МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2012 №1(15)

Свидетельство о регистрации: ПИ №ФС 77-33762
от 16 октября 2008 г.

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Адрес редакции: 634050, г. Томск, пр. Ленина, 36
E-mail: vestnik_pdm@mail.tsu.ru

В журнале публикуются результаты фундаментальных и прикладных научных исследований отечественных и зарубежных ученых, включая студентов и аспирантов, в области дискретной математики и её приложений в криптографии, компьютерной безопасности, кибернетике, информатике, программировании, теории надежности, интеллектуальных системах.

Периодичность выхода журнала: 4 номера в год.

Редактор *Н. И. Шидловская*
Верстка *И. А. Панкратовой*

Подписано к печати 16.03.2012.
Формат $60 \times 84\frac{1}{8}$. Усл. п. л. 14,2. Уч.-изд. л. 15,9. Тираж 300 экз.

Издательство ТГУ. 634029, Томск, ул. Никитина, 4
Отпечатано в типографии ТПУ.

СОДЕРЖАНИЕ

CONTENTS

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

DOI 10.17223/20710410/15/1

УДК 631.391:519.2

О СТАТИСТИЧЕСКИХ СВОЙСТВАХ НЕЛИНЕЙНОСТИ СУЖЕНИЙ БУЛЕВЫХ ФУНКЦИЙ НА СЛУЧАЙНО ВЫБРАННОЕ ПОДПРОСТРАНСТВО

А. Н. Алексейчук, С. Н. Конюшок

Институт специальной связи и защиты информации Национального технического университета Украины «Киевский политехнический институт», г. Киев, Украина

E-mail: alex-crypto@mail.ru, 3tooth@mail.ru

Показано, что для всех достаточно больших натуральных n относительная нелинейность произвольной булевой функции n переменных может быть статистически аппроксимирована относительной нелинейностью ее сужения на случайное подпространство (возможно, с выколотым нулевым вектором), размерность которого не зависит от n.

Ключевые слова: *булева функция, нелинейность, случайное подпространство, статистическая оценка.*

1. Постановка задачи и основной результат

Пусть $V_n = \{0,1\}^n$, $f : V_n \to \{0,1\}$ — булева функция n переменных, $\hat{f}(\alpha) = 2^{-n} \sum\limits_{x \in V_n} (-1)^{f(x) \oplus \alpha x}$, $\alpha \in V_n$, — ее нормированные коэффициенты Уолша — Адамара. Напомним (см., например, [1, с. 233]), что нелинейность N_f функции f определяется как расстояние от f до множества аффинных булевых функций n переменных и удовлетворяет следующему равенству:

$$N_f = 2^{n-1}(1 - \hat{f}_*),$$

где

$$\hat{f}_* = \max_{\alpha \in V_n} |\hat{f}(\alpha)|. \tag{1}$$

Назовем число $\tilde{N}_f = N_f / 2^{n-1}$ *относительной нелинейностью функции* f.

Пусть далее X — двоичная матрица размера $t \times n$, где $t < n$. Обозначим $\dot{f}_X(u) = f(uX)$, $u \in V_t^* = V_t \backslash \{0\}$, частичную булеву функцию, равную, с точностью до замены переменных $u \mapsto uX$, $u \in V_t^*$, сужению функции f на подпространство, порожденное строками матрицы X (возможно, с выколотым нулевым вектором). Положим

$$\eta_a(X) = \frac{1}{2^t - 1} \sum_{u \in V_t^*} (-1)^{\dot{f}_X(u) \oplus ua}, \quad a \in V_t; \tag{2}$$

$$\eta_*(X) = \max_{a \in V_t} |\eta_a(X)|. \tag{3}$$

Из данных определений следует, что величина $N_{\dot{f}_X} = \dfrac{2^t-1}{2}(1-\eta_*(X))$ равна расстоянию от функции $\dot{f}_X$ до множества аффинных функций t переменных, ограниченных на подмножество V_t^*. Назовем число $\tilde{N}_{\dot{f}_X} = 1-\eta_*(X)$ *относительной нелинейностью функции* $\dot{f}_X$.

Предположим теперь, что матрица X выбирается случайно и равновероятно из множества $F_{t\times n}$ всех $t\times n$-матриц над полем $F=\mathrm{GF}(2)$. Тогда случайную величину $\tilde{N}_{\dot{f}_X}$ можно рассматривать как статистическую оценку параметра $\tilde{N}_f$, и естественно поставить вопрос о свойствах этой оценки. Отметим, что связь между нормированными коэффициентами Уолша — Адамара функций f и $\dot{f}_X$ исследуется в работах [2–4] в связи с построением вероятностных алгоритмов нахождения линейных аппроксимаций, а также проверки ряда свойств булевых функций. В частности, в [3] показано, что при случайном равновероятном выборе матрицы X из множества $F_{t\times n}$ для любых $\alpha\in V_n$, $\varepsilon\in(0,1)$ выполняется неравенство

$$\mathbf{P}\{|\hat{f}(\alpha)-\eta_{X\alpha}(X)|\geqslant\varepsilon\}\leqslant\frac{\varepsilon^{-2}}{2^t-1}. \tag{4}$$

Аналогичное неравенство применительно к более общему случаю получено в [5], однако в указанных, а также в других известных авторам публикациях не затрагивается вопрос о связи между величинами (1) и (3) или, что то же самое, между относительными нелинейностями функций f и $\dot{f}_X$.

Основным результатом настоящей работы является следующая теорема.

Теорема. Для любых $\varepsilon,\delta\in(0,1)$ существует натуральное число $t_0=t_0(\varepsilon,\delta)$, такое, что для любых натуральных $n>t\geqslant t_0$ и произвольной функции $f: V_n\to\{0,1\}$ справедливо неравенство

$$\mathbf{P}\{|\tilde{N}_f-\tilde{N}_{\dot{f}_X}|\geqslant\varepsilon\}\leqslant\delta, \tag{5}$$

где X — случайная равновероятная $t\times n$-матрица над полем F.

Доказательство теоремы базируется, в основном, на анализе моментов случайных величин (2) и излагается в п. 2. Возможность практического применения теоремы к оцениванию нелинейности булевых функций обсуждается в п. 3.

2. Доказательство теоремы

Зафиксируем числа $n,t\in\mathbb{N}$, где $n>t$, $\varepsilon\in(0,1)$, функцию $f: V_n\to\{0,1\}$ и оценим сверху вероятность события $\{|\tilde{N}_f-\tilde{N}_{\dot{f}_X}|\geqslant\varepsilon\}=\{|\hat{f}_*-\eta_*(X)|\geqslant\varepsilon\}$.

Докажем ряд вспомогательных утверждений.

Лемма 1. Справедливо неравенство

$$\mathbf{P}\{\hat{f}_*\geqslant\eta_*(X)\ +\ \varepsilon\}\leqslant\frac{\varepsilon^{-2}}{2^t-1}.$$

Доказательство. Обозначим α^* вектор из множества V_n, такой, что $\hat{f}_*=|\hat{f}(\alpha^*)|$. На основании формулы (3) событие $\{\hat{f}_*\geqslant\eta_*(X)+\ \varepsilon\}$ влечет событие $\{|\hat{f}(\alpha^*)|\geqslant\geqslant|\eta_{X\alpha^*}(X)|+\varepsilon\}$, которое, в свою очередь, влечет событие $\{|\hat{f}(\alpha^*)-\eta_{X\alpha^*}(X)|\geqslant\varepsilon\}$. Отсюда на основании неравенства (4) следует, что

$$\mathbf{P}\{\hat{f}_*\geqslant\eta_*(X)\ +\ \varepsilon\}\leqslant\mathbf{P}\{|\hat{f}(\alpha^*)-\ \eta_{X\alpha^*}(X)|\ \geqslant\ \varepsilon\}\leqslant\frac{\varepsilon^{-2}}{2^t-1}.$$

Лемма доказана. ■

Следующее простое утверждение, по-видимому, хорошо известно, однако авторам не удалось найти источник, содержащий нужную формулировку.

Лемма 2. Для числа $l(k,n)$ линейно зависимых систем, состоящих из k двоичных векторов длины n, справедливо неравенство $l(k,n) < 2^{nk} \cdot 2^{k-n}$.

Доказательство. Число линейно независимых систем из k двоичных векторов длины n равно

$$2^{nk} - l(k,n) = (2^n-1)(2^n-2)\cdots(2^n-2^{k-1}) = 2^{nk}(1-2^{-n})(1-2^{-(n-1)})\cdots(1-2^{-(n-(k-1))}).$$

Отсюда на основании неравенства $\prod\limits_{i=1}^{N}(1-x_i) \geqslant 1 - \sum\limits_{i=1}^{N} x_i$, $x_i \in (0,1)$, $i = 1,\ldots,N$, следует, что

$$2^{nk} - l(k,n) \geqslant 2^{nk}\left(1 - 2^{-(n-k+1)}(1 + 2^{-1} + \cdots + 2^{-(k-1)})\right) > \\ > 2^{nk}\left(1 - 2^{-(n-k+1)} \cdot 2\right) = 2^{nk} - 2^{nk} \cdot 2^{k-n}.$$

Лемма доказана. ■

Для любого положительного четного числа $m \leqslant t+1$ обозначим

$$\pi_{m,t} = \sum_{a \in V_t} \mathbf{E}(\eta_a(X))^m. \tag{6}$$

Следующая лемма играет ключевую роль в доказательстве теоремы.

