминга.
Вход: слово $\beta \in H$, где $\beta \in B^n,\ n \geq 3,\ 2^{k-1} < n < 2^k$.
Выход: слово $\alpha \in A^m$, где $\beta = \varphi_H(\alpha),\ m = n - k$.
Описание алгоритма.
Вычеркивание проверочных рядов. Вычеркнуть в слове β разряды β_j для всех $j = 2^0, 2^1, \ldots, 2^{k-1}$, затем оставшееся слово обозначить α.

2.18 Линейные двоичные коды. Теорема о кодовом расстоянии линейных кодов.

Пусть $n \geq 1$. Определим для наборов из B^n операции сложения и умножения на число из B. Если $\beta, \gamma \in B^n$, то

$$\beta \oplus \gamma = (\beta_1 \oplus \gamma_1, \ldots, \beta_n \oplus \gamma_n) \in B^n.$$

Если $\beta \in B^n$ и $c \in B$, то

$$c\beta = (c \cdot \beta_1, \ldots, c \cdot \beta_n) \in B^n.$$

Наборы $\beta_1, \ldots, \beta_k \in B^n$ называются линейно независимыми, если из равенства

$$c_1\beta_1 \oplus \cdots \oplus c_k\beta_k = (0, \ldots, 0)$$

следует

$$c_1 = \cdots = c_k = 0.$$

В обратном случае наборы $\beta_1, \ldots, \beta_k \in B^n$ называются линейно зависимыми.
Множество $V \subseteq B^n$ называется линейным пространством если из $\beta, \gamma \in V$ следует $\beta \oplus \gamma \in V$.
Отметим, что если $V \subseteq B^n$ — линейное пространство, то для любого $k \geq 1$ для любых наборов $\beta_1, \ldots, \beta_k \in V$ и для любых $c_1, \ldots, c_k \in B$ верно

$$c_1\beta_1 \oplus \cdots \oplus c_k\beta_k \in V.$$

Если $V \subseteq B^n$ — линейное пространство, то наибольшее множество линейно независимых наборов из V называется его базисом. Известно, что любой базис V содержит одно и то же число наборов, называемое размерностью пространства V. Если $\beta_1, \ldots, \beta_k$ — базис V, то $|V| = 2^k$ и для любого набора $\beta \in V$ найдется однозначно определенное представление:

$$\beta = c_1\beta_1 \oplus \cdots \oplus c_k\beta_k,$$

где $c_1, \ldots, c_k \in B$.

Опр. *Пусть $n \geq 1$ и $C \subseteq B^n$ — равномерный код. Код C называется линейным, если множество C является линейным пространством.*

Весом $|\beta|$ набора $\beta \in B^n$ называется число его разрядов, равных единице.

Теорема. *Если $C \subseteq B^n$ — линейный код, $n \geq 1$, то для его кодового расстояния верно равенство:*

$$d_C = \min_{\beta \in C, \beta \neq (0,\ldots,0)} |\beta|.$$

Д-во. **1.** Сначала установим, что в C найдется набор, вес которого совпадает с d_C. По определению

$$d_C = \min_{\beta_1, \beta_2 \in C, \beta_1 \neq \beta_2} \rho(\beta_1, \beta_2).$$

Пусть кодовое расстояние достигается на паре наборов $\gamma_1, \gamma_2 \in C$, т.е. $\gamma_1 \neq \gamma_2$ и $d_C = \rho(\gamma_1, \gamma_2)$. Но C — линейный код, поэтому набор $\gamma = \gamma_1 \oplus \gamma_2$ также принадлежит коду C. Кроме того, вес $|\gamma|$ набора γ равен числу разрядов, в которых набор γ_1 и γ_2 различаются. Значит,

$$|\gamma| = \rho(\gamma_1, \gamma_2) = d_C.$$

2. Теперь покажем от обратного, что в C не найдутся ненулевые наборы, вес которых меньше d_C. Предположим, что для некоторого набора $\gamma' \in C$ верно $|\gamma'| < d_C$. Но C — линейный код, поэтому нулевой набор $(0, \ldots, 0) \in B^n$ также принадлежит коду C. Получаем противоречие:

$$d_C \leq \rho(\gamma', (0, \ldots, 0)) = |\gamma'| < d_C.$$

Значит, ненулевой набор с весом, меньшим d_C, в линейном коде C не найдется. □

2.19 Конечные автоматы. Функционирование конечного автомата. Автоматные функции. Канонические уравнения и диаграмма Мура конечного автомата. Единичная задержка, ее автоматность.

Опр. *Функционирование конечного автомата* $\mathcal{A} = (A, B, Q, \varphi, \psi, q_*)$ *на входном слове* $x = x(1)x(2)\dots x(m) \in A^*$ *описывается системой канонических уравнений:*

$$\begin{cases} y(t) = \varphi(x(t), q(t-1)), 1 \le t \le m, \\ q(t) = \psi(x(t), q(t-1)), 1 \le t \le m, \\ q(0) = q_*. \end{cases}$$

При этом говорят, что конечный автомат $\mathcal{A}$ *входное слово* $x = x(1)x(2)\dots x(m) \in A^*$ *преобразует в выходное слово* $y = y(1)y(2)\dots y(m) \in B^*$.

Пусть A — конечный алфавит. Бесконечным словом (или сверхсловом) в алфавите A назовем бесконечную последовательность букв этого алфавита. Множество всех сверхслов в алфавите A обозначим A^∞.