Лемма 3. Справедливо неравенство

$$\pi_{m,t} \leqslant \left(1 + \frac{1}{2^t - 1}\right) \sum_{\alpha \in V_n} \left(\hat{f}(\alpha)\right)^m + 2^{m-t-1}\left(1 + \frac{1}{2^t-1}\right)^m. \tag{7}$$

Доказательство. Преобразуем выражение (6):

$$\begin{aligned} \pi_{m,t} &= \sum_{a \in V_t} \mathbf{E}\left(\frac{1}{2^t-1}\sum_{u \in V_t^*}(-1)^{f(uX)\oplus ua}\right)^m = \\ &= \frac{1}{(2^t-1)^m}\mathbf{E}\left(\sum_{u^{(1)},\ldots,u^{(m)} \in V_t^*}\sum_{a \in V_t}(-1)^{f(u^{(1)}X)\oplus\cdots\oplus f(u^{(m)}X)\oplus(u^{(1)}\oplus\cdots\oplus u^{(m)})a}\right) = \\ &= \frac{2^t}{(2^t-1)^m}\mathbf{E}\left(\sum_{u^{(1)},\ldots,u^{(m-1)} \in V_t^*}(-1)^{f(u^{(1)}X)\oplus\cdots\oplus f(u^{(m-1)}X)\oplus f(u^{(1)}X\oplus\cdots\oplus u^{(m-1)}X)}\right) = \\ &= \frac{2^t}{(2^t-1)^m}\sum_{u^{(1)},\ldots,u^{(m-1)} \in V_t^*} 2^{-tn}\sum_{X \in F_{t\times n}}(-1)^{f(u^{(1)}X)\oplus\cdots\oplus f(u^{(m-1)}X)\oplus f(u^{(1)}X\oplus\cdots\oplus u^{(m-1)}X)}. \end{aligned} \tag{8}$$

Представим выражение в правой части (8) в виде суммы двух слагаемых

$$\pi_{m,t}^{(1)} = \frac{2^t}{(2^t-1)^m}\sum_{u^{(1)},\ldots,u^{(m-1)} \in V_t^*}^{(1)}(\cdots), \quad \pi_{m,t}^{(2)} = \frac{2^t}{(2^t-1)^m}\sum_{u^{(1)},\ldots,u^{(m-1)} \in V_t^*}^{(2)}(\cdots),$$

где символы $\Sigma^{(1)}$ и $\Sigma^{(2)}$ обозначают суммы по всем линейно независимым и линейно зависимым системам векторов $u^{(1)},\ldots,u^{(m-1)} \in V_t^*$ соответственно. Используя лемму 2, оценим значение $\pi_{m,t}^{(2)}$ следующим образом:

$$|\pi_{m,t}^{(2)}| = \frac{2^t}{(2^t-1)^m}\left|\sum_{u^{(1)},\ldots,u^{(m-1)} \in V_t^*}^{(2)} 2^{-tn}\sum_{X \in F_{t\times n}}(-1)^{f(u^{(1)}X)\oplus\cdots\oplus f(u^{(m-1)}X)\oplus f(u^{(1)}X\oplus\cdots\oplus u^{(m-1)}X)}\right| \leqslant$$

$$\leqslant \frac{2^t}{(2^t-1)^m} \sum_{u^{(1)},\dots,u^{(m-1)} \in V_t^*}^{(2)} 1 \leqslant \frac{2^t}{(2^t-1)^m} 2^{t(m-1)+m-1-t} = 2^{m-t-1}\left(1+\frac{1}{2^t-1}\right)^m. \qquad (9)$$

Оценим теперь значение $\pi_{m,t}^{(1)}$. Заметим, что если векторы $u^{(1)},\dots,u^{(m-1)} \in V_t$ линейно независимы, то для любого набора векторов $v^{(1)},\dots,v^{(m-1)} \in V_n$ существует ровно $\left(2^{t-(m-1)}\right)^n$ матриц $X \in F_{t\times n}$, таких, что $u^{(i)}X = v^{(i)}$, $i = 1,\dots,m-1$. Следовательно,

$$\pi_{m,t}^{(1)} = \frac{2^t}{(2^t-1)^m} \sum_{u^{(1)},\dots,u^{(m-1)} \in V_t^*}^{(1)} 2^{-tn} \cdot 2^{tn-(m-1)n} \sum_{v^{(1)},\dots,v^{(m-1)} \in V_n} (-1)^{f(v^{(1)})\oplus\dots\oplus f(v^{(m-1)})\oplus f(v^{(1)}\oplus\dots\oplus v^{(m-1)})}.$$

Далее, поскольку

$$\sum_{\alpha\in V_n}\left(\hat f(\alpha)\right)^m = \sum_{\alpha\in V_n} 2^{-mn} \sum_{v^{(1)},\dots,v^{(m)}\in V_n} (-1)^{f(v^{(1)})\oplus\dots\oplus f(v^{(m)})\oplus\alpha(v^{(1)}\oplus\dots\oplus v^{(m)})} =$$
$$= 2^{-(m-1)n} \sum_{v^{(1)},\dots,v^{(m-1)}\in V_n} (-1)^{f(v^{(1)})\oplus\dots\oplus f(v^{(m-1)})\oplus f(v^{(1)}\oplus\dots\oplus v^{(m-1)})},$$

то

$$\pi_{m,t}^{(1)} = \frac{2^t}{(2^t-1)^m} \sum_{\alpha\in V_n}\left(\hat f(\alpha)\right)^m \sum_{u^{(1)},\dots,u^{(m-1)}\in V_t^*}^{(1)} 1 =$$
$$= \frac{2^t(2^t-1)^{m-1}}{(2^t-1)^m} \sum_{\alpha\in V_n}\left(\hat f(\alpha)\right)^m = \left(1+\frac{1}{2^t-1}\right)\sum_{\alpha\in V_n}\left(\hat f(\alpha)\right)^m. \qquad (10)$$

Из соотношений (8)–(10) следует неравенство (7). ■

Лемма 4. Для любого положительного четного числа $m \leqslant t+1$ справедливо неравенство

$$\mathbf{P}\{\hat f_* + \varepsilon \leqslant \eta_*(X)\} \leqslant \varepsilon^{-2}\left(1+\frac{1}{2^t-1}\right)\left(\frac{1}{1+\varepsilon}\right)^{m-2} + \varepsilon^{-m}2^{m-t-1}\left(1+\frac{1}{2^t-1}\right)^m.$$

Доказательство. Из формулы (3) и неравенства Чебышева следует, что

$$\mathbf{P}\{\hat f_* + \varepsilon \leqslant \eta_*(X)\} = \mathbf{P}\left(\bigcup_{a\in V_t}\{\hat f_* + \varepsilon \leqslant |\eta_a(X)|\}\right) \leqslant \sum_{a\in V_t}\mathbf{P}\{\hat f_* + \varepsilon \leqslant |\eta_a(X)|\} \leqslant$$
$$\leqslant (\hat f_* + \varepsilon)^{-m}\sum_{a\in V_t}\mathbf{E}(\eta_a(X))^m = (\hat f_* + \varepsilon)^{-m}\pi_{m,t}.$$

Следовательно, на основании леммы 3

$$\mathbf{P}\{\hat f_* + \varepsilon \leqslant \eta_*(X)\} \leqslant (\hat f_* + \varepsilon)^{-m}\left(1+\frac{1}{2^t-1}\right)\sum_{\alpha\in V_n}\left(\hat f(\alpha)\right)^m + (\hat f_* + \varepsilon)^{-m}2^{m-t-1}\left(1+\frac{1}{2^t-1}\right)^m.$$

Далее, согласно формуле (1) и равенству Парсеваля,

$$\sum_{\alpha\in V_n}\left(\hat f(\alpha)\right)^m \leqslant \hat f_*^{m-2}\sum_{\alpha\in V_n}\left(\hat f(\alpha)\right)^2 = \hat f_*^{m-2},$$

откуда следует, что

$$\mathbf{P}\{\hat f_* + \varepsilon \leqslant \eta_*(X)\} \leqslant$$
$$\leqslant (\hat f_* + \varepsilon)^{-2}\left(1+\frac{1}{2^t-1}\right)\left(\frac{\hat f_*}{\hat f_* + \varepsilon}\right)^{m-2} + (\hat f_* + \varepsilon)^{-m}2^{m-t-1}\left(1+\frac{1}{2^t-1}\right)^m \leqslant$$

$$\leqslant \varepsilon^{-2}\left(1+\frac{1}{2^t-1}\right)\left(\frac{1}{1+\varepsilon}\right)^{m-2}+\varepsilon^{-m}2^{m-t-1}\left(1+\frac{1}{2^t-1}\right)^m.$$

Лемма доказана. ■

Завершение доказательства теоремы. На основании леммы 1 и леммы 4 для любых $n,t,m\in\mathbb{N}$, где $n>t\geqslant m-1$, m четно, $\varepsilon\in(0,1)$ и $f:V_n\to\{0,1\}$, справедливы следующие соотношения:

$$\begin{gathered}\mathbf{P}\{|\tilde{N}_f-\tilde{N}_{\dot{f}_X}|\geqslant\varepsilon\}=\mathbf{P}\{\hat{f}_*\geqslant\eta_*(X)+\varepsilon\}+\mathbf{P}\{\hat{f}_*+\varepsilon\leqslant\eta_*(X)\}\leqslant\\ \leqslant\frac{\varepsilon^{-2}}{2^t-1}+\varepsilon^{-2}\left(1+\frac{1}{2^t-1}\right)\left(\frac{1}{1+\varepsilon}\right)^{m-2}+\varepsilon^{-m}2^{m-t-1}\left(1+\frac{1}{2^t-1}\right)^m\leqslant\\ \leqslant\frac{\varepsilon^{-2}}{2^{m-1}-1}+\varepsilon^{-2}\left(1+\frac{1}{2^{m-1}-1}\right)\left(\frac{1}{1+\varepsilon}\right)^{m-2}+\varepsilon^{-m}2^{m-t-1}\left(1+\frac{1}{2^{m-1}-1}\right)^m.\end{gathered}\tag{11}$$

Пусть теперь $\delta\in(0,1)$. Выберем наименьшее четное число $m_0>0$, такое, что

$$\frac{\varepsilon^{-2}}{2^{m_0-1}-1}+\varepsilon^{-2}\left(1+\frac{1}{2^{m_0-1}-1}\right)\left(\frac{1}{1+\varepsilon}\right)^{m_0-2}\leqslant\frac{\delta}{2},\tag{12}$$

и наименьшее натуральное число $t_0\geqslant m_0-1$, такое, что

$$\varepsilon^{-m_0}2^{m_0-t_0-1}\left(1+\frac{1}{2^{m_0-1}-1}\right)^{m_0}\leqslant\frac{\delta}{2}.\tag{13}$$

В силу соотношений (11) для любых $n>t\geqslant t_0$ выполняется неравенство (5), что и требовалось доказать.

3. Заключительные замечания

Полученная теорема позволяет предложить вероятностный алгоритм оценивания нелинейности функции $f:V_n\to\{0,1\}$ с точностью $2^{n-1}\varepsilon$ и достоверностью не менее $1-\delta$, $\varepsilon,\delta\in(0,1)$, состоящий в вычислении значения случайной величины $2^{n-1}(1-\eta_*(X))$, где X — случайная равновероятная матрица размера $t_0\times n$ над полем F, а число $t_0<n$ определяется из соотношений (12), (13). Нетрудно видеть, что при вычислении всех значений (2) с помощью быстрого преобразования Адамара (см., например, [1, с. 217]) трудоемкость указанного алгоритма составляет $\mathrm{O}(2^{t_0}t_0n)$ двоичных операций, где t_0 зависит только от ε и δ. Однако значения t_0 быстро растут с уменьшением параметра ε, поэтому применение этого алгоритма на практике оказывается неэффективным.

Вместе с тем, согласно лемме 1, для любых $\varepsilon,\delta\in(0,1)$ и $n>t=\lceil\log(1+\varepsilon^{-2}\delta^{-1})\rceil$ справедливо неравенство

$$\mathbf{P}\{2^{n-1}(1-\eta_*(X))-2^{n-1}\varepsilon\leqslant N_f\}\geqslant1-\delta,$$

где X — случайная равновероятная $t\times n$-матрица над полем F, причем для вычисления указанной нижней оценки параметра N_f достаточно выполнить $\mathrm{O}(n\varepsilon^{-2}\delta^{-1}\log(\varepsilon^{-2}\delta^{-1}))$ двоичных операций. Далее, в качестве верхней оценки нелинейности функции f можно использовать случайную величину $2^{n-1}(1-\bar{\eta}_m(X))$, где $\bar{\eta}_m(X)=\left(2^{-t}\sum\limits_{a\in V_t}(\eta_a(X))^m\right)^{\frac{1}{m}}$,

$m \geqslant 4$ — четное число. Опираясь на лемму 3 и проводя рассуждение, почти дословно повторяющее доказательство леммы 4, нетрудно убедиться в том, что для любых $\varepsilon, \delta \in (0,1)$ и $n > t \geqslant m-1$, удовлетворяющих условию

$$\varepsilon^{-2}2^{-t}\left(1+\frac{1}{2^t-1}\right)\left(\frac{1}{1+\varepsilon}\right)^{m-2} \leqslant \frac{\delta}{2}, \quad \varepsilon^{-m}2^{m-2t-1}\left(1+\frac{1}{2^t-1}\right)^{m} \leqslant \frac{\delta}{2},$$

справедливо неравенство

$$\mathbf{P}\{N_f \leqslant 2^{n-1}(1-\bar{\eta}_m(X)) + 2^{n-1}\varepsilon\} \geqslant 1-\delta,$$

где X — случайная равновероятная $t \times n$-матрица над полем F. При фиксированном m и $t = \lceil 1/2 \cdot \log(4^m\varepsilon^{-m}\delta^{-1})\rceil$ для вычисления указанной верхней оценки параметра N_f требуется выполнить $\mathrm{O}(2^t tn) = \mathrm{O}\left(n\varepsilon^{-\frac{m}{2}}\delta^{-\frac{1}{2}}\log(\varepsilon^{-\frac{m}{2}}\delta^{-\frac{1}{2}})\right)$ двоичных и $\mathrm{O}(2^t) = \mathrm{O}\left(\varepsilon^{-\frac{m}{2}}\delta^{-\frac{1}{2}}\right)$ арифметических операций (сложения и возведения в степень вещественных чисел), что приводит к алгоритму, трудоемкость которого полиномиально зависит от n, ε^{-1} и δ^{-1}.

ЛИТЕРАТУРА

1. *Логачев О. А., Сальников А. А., Ященко В. В.* Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2004. 470 с.
2. *Levin L. A.* Randomness and non-determinism // J. Symbolic Logic. 1993. V. 58. No. 3. P. 1102–1103.
3. *Bshouty N., Jackson J., and Tamon C.* More efficient PAC-learning of DNF with membership queries under the uniform distribution // Proc. 12th Annual Conf. on Comput. Learning Theory. NY, USA: ACM,1999. P. 286–295.
4. *Gopalan P., O'Donnell R., Servedio A., et al.* Testing Fourier dimensionality and sparsity // SIAM J. Comput. 2011. V. 40(4). P. 1075–1100.
5. *Алексейчук А. Н., Шевцов А. С.* Быстрый алгоритм статистического оценивания максимальной несбалансированности билинейных аппроксимаций булевых отображений // Прикладная дискретная математика. 2011. № 3(13). С. 5–11.

2012 Теоретические основы прикладной дискретной математики №1(15)

DOI 10.17223/20710410/15/2

УДК 512.642, 519.712.43, 512.541

ТЕОРИЯ ПОЛНЫХ ОРТОГОНАЛЬНЫХ ПРЯМЫХ РАЗЛОЖЕНИЙ ВЕКТОРНЫХ ПРОСТРАНСТВ[1]

М. И. Анохин

Институт проблем информационной безопасности
Московского государственного университета им. М. В. Ломоносова, г. Москва, Россия

E-mail: anokhin@mccme.ru

Построена теория полных ортогональных (относительно обобщенных ортогональностей, заданных некоторыми частичными симметричными билинейными функциями) прямых разложений векторных пространств V, таких, что фактор-пространство V по некоторому специальному подпространству конечномерно. Основным результатом этой теории является некоторое описание всех таких разложений. Вышеупомянутая теория имеет приложение к теории прямых разложений p-ичных функций, где p — простое число.

Ключевые слова: *векторное пространство, ортогональность, частичная симметричная билинейная функция, полное разложение, абелева группа.*

1. Введение

1.1. Прямые разложения функций φ и ∇_φ-ортогональные прямые разложения абелевых групп

В настоящей работе продолжены исследования, начатые в [1]. В более общем виде, чем в [1], предмет этих исследований можно описать следующим образом. Пусть A и B — абелевы группы, φ — функция из A в B, $A = \bigoplus\limits_{m\in M} A_m$ — некоторое разложение группы A в прямую сумму семейства подгрупп $(A_m \,|\, m \in M)$, индексированного элементами конечного множества M (такие разложения будем для краткости называть *конечными прямыми* разложениями). Тогда естественно сказать, что функция φ *разлагается в прямую сумму* семейства функций $(\varphi_m \,|\, m \in M)$ для указанного прямого разложения группы A, если $\varphi_m \colon A_m \to B$ и

$$\varphi\left(\sum_{m\in M} x_m\right) = \sum_{m\in M} \varphi_m(x_m)$$

при любых $x_m \in A_m$, $m \in M$. Такие разложения, называемые в дальнейшем *прямыми* разложениями функции φ, будем записывать в виде $\varphi = \bigoplus\limits_{m\in M} \varphi_m$ и отождествлять с соответствующими семействами функций $(\varphi_m \,|\, m \in M)$. Нас интересуют всевозможные прямые разложения функции φ и, в частности, конечные прямые разложения группы A, для которых прямые разложения φ существуют.

Определим теперь зависящее от функции φ бинарное отношение ∇_φ на группе A, которое играет важную роль в изучении прямых разложений этой функции. Для произвольного $a \in A$ обозначим через $\mathbf{D}_a\,\varphi$ функцию из A в B, определенную равенством

$$(\mathbf{D}_a\,\varphi)(x) = \varphi(x+a) - \varphi(x)$$

[1]Работа поддержана грантами РФФИ № 07-01-00154, 10-01-00475.

при всех $x \in A$. Функция $\mathbf{D}_a\,\varphi$ называется *производной* функции φ по направлению a; это понятие хорошо известно (см., например, [2, определение 3], [3, определение 2.68]). Тогда $u \bigtriangledown_\varphi w$ (где $u, w \in A$) в том и только в том случае, когда функция $\mathbf{D}_u\,\mathbf{D}_w\,\varphi$ тождественно равна нулю на A. Отношение $\bigtriangledown_\varphi$ обладает свойствами, аналогичными некоторым свойствам ортогональностей на векторных пространствах. Например, $u \bigtriangledown_\varphi w$ тогда и только тогда, когда $\sigma_\varphi(u, w) = 0$, где $u, w \in A$, а σ_φ — частичная симметричная билинейная над кольцом целых чисел функция на A (в смысле естественного аналога определения из п. 4.1), принимающая значения в B и заданная следующим образом:

$$\sigma_\varphi(u, w) \begin{cases} = b, \text{ если } \exists\, b \in B\ \forall\, x \in A\ (\mathbf{D}_u\,\mathbf{D}_w\,\varphi)(x) = b; \\ \text{не определено в противном случае.} \end{cases} \tag{1}$$

Другими словами, (1) означает, что $\sigma_\varphi(u, w)$ определено, если и только если функция $\mathbf{D}_u\,\mathbf{D}_w\,\varphi$ тождественно равна на A некоторому элементу $b \in B$, который в этом случае и является значением $\sigma_\varphi(u, w)$. Кроме того, $0 \bigtriangledown_\varphi a$ (или, что эквивалентно, $\sigma_\varphi(0, a) = 0$) при всех $a \in A$. Следовательно, если назвать *ортогональностью* на A всякое бинарное симметричное отношение $\perp$ на этой группе, такое, что для любого $a \in A$ множество $\mathsf{C}_A^{\perp}(a) = \{x \in A \,|\, x \perp a\}$ является подгруппой A, то $\bigtriangledown_\varphi$ — ортогональность на A. Ортогональностью на A является также область определения $\diamondsuit_\varphi$ функции σ_φ, рассматриваемая как бинарное отношение. Кроме того, $\diamondsuit_\varphi$ удовлетворяет следующему условию:

$$\forall\, x, y \in A\ \left(\mathsf{C}_A^{\diamondsuit_\varphi}(x) + \mathsf{C}_A^{\diamondsuit_\varphi}(y) = A \implies \mathsf{C}_A^{\diamondsuit_\varphi}(x + y) = \mathsf{C}_A^{\diamondsuit_\varphi}(x) \cap \mathsf{C}_A^{\diamondsuit_\varphi}(y)\right). \tag{2}$$

Это доказывается точно так же, как и соответствующее утверждение в лемме 10 из [1]. Очевидно также, что если A и B являются элементарными абелевыми 2-группами (другими словами, аддитивными группами векторных пространств над полем $\mathrm{GF}(2)$), то $a \bigtriangledown_\varphi a$ для всех $a \in A$.

В этом пункте через $\perp$ будем обозначать произвольную ортогональность на A. Прямое разложение $A = \bigoplus\limits_{m \in M} A_m$ группы A называется $\perp$-*ортогональным*, если $a_1 \perp a_2$ для любых $a_1 \in A_{m_1}$, $a_2 \in A_{m_2}$, где $m_1, m_2 \in M$ и $m_1 \neq m_2$. Связь $\bigtriangledown_\varphi$-ортогональных прямых разложений группы A с прямыми разложениями функции φ выражается следующей теоремой, которая доказывается точно так же, как и теорема 1 из [1].