Опр. *Конечный автомат* $\mathcal{A}$ *каждое сверхслово* $x \in A^\infty$ *преобразует в однозначно определенное сверхслово* $y \in B^\infty$. *Значит, конечный автомат* $\mathcal{A}$ *определяет некоторую функцию*

$$f_{\mathcal{A}} : A^\infty \to B^\infty,$$

которую назовем отображением, осуществляемое автоматом $\mathcal{A}$.

Опр. *Пусть* A *и* B *— конечный алфавиты и* $f : A^\infty \to B^\infty$. *Функция* f *называется автоматной, если найдется такой конечный автомат* $\mathcal{A} = (A, B, Q, \varphi, \psi, q_*)$, *что* $f_{\mathcal{A}} = f$.

Рассмотрим способы представления конечных автоматов и соответствующих автоматных функций.

1. *Канонические уравнения.*
Конечный автомат $\mathcal{A} = (A, B, Q, \varphi, \psi, q_*)$ (и автоматную функцию $f_{\mathcal{A}}$) можно задавать каноническим уравнениями:

$$\begin{cases} y(t) = \varphi(x(t), q(t-1)), \\ q(t) = \psi(x(t), q(t-1)), \\ q(0) = q_*. \end{cases}$$

При этом часто удобно, чтобы в правых частях находились функции алгебры логики. Для этого элементы множеств A, B, Q кодируют однозначным алфавитным равномерным кодом в алфавите $\{0, 1\}$. А затем переписывают функции $\varphi(t),\ \psi(t)$ в соответствии с этим кодированием.

2. *Диаграмма мура.*
Диаграммой Мура (или диаграммой переходов) конечного автомата $\mathcal{A} = (A, B, Q, \varphi, \psi, q_*)$ (и автоматной функции $f_{\mathcal{A}}$) называется ориентированный граф с пометками $D_{\mathcal{A}} = (V_{\mathcal{A}}, E_{\mathcal{A}})$, в котором: $V_{\mathcal{A}} = Q$; $E_{\mathcal{A}} = \{(q, \psi(a, q)) : a \in A, q \in Q\}$, причем дуге $(q, \psi(a, q)) \in E_{\mathcal{A}}$ приписана пометка $a(\varphi(a, q))$; вершина $q_* \in V_{\mathcal{A}}$ помечена звездочкой $*$.

Опр. *Пусть* $A = B = \{0,1\}$ *и* $z : A^\infty \to B^\infty$, *где*

$$z(x(1)x(2)\dots x(t)\dots) = 0x(1)x(2)\dots x(t-1)\dots.$$

Она называется функцией единичной задержки.

Утверждение. *Функция единичной задержки является автоматной.*

Д-во. Отображение z осуществляется конечным автоматом $\mathcal{A} = (A, B, Q = \{0,1\}, \varphi, \psi, q_* = 0)$, где состояние $q = 0$ означает "в предыдущий момент времени на входе был 0"; состояние $q = 1$ означает "в предыдущий момент времени на входе был 1". Найдем таблицы функций $\varphi,\ \psi$ и канонические уравнения:

$q \in Q$	$a \in A$	φ	ψ
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

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

□

2.20 Схемы из функциональных элементов. Выразимость функции алгебры логики схемой из функциональных элементов в базисе из конъюнкции, дизъюнкции и отрицания.

Рассмотрим СФЭ $S(x_1, \dots, x_n; y_1, \dots, y_m)$ в базисе $B_0 = \{x \cdot y,\ x \vee y,\ \overline{x}\}$. Пусть она построена по орграфу $G = (V, E)$. Тогда в любой ее вершине $v \in V$ вычисляется некоторая функция $f_v(x_1, \dots, x_n) \in P_2$, по индукции однозначно определяемая по СФЭ.

Базис индукции. Если v — входная вершина СФЭ и ей приписана входная переменная x_i, то

$$f_v = x_i,$$

т.е. в вершине v вычисляется функция, тождественно равная переменной x_i.
Индуктивный переход. **1.** Если v —внутренняя вершина СФЭ и ей приписано отрицание $\overline{x}$, причем $(w, v) \in E$, то

$$f_v = \overline{f}_w,$$

т.е. в вершине v вычисляется функция, равная отрицанию той функции, которая вычисляется в вершине w, из которой ведет дуга в вершину v.
2. Если v — внутренняя вершина СФЭ и ей приписана конъюнкция & (дизъюнкция $\vee$), причем $(w_1, v) \in E,\ (w_2, v) \in E$, где $w_1 \neq w_2$, то

$$f_v = f_{w_1} \cdot f_{w_2} \quad (f_v = f_{w_1} \vee f_{w_2}),$$

т.е. в вершине v вычисляется функция, равная конъюнкции (дизъюнкции) тех функций, которые вычисляются в вершинах w_1 и w_2, из которых ведут дуги в вершину v.

Любую ли функцию $f \in P_2$ можно вычислить некоторой СФЭ в базисе $B_0 = \{x \cdot y,\ x \vee y,\ \overline{x}\}$? Да, т.к. множество

$$B_0 = \{x \cdot y,\ x \vee y,\ \overline{x}\} \subseteq P_2$$

является полной системой.
Значит, функцию f можно записать некоторой формулой над множеством B_0. А затем по этой формуле построить соответствующую СФЭ, вычисляющую функцию f.