Теорема 1.1. Пусть $A = \bigoplus\limits_{m \in M} A_m$ — произвольное конечное прямое разложение группы A. Тогда для этого разложения существуют прямые разложения функции φ, если и только если оно $\bigtriangledown_\varphi$-ортогонально и $(M \neq \varnothing \vee \varphi(0) = 0)$. Кроме того, если выполнены эквивалентные условия предыдущего предложения, то прямые разложения функции φ для рассматриваемого прямого разложения группы A — это в точности разложения вида

$$\varphi = \bigoplus_{m \in M} (\varphi|_{A_m} + b_m),$$

где $b_m \in B$ (рассматриваемый в указанном прямом разложении φ как функция-константа на A_m) и $\sum\limits_{m \in M} b_m = (1 - |M|)\varphi(0)$.

Теорема 1.1 показывает, что изучение прямых разложений произвольной функции $\varphi\colon A \to B$ эквивалентно изучению $\bigtriangledown_\varphi$-ортогональных конечных прямых разложений группы A. При этом, очевидно, достаточно ограничиться *нетривиальными* прямыми

разложениями группы A, т. е. прямыми разложениями этой группы, состоящими из ненулевых подгрупп. Кроме того, если группа A удовлетворяет условию минимальности для подгрупп, то всякое нетривиальное $\perp$-ортогональное конечное прямое разложение группы A может быть продолжено до полного $\perp$-ортогонального конечного прямого разложения этой группы. Поэтому в данном случае изучение прямых разложений функции φ сводится к изучению полных ∇_φ-ортогональных конечных прямых разложений этой группы. Здесь $\perp$-ортогональное прямое разложение группы A называется *полным $\perp$-ортогональным прямым* разложением, если оно нетривиально и не может быть продолжено до какого-либо другого нетривиального $\perp$-ортогонального прямого разложения или, другими словами, если оно состоит из $\perp$-*неразложимых* подгрупп (т. е. ненулевых подгрупп, не допускающих разложения в прямую сумму двух $\perp$-ортогональных ненулевых подгрупп; $\perp$-ортогональность подгрупп U и W означает, что $u \perp w$ для всех $u \in U$ и $w \in W$).

1.2. Краткое изложение результатов

В п. 1.1 мы видели, что если A и B — абелевы группы, причем A удовлетворяет условию минимальности для подгрупп, то изучение прямых разложений произвольной функции $\varphi\colon A \to B$ в некотором смысле эквивалентно изучению полных ∇_φ-ортогональных конечных прямых разложений группы A. Однако, по мнению автора, изучение последних в общем случае очень сложно и вряд ли может привести к каким-либо глубоким результатам. Материал п. 1.1 был приведен в основном с целью обоснования актуальности задачи. В настоящей работе рассматривается несколько другой случай, а именно: пусть V — векторное пространство над произвольным полем F. Все введенные выше понятия, относящиеся к абелевым группам (производная функции по направлению, ортогональность и т. д.), естественно переносятся на случай векторных пространств. Например, *ортогональностью* на V называется любое бинарное симметричное отношение $\perp$ на этом пространстве, такое, что для любого $a \in V$ множество $\mathsf{C}_V^\perp(a) = \{x \in V \mid x \perp a\}$ является подпространством пространства V (см. также п. 3.1). Пусть σ — частичная симметричная билинейная над полем F функция на пространстве V, принимающая значения в F и такая, что $\sigma(0, a)$ определено (и, следовательно, равно 0) для всех $a \in V$ (подробнее см. п. 4.1). Тогда ортогональностями на V являются область определения $\Diamond$ функции σ (рассматриваемая как бинарное отношение) и бинарное отношение ∇, определенное следующим образом:

$$x \mathrel{\nabla} y \iff (x, y) \in \Diamond \ \ \& \ \ \sigma(x, y) = 0,$$

где $x, y \in V$. Возможно, что ортогональности (в указанном выше смысле) и частичные симметричные билинейные функции на векторных пространствах представляют самостоятельный интерес и заслуживают отдельного изучения.

В отличие от п. 1.1, здесь мы не предполагаем, что функция σ связана с какой-либо функцией φ. Предположим, что φ — произвольная функция из V в F. Тогда взять в качестве σ функцию σ_φ, определенную формулой (1), не всегда возможно, так как σ_φ будет частичной симметричной билинейной функцией над кольцом целых чисел и, если $\operatorname{char} F \neq 0$, над простым подполем поля F, но, вообще говоря, не над F. В случае, когда σ_φ является частичной симметричной билинейной функцией над полем F, мы называем σ_φ, $\Diamond_\varphi$ и ∇_φ *ассоциированными* с функцией φ.

Пусть T — множество всех $a \in V$, таких, что $\mathsf{C}_V^\Diamond(a) = V$ и $a \mathrel{\nabla} b$ для любого $b \in V$, удовлетворяющего равенству $\mathsf{C}_V^\Diamond(b) = V$ (см. также (17)). Предположим, что простран-

ство V/T конечномерно и

$$\forall\, x, y \in V \ \left(\mathsf{C}_V^{\Diamond}(x) + \mathsf{C}_V^{\Diamond}(y) = V \implies \mathsf{C}_V^{\Diamond}(x+y) = \mathsf{C}_V^{\Diamond}(x) \cap \mathsf{C}_V^{\Diamond}(y)\right). \tag{3}$$

В этом случае удалось построить некоторую теорию полных ∇-ортогональных прямых разложений пространства V. Как уже отмечалось в п. 1.1 (см. условие (2)), если ортогональность $\Diamond$ ассоциирована с некоторой функцией из V в F, то условие (3) выполнено.

Основным результатом настоящей работы является описание всех полных ∇-ортогональных прямых разложений пространства V (теорема 5.2). Это описание или его промежуточный вариант (теорема 5.1) существенно используются в доказательстве критериев каждого из следующих условий:

— ∇-неразложимость пространства V (теорема 6.1). В некотором частном случае критерий этого условия был дан в § 5 работы [1];
— σ-эквивалентность двух полных ∇-ортогональных прямых разложений пространства V, если $\operatorname{char} F \neq 2$ (п. 1 теоремы 6.2). Здесь два прямых разложения пространства V называются *σ-эквивалентными*, если одно из них (с точностью до перестановки индексов) переводится в другое некоторым автоморфизмом этого пространства, сохраняющим σ. Если же $\operatorname{char} F = 2$, то при некотором дополнительном условии (которое заведомо выполнено в случае, когда σ ассоциирована с некоторой функцией из V в F) любые два полных ∇-ортогональных прямых разложения пространства V σ-эквивалентны (п. 2 теоремы 6.2). Кроме того, показано, что если $\operatorname{char} F \neq 2$, то для любых двух полных ∇-ортогональных прямых разложений пространства V имеет место локальная $(\Diamond, \nabla)$-эквивалентность, являющаяся некоторым ослабленным вариантом σ-эквивалентности (теорема 6.3);
— единственность полного ∇-ортогонального прямого разложения пространства V (теорема 6.4). Ранее в теореме 3 работы [1] было дано некоторое условие, достаточное для этого (в рассматриваемом там частном случае).

Рассмотрим случай, когда поле F конечно и пространство V конечномерно. Ценность указанных результатов состоит еще и в том, что в данном случае они имеют в некотором смысле эффективный характер. А именно, предположим, что носители F и V содержатся в множестве всех конечных строк, состоящих из нулей и единиц. Пусть также поле F задается для алгоритмов с помощью своего носителя и таблиц сложения и умножения, пространство V — посредством своего носителя вместе с таблицами сложения в V и умножения элементов V на элементы F, а функция σ — в виде таблицы своих значений. Тогда из описания всех полных ∇-ортогональных прямых разложений пространства V следует существование полиномиального недетерминированного алгоритма, множество всех выходных значений которого при вычислении на произвольном входе вида (F, V, σ) совпадает с множеством всех таких разложений пространства V (теорема 6.5). Кроме того, полученные критерии перечисленных выше условий могут быть проверены детерминированными алгоритмами за полиномиальное время по (F, V, σ) (теоремы 6.6 и 6.7; разумеется, предполагается, что при проверке σ-эквивалентности двух полных ∇-ортогональных прямых разложений пространства V на вход алгоритму подаются также эти разложения). Подробнее об этом см. в п. 6.4.

Предположим теперь, что $F = \mathrm{GF}(p)$, где p — простое число, V конечномерно и φ — произвольная функция из V в F. Такие функции φ называют p-ичными; именно этот случай рассматривался в работе [1]. Тогда σ_φ является частичной симметричной билинейной над полем F функцией на V, причем, как уже отмечалось, $\sigma_\varphi(0, a) = 0$

для всех $a \in V$ и для $\Diamond = \Diamond_\varphi$ выполнено условие (3). Это показывает, что к $\sigma = \sigma_\varphi$ применима теория, построенная в настоящей работе. В то же время указанный случай по существу эквивалентен частному случаю ситуации п. 1.1, когда A — конечная элементарная абелева p-группа, а B — циклическая группа порядка p. Поэтому ввиду сказанного в п. 1.1 мы получаем некоторую теорию прямых разложений функций из A в B, где A и B указаны в предыдущем предложении. Возможно, что общая теория полных $\triangledown$-ортогональных прямых разложений векторных пространств, построенная в настоящей работе, имеет и другие приложения.

2. Определения, обозначения и необходимые факты

На протяжении всей работы через F обозначается некоторое поле, а через $\operatorname{char} F$ — характеристика этого поля. Пусть также F^* — мультипликативная группа всех ненулевых элементов поля F, а $(F^*)^2 = \{\, x^2 \mid x \in F^* \}$. Если не оговорено иное, все объекты и понятия линейной алгебры рассматриваются над основным полем F. Как обычно, через $\mathrm{GF}(q)$ обозначаем конечное поле из q элементов.

Под термином «граф» в настоящей работе понимается неориентированный граф без петель и кратных ребер. Мы допускаем случай, когда граф не имеет вершин. В этом случае граф не имеет также ребер и связных компонент (последние предполагаются непустыми). Внутреннюю прямую сумму подгрупп абелевой группы или подпространств векторного пространства обозначаем символом $\oplus$, а объединение попарно непересекающихся множеств — символом $\sqcup$. Отметим, что символы $\subset$ и $\supset$ в настоящей работе обозначают строгие включения. Семейства каких-либо элементов записываются с помощью круглых скобок (например, $(x_m \mid m \in M)$); при этом мы не предполагаем, что элементы семейства, соответствующие разным индексам, различны. Для записи множеств, как обычно, используются фигурные скобки; в отличие от семейств, каждый элемент множества входит в него лишь один раз. Мы обозначаем одним и тем же символом (например, π) соответствующие друг другу бинарное отношение на произвольном множестве X, бинарный предикат на этом множестве и подмножество $X \times X$. При этом

$$a \,\pi\, b \iff \pi(a,b) = 1 \iff (a,b) \in \pi$$

для произвольных $a, b \in X$. Здесь в первом условии π рассматривается как бинарное отношение на X, во втором — как бинарный предикат на X, а в третьем — как подмножество $X \times X$. Произвольное бинарное отношение π на множестве X естественно переносится с элементов на подмножества этого множества, а именно: если $A, B \subseteq X$, то запись $A \,\pi\, B$ означает, что $a \,\pi\, b$ для всех $a \in A$ и $b \in B$. Кроме того, вместо $\{a\} \,\pi\, B$ и $A \,\pi\, \{b\}$, где $a, b \in X$ и $A, B \subseteq X$, пишем $a \,\pi\, B$ и $A \,\pi\, b$ соответственно.

Через V всегда обозначаем некоторое векторное пространство. Пространство V может иметь специальный вид, если это указано явно. Пусть A — подмножество, а W — подпространство V. Тогда $\langle A \rangle$ обозначает подпространство V, порожденное множеством A. Если $A = \{a_1, \ldots, a_n\}$ то, как обычно, будем писать $\langle a_1, \ldots, a_n \rangle$ вместо $\langle \{a_1, \ldots, a_n\} \rangle$. Через $\nu_{A,W}$ будем обозначать ограничение на A естественного гомоморфизма V на V/W. Всякое подпространство R пространства V, удовлетворяющее равенству $V = R \oplus W$, называется *прямым дополнением* к W в V. Очевидно, что если R — произвольное прямое дополнение к W в V, то $\nu_{R,W}$ — изоморфизм R на V/W. Последним утверждением будем пользоваться, не оговаривая этого особо. Произвольный элемент вида $v + \{0\} \in V/\{0\}$, где $v \in V$, естественно отождествляется с v, а фактор-

пространство $V/\{0\}$ — с V. Под суммой $\sum\limits_{m\in M} U_m$ некоторого семейства $(U_m \mid m \in M)$ подпространств V, как обычно, понимается $\left\langle \bigcup\limits_{m\in M} U_m \right\rangle$.

Если $\mathfrak{S}$ — некоторое множество подпространств пространства V, то будем писать $\sum \mathfrak{S}$ и $\bigoplus \mathfrak{S}$ вместо $\sum\limits_{S\in\mathfrak{S}} S$ и $\bigoplus\limits_{S\in\mathfrak{S}} S$ соответственно, чтобы не загромождать обозначения. Таким образом, здесь множество $\mathfrak{S}$ рассматривается как семейство $(S \mid S \in \mathfrak{S})$. Пусть теперь $\mathfrak{X}$ — некоторое множество подмножеств какого-либо множества M. Тогда, по аналогии с вышеуказанными обозначениями, вместо $\bigcup\limits_{X\in\mathfrak{X}} X$ и $\bigsqcup\limits_{X\in\mathfrak{X}} X$ используем записи $\bigcup \mathfrak{X}$ и $\bigsqcup \mathfrak{X}$ соответственно. Кроме того, если μ — некоторое отображение, определенное на M, то через $\mu(\mathfrak{X})$ будем обозначать множество $\{\mu(X) \mid X \in \mathfrak{X}\}$.

Под словами «разложение (векторного) пространства» понимается разложение этого пространства в сумму некоторого семейства его подпространств. Такие разложения естественно отождествляются с соответствующими семействами подпространств. Будем пользоваться (не оговаривая этого особо) следующим очевидным замечанием: если $V = \sum\limits_{m\in M} U_m$ и $V = \bigoplus\limits_{m\in M} W_m$ — разложения пространства V, причем $U_m \subseteq W_m$ при любом $m \in M$, то $U_m = W_m$ для всех $m \in M$.

Предположим, что на пространстве V определены предикаты $\pi_1, \ldots, \pi_k$, имеющие конечные арности $l_1, \ldots, l_k$ соответственно. Тогда соответствующую алгебраическую систему сигнатуры, состоящую из символов стандартной сигнатуры векторных пространств над полем F и символа l_i-арного предиката для каждого $i \in \{1, \ldots, k\}$, будем обозначать через $(V; \pi_1, \ldots, \pi_k)$. Вообще говоря, предикаты $\pi_1, \ldots, \pi_k$ могут быть частичными и/или многозначными; в этом случае мы рассматриваем алгебраические системы в несколько обобщенном смысле. Для обобщенных алгебраических систем указанного вида используем только понятия изоморфизма и автоморфизма, которые определяются естественным образом. А именно, пусть на некотором векторном пространстве V' определены предикаты $\pi'_1, \ldots, \pi'_k$, причем π'_i имеет ту же арность l_i, что и π_i, для каждого $i \in \{1, \ldots, k\}$. Тогда изоморфизм векторных пространств $\alpha\colon V \to V'$ называется *изоморфизмом* $(V; \pi_1, \ldots, \pi_k)$ на $(V'; \pi'_1, \ldots, \pi'_k)$, если для любых $i \in \{1, \ldots, k\}$ и $v_1, \ldots, v_{l_i} \in V$

$$\pi'_i(\alpha(v_1), \ldots, \alpha(v_{l_i})) \text{ определено } \iff \pi_i(v_1, \ldots, v_{l_i}) \text{ определено,}$$

и в этом случае $\pi'_i(\alpha(v_1), \ldots, \alpha(v_{l_i})) = \pi_i(v_1, \ldots, v_{l_i})$. Отношение изоморфизма для алгебраических систем будет обозначаться символом $\cong$. Кроме того, как обычно, $\operatorname{Aut} A$ обозначает группу всех автоморфизмов произвольной алгебраической системы A. Разложения $V = \sum\limits_{m\in M} U_m$ и $V = \sum\limits_{n\in N} W_n$ пространства V называем $(\pi_1, \ldots, \pi_k)$-*эквивалентными*, если существуют биекция $\mu\colon M \to N$ и автоморфизм $\alpha \in \operatorname{Aut}(V; \pi_1, \ldots, \pi_k)$, такие, что $\alpha(U_m) = W_{\mu(m)}$ для всех $m \in M$. Назовем также указанные разложения пространства V *локально* $(\pi_1, \ldots, \pi_k)$-*эквивалентными*, если существует биекция $\mu\colon M \to N$, такая, что $(U_m; \pi_1, \ldots, \pi_k) \cong (W_{\mu(m)}; \pi_1, \ldots, \pi_k)$ для всех $m \in M$. При $k = 1$ опускаем скобки вокруг π_1, когда речь идет о (локальной) (π_1)-эквивалентности разложений. Очевидно, что из $(\pi_1, \ldots, \pi_k)$-эквивалентности двух разложений пространства V следует их локальная $(\pi_1, \ldots, \pi_k)$-эквивалентность. Легко также видеть, что разложения $V = \Sigma\mathfrak{S}$ и $V = \Sigma\mathfrak{T}$ пространства V $(\pi_1, \ldots, \pi_k)$-эквивалентны тогда и только тогда, когда $\alpha(\mathfrak{S}) = \mathfrak{T}$ для некоторого $\alpha \in \operatorname{Aut}(V; \pi_1, \ldots, \pi_k)$, и локаль-

но $(\pi_1, \ldots, \pi_k)$-эквивалентны, если и только если $(U; \pi_1, \ldots, \pi_k) \cong (W_U; \pi_1, \ldots, \pi_k)$ для некоторой биекции $U \mapsto W_U$ множества $\mathfrak{S}$ на множество $\mathfrak{T}$.

Следующая лемма обобщает лемму 5 из [1] и доказывается аналогично ей.

Лемма 2.1. Пусть ω — какое-либо отображение из V в некоторое частично упорядоченное множество, удовлетворяющее условию максимальности; I — множество всех $v \in V \setminus \{0\}$, для которых не существует представления в виде $v_1 + v_2$, где $v_i \in V$ и $\omega(v_i) > \omega(v)$ при $i = 1, 2$. Тогда $\langle I \rangle = V$.

Доказательство. Предположим, что множество $V \setminus \langle I \rangle$ непусто. Выберем некоторый элемент $v \in V \setminus \langle I \rangle$, для которого $\omega(v)$ максимален в множестве $\omega(V \setminus \langle I \rangle)$. Так как $v \in V \setminus \{0\}$ и $v \notin I$, v может быть представлен в виде $v_1 + v_2$, где $v_i \in V$ и $\omega(v_i) > \omega(v)$ при $i = 1, 2$. Но тогда $v_1, v_2 \in \langle I \rangle$ (ввиду максимальности $\omega(v)$ в $\omega(V \setminus \langle I \rangle)$) и, следовательно, $v \in \langle I \rangle$, что противоречит выбору v. ■

3. Ортогональности на векторных пространствах

3.1. Определения, обозначения и необходимые факты

Следуя [1], назовем *ортогональностью* на векторном пространстве V любое бинарное симметричное отношение $\perp$ на этом пространстве, такое, что для любого $a \in V$ множество $\{x \in V \mid x \perp a\}$ является подпространством пространства V. Здесь через $\perp$ будем обозначать некоторую ортогональность на V. Ортогональность $\perp$ может иметь специальный вид, если это указано явно. Алгебраическую систему $(V; \perp)$, где $\perp$ рассматривается как бинарный предикат, естественно назвать (*векторным*) *пространством с ортогональностью*. Если $a \perp b$, где $a, b \in V$, то говорят, что элементы a и b $\perp$-*ортогональны*. Аналогично, подмножества A и B пространства V называются $\perp$-*ортогональными*, если $A \perp B$.

Для множества $A \subseteq V$ его $\perp$-*ортогональное дополнение* $\mathsf{C}_B^{\perp}(A)$ в множестве $B \subseteq V$ определяется равенством

$$\mathsf{C}_B^{\perp}(A) = \{b \in B \mid b \perp a \text{ для всех } a \in A\}.$$

Очевидно, что если U — подпространство V, то $\mathsf{C}_U^{\perp}(A)$ является подпространством U. Вместо $\mathsf{C}_B^{\perp}(\{a_1, \ldots a_n\})$, где $a_1, \ldots a_n \in V$ и $B \subseteq V$, пишем $\mathsf{C}_B^{\perp}(a_1, \ldots a_n)$. Отметим, что в работе [1] множества $\mathsf{C}_V^{\perp}(a)$ и $\mathsf{C}_V^{\perp}(A)$ обозначались соответственно через $a^{\perp}$ и $A^{\perp}$.

Легко видеть, что класс всех векторных пространств с ортогональностями является квазимногообразием (напомним, что мы рассматриваем векторные пространства над фиксированным полем F). Поэтому, в частности, этот класс замкнут относительно взятия подсистем и декартовых произведений (а следовательно, и прямых произведений). Разумеется, под прямым произведением семейства $((V_m; \perp_m) \mid m \in M)$ векторных пространств с ортогональностями понимается подсистема декартова произведения этого семейства, состоящая из всех $(a_m \mid m \in M)$, таких, что $a_m \in V_m$ и множество $\{m \in M \mid a_m \neq 0\}$ конечно. Пусть теперь W — произвольное подпространство $\mathsf{C}_V^{\perp}(V)$. Тогда фактор-система пространства с ортогональностью $(V; \perp)$ по W является пространством с ортогональностью. Полученная при взятии этой фактор-системы ортогональность на V/W совпадает с ограничением на множество всех смежных классов V по W определенного выше отношения $\perp$-ортогональности подмножеств V. Поэтому указанную ортогональность на V/W будем обозначать так же, как и исходную ортогональность на V. Очевидно, что

$$a + W \perp b + W \iff a \perp b \tag{4}$$

для произвольных $a, b \in V$. Используя эту эквивалентность, легко получить равенство

$$\mathsf{C}^{\perp}_{V/W}(V/W) = \mathsf{C}^{\perp}_{V}(V)/W. \tag{5}$$

В частности,

$$\mathsf{C}^{\perp}_{V/\mathsf{C}^{\perp}_{V}(V)}\left(V/\mathsf{C}^{\perp}_{V}(V)\right) = \{0\} \tag{6}$$

(см. также лемму 4 из [1]).

Замечание 3.1. Пусть W — подпространство $\mathsf{C}^{\perp}_{V}(V)$, а R — произвольное прямое дополнение к W в V. Тогда ввиду эквивалентности (4) $\nu_{R,W}$ является изоморфизмом $(R;\perp)$ на $(V/W;\perp)$.

3.2. Ортогональные разложения векторных пространств

Приведем определения нескольких понятий, важных для настоящей работы. Разложение $V = \sum\limits_{m\in M} V_m$ пространства V (где M — некоторое множество) назовём *нетривиальным*, если $V_m \neq \{0\}$ для всех $m \in M$. Вышеуказанное разложение пространства V называется $\perp$-*ортогональным*, если $V_{m_1} \perp V_{m_2}$ для любых различных $m_1, m_2 \in M$. Пространство $V \neq \{0\}$ называется $\perp$-*разложимым*, если существует нетривиальное $\perp$-ортогональное разложение этого пространства в прямую сумму двух подпространств; в противном случае V называется $\perp$-*неразложимым*. Отметим, что $\{0\}$ не считается ни $\perp$-разложимым, ни $\perp$-неразложимым пространством. Нетривиальное $\perp$-ортогональное прямое разложение $V = \bigoplus\limits_{m\in M} V_m$ пространства V называется *полным* $\perp$-*ортогональным* разложением, если для каждого $m \in M$ подпространство V_m $\perp$-неразложимо или, другими словами, если это разложение не может быть продолжено до какого-либо другого нетривиального $\perp$-ортогонального прямого разложения. Очевидно, что если $V = \bigoplus\limits_{m\in M} V_m$ — нетривиальное прямое разложение V (в частности, полное $\perp$-ортогональное разложение), то $V_{m_1} \neq V_{m_2}$ при $m_1 \neq m_2$ $(m_1, m_2 \in M)$, поэтому данное разложение можно записать в виде $V = \bigoplus \mathfrak{S}$, где $\mathfrak{S} = \{V_m \,|\, m \in M\}$.

Замечание 3.2 [1, лемма 3]. Предположим, что $\mathsf{C}^{\perp}_{V}(V) = \{0\}$. Пусть $V = U + W$ — $\perp$-ортогональное разложение пространства V. Тогда $W = \mathsf{C}^{\perp}_{V}(U)$. Действительно, включение $W \subseteq \mathsf{C}^{\perp}_{V}(U)$ тривиально. Пусть теперь $v \in \mathsf{C}^{\perp}_{V}(U)$. Представим v в виде $u + w$, где $u \in U$ и $w \in W$. Тогда $u \perp U$ (так как $v \perp U$ и $w \perp U$) и $u \perp W$. Поэтому $u \perp V$. Следовательно, $u = 0$ и $v = w \in W$. Таким образом, $\mathsf{C}^{\perp}_{V}(U) \subseteq W$.

Замечание 3.3. Предположим, что $\mathsf{C}^{\perp}_{V}(V) = \{0\}$. Тогда всякое $\perp$-ортогональное разложение $V = \sum\limits_{m\in M} V_m$ пространства V является прямым. Действительно, пусть $v \in V_m \bigcap \sum\limits_{k\in M\setminus\{m\}} V_k$, где $m \in M$. Тогда $v \perp V_k$ для любого $k \in M\setminus\{m\}$ (так как $v \in V_m$) и $v \perp V_m$ (так как $v \in \sum\limits_{k\in M\setminus\{m\}} V_k$). Поэтому $v \perp V$ и, следовательно, $v = 0$. Таким образом, $V_m \bigcap \sum\limits_{k\in M\setminus\{m\}} V_k = \{0\}$ для любого $m \in M$ (см. также лемму 3 работы [1]).

Следующая лемма усиливает лемму 6 из [1].

Лемма 3.1. Пусть $V = \sum\limits_{m\in M} V_m$ — $\perp$-ортогональное разложение пространства V. Тогда

$$\mathsf{C}^{\perp}_{V}(V) = \sum_{m\in M} \mathsf{C}^{\perp}_{V_m}(V), \tag{7}$$

причем $\mathsf{C}^{\perp}_{V_m}(V) = \mathsf{C}^{\perp}_{V}(V) \cap V_m = \mathsf{C}^{\perp}_{V_m}(V_m)$ для всех $m \in M$.

Доказательство. Включение $\sum\limits_{m\in M}\mathsf{C}^{\perp}_{V_m}(V)\subseteq\mathsf{C}^{\perp}_{V}(V)$ тривиально. Пусть теперь $v\in\mathsf{C}^{\perp}_{V}(V)$. Представим v в виде $\sum\limits_{m\in M}v_m$, где $v_m\in V_m$ при всех $m\in M$, причем $\{m\in M\,|\,v_m\neq 0\}$ конечно. Для произвольного $m\in M$ положим $v'_m=\sum\limits_{k\in M\setminus\{m\}}v_k$. Тогда $v_m=v-v'_m\perp V_m$ (так как $v\perp V_m$ и $v'_m\perp V_m$) и $v_m\perp\sum\limits_{k\in M\setminus\{m\}}V_k$, поэтому $v_m\perp V$ при любом $m\in M$. Следовательно, $v\in\sum\limits_{m\in M}\mathsf{C}^{\perp}_{V_m}(V)$. Таким образом, равенство (7) доказано. Для произвольного $m\in M$ равенство $\mathsf{C}^{\perp}_{V_m}(V)=\mathsf{C}^{\perp}_{V}(V)\cap V_m$ очевидно, а равенство $\mathsf{C}^{\perp}_{V_m}(V)=\mathsf{C}^{\perp}_{V_m}(V_m)$ вытекает из $\perp$-ортогональности разложения $V=\sum\limits_{m\in M}V_m$. Лемма доказана. ■

Следующая лемма показывает, что задача описания всех полных $\perp$-ортогональных разложений пространства V эквивалентна задаче описания всех таких разложений пространства $V/\mathsf{C}^{\perp}_{V}(V)$.

Лемма 3.2. Пусть $S=\mathsf{C}^{\perp}_{V}(V)$, а $\mathcal{D}$ — множество всех троек $(\mathfrak{B},R,\mathfrak{C})$, таких, что $S=\bigoplus\mathfrak{B}$ — разложение пространства S в прямую сумму одномерных подпространств, R — прямое дополнение к S в V и $V/S=\bigoplus\mathfrak{C}$ — полное $\perp$-ортогональное разложение пространства V/S. Тогда отображение

$$(\mathfrak{B},R,\mathfrak{C})\mapsto\mathfrak{B}\sqcup\nu^{-1}_{R,S}(\mathfrak{C}),\quad(\mathfrak{B},R,\mathfrak{C})\in\mathcal{D},\tag{8}$$

является биекцией $\mathcal{D}$ на множество всех $\mathfrak{S}$, таких, что $V=\bigoplus\mathfrak{S}$ — полное $\perp$-ортогональное разложение пространства V. Если $\mathfrak{S}$ — образ тройки $(\mathfrak{B},R,\mathfrak{C})\in\mathcal{D}$ при отображении (8), то эти $\mathfrak{B}$, R и $\mathfrak{C}$ могут быть найдены по $\mathfrak{S}$ следующим образом:

$$\mathfrak{B}=\{U\in\mathfrak{S}\,|\,U\cap S\neq\{0\}\}=\{U\in\mathfrak{S}\,|\,\mathsf{C}^{\perp}_{U}(U)\neq\{0\}\}=\{U\in\mathfrak{S}\,|\,U\subseteq S\};\tag{9}$$

$$R=\bigoplus(\mathfrak{S}\setminus\mathfrak{B});\tag{10}$$

$$\mathfrak{C}=\nu_{R,S}(\mathfrak{S}\setminus\mathfrak{B}).\tag{11}$$

Доказательство. Пусть $(\mathfrak{B},R,\mathfrak{C})\in\mathcal{D}$. Из замечания 3.1 следует, что

$$R=\bigoplus\nu^{-1}_{R,S}(\mathfrak{C})\tag{12}$$

— полное $\perp$-ортогональное разложение пространства R. Поэтому

$$V=S\oplus R=(\bigoplus\mathfrak{B})\oplus\big(\bigoplus\nu^{-1}_{R,S}(\mathfrak{C})\big)$$

является полным $\perp$-ортогональным разложением пространства V ($\perp$-неразложимость всех пространств из $\mathfrak{B}$ и условие $S\perp R$ тривиальны).

Предположим теперь, что $V=\bigoplus\mathfrak{S}$ — полное $\perp$-ортогональное разложение пространства V. Пусть $\mathfrak{B}$, R и $\mathfrak{C}$ определены формулами (9)–(11). Из леммы 3.1 вытекает, что

$$S=\bigoplus_{U\in\mathfrak{S}}(U\cap S)=\bigoplus_{U\in\mathfrak{B}}(U\cap S),$$

причем $U\cap S=\mathsf{C}^{\perp}_{U}(U)$ для всех $U\in\mathfrak{S}$ и, следовательно, $\mathfrak{B}=\{U\in\mathfrak{S}\,|\,\mathsf{C}^{\perp}_{U}(U)\neq\{0\}\}$. Пусть $W\in\mathfrak{B}$. Выберем некоторый элемент $w\in(W\cap S)\setminus\{0\}$ и какое-либо прямое дополнение W' к $\langle w\rangle$ в W. Так как $W=\langle w\rangle\oplus W'$ и $\langle w\rangle\perp W'$, из $\perp$-неразложимости W вытекает, что $W'=\{0\}$. Следовательно, $W=\langle w\rangle=W\cap S\subseteq S$. Это показывает, что

$\mathfrak{B} = \{U \in \mathfrak{S} \,|\, U \subseteq S\}$ и что $S = \bigoplus \mathfrak{B}$ — разложение пространства S в прямую сумму одномерных подпространств.

Очевидно, что R — прямое дополнение к $S = \bigoplus \mathfrak{B}$ в V. Так как $R = \bigoplus(\mathfrak{S} \setminus \mathfrak{B})$ — полное $\perp$-ортогональное разложение пространства R, из замечания 3.1 вытекает, что $V/S = \bigoplus \mathfrak{C}$ — полное $\perp$-ортогональное разложение пространства V/S. Следовательно, $(\mathfrak{B}, R, \mathfrak{C}) \in \mathcal{D}$. Кроме того, $\nu_{R,S}^{-1}(\mathfrak{C}) = \mathfrak{S} \setminus \mathfrak{B}$. Это показывает, что $(\mathfrak{B}, R, \mathfrak{C})$ является прообразом $\mathfrak{S}$ при отображении (8). Легко видеть (используя равенство (12)), что других прообразов $\mathfrak{S}$ при отображении (8) нет. Таким образом, отображение (8) биективно. ■

3.3. Одно условие для пространств с ортогональностями

В дальнейшем нам понадобится следующее условие для пространства с ортогональностью $(V; \perp)$:

$$\forall\, x, y \in V \;\; \big(\mathsf{C}_V^{\perp}(x) + \mathsf{C}_V^{\perp}(y) = V \implies \mathsf{C}_V^{\perp}(x+y) = \mathsf{C}_V^{\perp}(x) \cap \mathsf{C}_V^{\perp}(y)\big) \tag{13}$$

(см. также условие (2) из работы [1]). Очевидно, что если $\perp = V \times V$, то $(V; \perp)$ удовлетворяет условию (13).

Замечание 3.4. Пусть $(V; \perp)$ — декартово или прямое произведение семейства $((V_m; \perp_m) \,|\, m \in M)$ векторных пространств с ортогональностями. Тогда легко видеть, что для любого $v = (v_m \,|\, m \in M) \in V$ (где $v_m \in V_m$) пространство $\mathsf{C}_V^{\perp}(v)$ есть декартово или соответственно прямое произведение семейства $\big(\mathsf{C}_{V_m}^{\perp_m}(v_m) \,\big|\, m \in M\big)$. Из этого нетрудно получить, что класс всех векторных пространств с ортогональностями (над фиксированным полем F), удовлетворяющих условию (13), замкнут относительно взятия декартовых и прямых произведений.

Замечание 3.5. Пусть $V = U + W$ — $\perp$-ортогональное разложение пространства V. Тогда если $\mathsf{C}_U^{\perp}(x) + \mathsf{C}_U^{\perp}(y) = U$, где $x, y \in U$, то $\mathsf{C}_V^{\perp}(x) + \mathsf{C}_V^{\perp}(y) = V$. Из этого следует, что если $(V; \perp)$ удовлетворяет условию (13), то и $(U; \perp)$ удовлетворяет этому условию.

Замечание 3.6. Пусть W — произвольное подпространство $\mathsf{C}_V^{\perp}(V)$. Тогда, используя замечания 3.1, 3.4 и 3.5 и эквивалентность (4), легко показать, что $(V/W; \perp)$ удовлетворяет условию (13), если и только если $(V; \perp)$ удовлетворяет этому условию.

Лемма 3.3. Предположим, что $(V; \perp)$ удовлетворяет условию (13). Пусть $V = \sum\limits_{m \in M} V_m$ — $\perp$-ортогональное разложение пространства V. Тогда если $x_m, y_m \in V_m$ для всех $m \in M$, причем $\{m \in M \,|\, x_m \neq 0\}$ и $\{m \in M \,|\, y_m \neq 0\}$ конечны, то

$$\sum_{m \in M} x_m \perp \sum_{m \in M} y_m \iff \forall\, m \in M \;\; x_m \perp y_m.$$

Доказательство. Импликация «$\Longleftarrow$» очевидна. Пусть теперь $\sum\limits_{m \in M} x_m \perp \sum\limits_{m \in M} y_m$. Фиксируем $m \in M$ и положим $z = \sum\limits_{k \in M \setminus \{m\}} x_k$. Легко видеть, что $\mathsf{C}_V^{\perp}(x_m) + \mathsf{C}_V^{\perp}(z) = V$, так как $\mathsf{C}_V^{\perp}(x_m) \supseteq V_k$ для любого $k \in M \setminus \{m\}$ и $\mathsf{C}_V^{\perp}(z) \supseteq V_m$. Поэтому из условия (13) вытекает, что $\mathsf{C}_V^{\perp}\left(\sum\limits_{k \in M} x_k\right) \subseteq \mathsf{C}_V^{\perp}(x_m)$. В частности, $x_m \perp \sum\limits_{k \in M} y_k$. В то же время $x_m \perp \sum\limits_{k \in M \setminus \{m\}} y_k$. Следовательно, $x_m \perp y_m$. Импликация «$\Longrightarrow$» доказана. ■

Замечание 3.7. Предположим, что $(V; \perp)$ удовлетворяет условию (13). Пусть $V = \bigoplus\limits_{m \in M} U_m$ и $V = \bigoplus\limits_{n \in N} W_n$ — $\perp$-ортогональные разложения пространства V. Пусть

также μ — биекция M на N и α_m — изоморфизм $(U_m;\perp)$ на $(W_{\mu(m)};\perp)$ для каждого $m\in M$. Тогда, используя лемму 3.3, легко видеть, что отображение $\alpha\colon V\to V$, определенное равенством

$$\alpha\left(\sum_{m\in M} u_m\right)=\sum_{m\in M}\alpha_m(u_m),$$

где $u_m\in U_m$ и $\{m\in M\,|\,u_m\neq 0\}$ конечно, является автоморфизмом $(V;\perp)$. Очевидно также, что $\alpha(U_m)=W_{\mu(m)}$ при любом $m\in M$. Поэтому рассматриваемые разложения пространства V $\perp$-эквивалентны, если и только если они локально $\perp$-эквивалентны.

Приведем нужные нам результаты из работы [1]. Следуя указанной работе, назовем элемент $v\in V$ $\perp$-*неразложимым*, если он не может быть представлен в виде v_1+v_2, где $v_i\in V$ и $\mathsf{C}_V^{\perp}(v_i)\supset\mathsf{C}_V^{\perp}(v)$ при $i=1,2$. В следующих двух леммах через I обозначается множество всех ненулевых $\perp$-неразложимых элементов пространства V.

Лемма 3.4. Предположим, что $(V;\perp)$ удовлетворяет условию (13), причем $\mathsf{C}_V^{\perp}(V)=\{0\}$. Пусть $V=\bigoplus\limits_{m\in M}V_m$ — $\perp$-ортогональное разложение пространства V (всякое такое разложение является прямым ввиду замечания 3.3). Тогда $I\subseteq\bigcup\limits_{m\in M}V_m$ или, что эквивалентно, $I=\bigsqcup\limits_{m\in M}(I\cap V_m)$.

Лемма 3.5. Предположим, что $(V;\perp)$ удовлетворяет условию (13), причем V конечномерно и $\mathsf{C}_V^{\perp}(V)=\{0\}$. Пусть $\triangle$ — некоторая ортогональность на V, содержащаяся в $\perp$. Пусть также $\mathfrak{X}$ — множество всех связных компонент графа с множеством вершин I, в котором две различные вершины a и b смежны тогда и только тогда, когда они не $\triangle$-ортогональны. Тогда $V=\bigoplus\limits_{X\in\mathfrak{X}}\langle X\rangle$, причем это равенство дает единственное полное $\triangle$-ортогональное разложение пространства V.

Леммы 3.4 и 3.5 доказываются точно так же, как соответственно леммы 7 и 8 из работы [1].

4. Частичные симметричные билинейные функции на векторных пространствах, принимающие значения в основном поле

4.1. Определения, обозначения и необходимые факты

Пусть σ — функция из $\lozenge$ в F, где $\lozenge$ — некоторое подмножество $V\times V$. Тогда σ называется *частичной симметричной билинейной функцией* на V, если

$$a\lozenge b\implies b\lozenge a\ \&\ \sigma(b,a)=\sigma(a,b), \tag{14}$$

$$a_1\lozenge b\ \&\ a_2\lozenge b\implies a_1+a_2\lozenge b\ \&\ \sigma(a_1+a_2,b)=\sigma(a_1,b)+\sigma(a_2,b), \tag{15}$$

$$a\lozenge b\implies \lambda a\lozenge b\ \&\ \sigma(\lambda a,b)=\lambda\sigma(a,b) \tag{16}$$

для любых $a,a_1,a_2,b\in V$ и $\lambda\in F$. Очевидно, что из условий (14)–(16) следует линейность σ и по второму аргументу, т. е.

$$a\lozenge b_1\ \&\ a\lozenge b_2\implies a\lozenge b_1+b_2\ \&\ \sigma(a,b_1+b_2)=\sigma(a,b_1)+\sigma(a,b_2),$$
$$a\lozenge b\implies a\lozenge \lambda b\ \&\ \sigma(a,\lambda b)=\lambda\sigma(a,b)$$

при произвольных $a,b,b_1,b_2\in V$ и $\lambda\in F$. Альтернативным образом можно считать, что σ определена на всем $V\times V$, но принимает значения в $F\sqcup\{\infty\}$. В этом случае σ называется частичной симметричной билинейной функцией на V, если условия (14)–(16) выполняются для множества $\lozenge=\{(a,b)\,|\,a,b\in V,\,\sigma(a,b)\in F\}$. Очевидно,

что эти два подхода по сути эквивалентны. Все частичные симметричные билинейные функции, рассматриваемые в настоящей работе, предполагаются принимающими значения в основном поле F, если не оговорено противное.

Вообще говоря, область определения $\lozenge$ частичной симметричной билинейной функции σ может быть пустой. Но этот случай не представляет интереса. Если же $\lozenge \neq \varnothing$, то $V_0 = \{x \in V \mid 0 \lozenge x\}$ — подпространство V и $\lozenge \subseteq V_0 \times V_0$. Следовательно, в этом случае σ является по существу частичной симметричной билинейной функцией на V_0, причем $0 \lozenge V_0$.

В дальнейшем, если не оговорено противное, через σ будем обозначать некоторую частичную симметричную билинейную функцию на V с областью определения $\lozenge \subseteq \subseteq V \times V$. Кроме того, предполагаем, что $0 \lozenge V$; это предположение почти не ограничивает общность (см. предыдущий абзац). Функция σ может иметь специальный вид, если это указано явно. Алгебраическую систему $(V; \sigma)$, где σ рассматривается как частичный многозначный бинарный предикат, будем называть *(векторным) пространством с частичной симметричной билинейной функцией.*

Очевидно, что $\lozenge$ является ортогональностью на V. Определим бинарное отношение $\triangledown$ на V следующим образом:

$$a \triangledown b \iff a \lozenge b \ \&\ \sigma(a, b) = 0,$$

где $a, b \in V$. Тогда легко видеть, что $\triangledown$ — ортогональность на V. Положим для краткости

$$S_0 = \mathsf{C}_V^{\triangledown}(V), \quad S_1 = \mathsf{C}_V^{\lozenge}(V), \quad T = \mathsf{C}_{S_1}^{\triangledown}(S_1). \tag{17}$$

Очевидно, что $S_0 \subseteq T \subseteq S_1$.

Замечание 4.1. Легко видеть, что отображение, ставящее в соответствие произвольному смежному классу $a + S_0$, где $a \in S_1$, линейную функцию $v + T \mapsto \sigma(a, v)$ (где $v \in V$), определено корректно и является инъективным гомоморфизмом S_1/S_0 в векторное пространство всех линейных функций из V/T в F (т. е. в сопряженное пространство для V/T). Поэтому если V/T конечномерно, то и S_1/S_0 конечномерно, причем $\dim(S_1/S_0) \leqslant \dim(V/T)$. Следовательно, конечномерность V/T эквивалентна конечномерности V/S_0.

Обозначим через ρ естественный гомоморфизм V/S_0 на V/T, т. е. гомоморфизм $a + S_0 \mapsto a + T$, где $a \in V$. Из эквивалентности (4) следует, что

$$x \lozenge y \iff \rho(x) \lozenge \rho(y) \tag{18}$$

для любых $x, y \in V/S_0$. Нам потребуется также легко проверяемое равенство

$$\rho^{-1}(\langle X \rangle) = \big\langle \rho^{-1}(X) \big\rangle + (T/S_0), \tag{19}$$

в котором X — произвольное подмножество V/T.

Замечание 4.2. Пусть L — подпространство V/T и Q — произвольное прямое дополнение к T/S_0 в $\rho^{-1}(L)$. Тогда очевидно, что $Q \cong \rho^{-1}(L)/(T/S_0) \cong L$ и, следовательно, $\dim Q = \dim L$.

Если U — подпространство V, то очевидно, что σ является частичной симметричной билинейной функцией на U с областью определения $\lozenge_U = \lozenge \cap (U \times U)$, причем $0 \lozenge_U U$. Пусть теперь W — подпространство S_0. Тогда легко видеть, что формула

$$(a + W, b + W) \mapsto \sigma(a, b), \quad a, b \in V,\ a \lozenge b,$$

корректно определяет частичную симметричную билинейную функцию на V/W с областью определения

$$\{(a+W, b+W) \,|\, a, b \in V,\ a \Diamond b\} = \{(x, y) \,|\, x, y \in V/W,\ x \Diamond y\}$$

(см. эквивалентность (4)), причем $0 \Diamond V/W$. Эту частичную симметричную билинейную функцию мы обозначаем так же, как и исходную, т. е. через σ.

Замечание 4.3. Пусть W — подпространство S_0 и R — произвольное прямое дополнение к W в V. Тогда $\nu_{R,W}$ является не только изоморфизмом $(R; \triangledown)$ на $(V/W; \triangledown)$ и $(R; \Diamond)$ на $(V/W; \Diamond)$ (см. замечание 3.1), но и изоморфизмом $(R; \sigma)$ на $(V/W; \sigma)$.

Предположим, что каждому $m \in M$, где M — некоторое множество, поставлены в соответствие векторное пространство V_m и частичная симметричная билинейная функция σ_m на V_m с областью определения $\Diamond_m \subseteq V_m \times V_m$. Пусть $(V; \Diamond)$ — прямое произведение семейства $((V_m; \Diamond_m) \,|\, m \in M)$ векторных пространств с ортогональностями. Определим функцию $\sigma\colon \Diamond \to F$ равенством

$$\sigma((a_m \,|\, m \in M), (b_m \,|\, m \in M)) = \sum_{m \in M} \sigma_m(a_m, b_m),$$

где $a_m, b_m \in V_m$ при любом $m \in M$, $(a_m \,|\, m \in M), (b_m \,|\, m \in M) \in V$ (т. е. $\{m \in M \,|\, a_m \neq \neq 0\}$ и $\{m \in M \,|\, b_m \neq 0\}$ конечны) и $(a_m \,|\, m \in M) \Diamond (b_m \,|\, m \in M)$ (указанная сумма определена, так как множество $\{m \in M \,|\, \sigma_m(a_m, b_m) \neq 0\}$ конечно). Тогда непосредственно проверяется, что σ — частичная симметричная билинейная функция на V с областью определения $\Diamond$, причем если $0 \Diamond V_m$ для всех $m \in M$, то $0 \Diamond V$. Будем называть $(V; \sigma)$ *прямым ортогональным произведением* семейства $((V_m; \sigma_m) \,|\, m \in M)$ пространств с частичными симметричными билинейными функциями.

Пусть снова σ — произвольная частичная симметричная билинейная функция на некотором векторном пространстве V с областью определения $\Diamond$, причем $0 \Diamond V$. Следующая лемма является аналогом замечания 3.7 и доказывается аналогично ему.

Лемма 4.1. Предположим, что $(V; \Diamond)$ удовлетворяет условию (13). Пусть $V = \bigoplus\limits_{m \in M} U_m$ и $V = \bigoplus\limits_{n \in N} W_n$ — $\triangledown$-ортогональные разложения пространства V. Тогда

1) если μ — биекция M на N и α_m — изоморфизм $(U_m; \sigma)$ на $(W_{\mu(m)}; \sigma)$ для каждого $m \in M$, то отображение $\alpha\colon V \to V$, определенное равенством

$$\alpha\left(\sum_{m \in M} u_m\right) = \sum_{m \in M} \alpha_m(u_m),$$

где $u_m \in U_m$ и $\{m \in M \,|\, u_m \neq 0\}$ конечно, является автоморфизмом $(V; \sigma)$;

2) рассматриваемые разложения пространства V σ-эквивалентны, если и только если они локально σ-эквивалентны.

4.2. Пространства типов 1 и 2

Бóльшая часть приводимого здесь материала известна в линейной алгебре, хотя и в других терминах.

Назовем $(V; \sigma)$ или, короче, V пространством *типа 1*, если $V = \langle a \rangle$ для некоторого $a \in V$, такого, что $a \Diamond a$ (и, следовательно, $V \Diamond V$, т. е. $\Diamond = V \times V$) и $a \not\perp a$ (или, что эквивалентно, $\sigma(a, a) \neq 0$). Кроме того, $(V; \sigma)$ или просто V будет называться пространством *типа 2*, если $\operatorname{char} F = 2$, $V \Diamond V$ и $V = \langle a, b \rangle$ для некоторых $a, b \in V$, удовлетворяющих равенствам $\sigma(a, a) = \sigma(b, b) = 0$ и $\sigma(a, b) = 1$ (из которых следует,

что a и b линейно независимы). Другими словами, $(V;\sigma)$ имеет тип 2, если и только если $\operatorname{char} F = 2$, $\lozenge = V \times V$ и матрица билинейной функции σ в некотором базисе пространства V имеет вид $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Очевидно, что если $(V;\sigma)$ — пространство типа 1 или типа 2, то билинейная функция σ невырождена на V (т. е. $\mathsf{C}_V^{\triangledown}(V) = \{0\}$).

Пусть $(V;\sigma)$ — пространство типа 1. Поставим ему в соответствие смежный класс $\varkappa(V;\sigma) \in F^*/(F^*)^2$ следующим образом:

$$\varkappa(V;\sigma) = \sigma(v,v)(F^*)^2,$$

где v — произвольный элемент $V \setminus \{0\}$; очевидно, что это определение корректно.

Замечание 4.4. Пусть $(V;\sigma)$ и $(V';\sigma')$ — пространства типа 1. Тогда легко видеть, что $(V;\sigma) \cong (V';\sigma')$, если и только если $\varkappa(V;\sigma) = \varkappa(V';\sigma')$.

Если пространство V имеет тип 1, то его $\triangledown$-неразложимость тривиальна. Следующая лемма утверждает $\triangledown$-неразложимость пространств типа 2.

Лемма 4.2. Если V — пространство типа 2, то оно $\triangledown$-неразложимо.

Доказательство. Предположим, что V — $\triangledown$-разложимое пространство типа 2. Тогда $V = \langle u\rangle \oplus \langle w\rangle$ для некоторых $u, w \in V \setminus \{0\}$, таких, что $u \triangledown w$. Кроме того, $u \triangledown u$ и $w \triangledown w$, так как $v \triangledown v$ для всех $v \in V$. Следовательно, $V \triangledown V$, что неверно. ■

Лемма 4.3. Пусть $\lozenge = V \times V$ и $V \not\triangledown V$. Тогда V содержит подпространство типа 1 или типа 2.

Доказательство. Предположим сначала, что существует $a \in V$, для которого $a \not\triangledown a$. Тогда $\langle a\rangle$ — подпространство типа 1 пространства V. Считаем теперь, что

$$\forall\, x \in V \;\; x \triangledown x. \tag{20}$$

Это возможно только при $\operatorname{char} F = 2$, так как в противном случае из условия (20) ввиду того, что

$$\forall x, y \in V \,(\, \sigma(x,y) = (\sigma(x+y, x+y) - \sigma(x,x) - \sigma(y,y))/2 \,),$$

следует, что $V \triangledown V$, вопреки условию леммы. Выберем некоторые $a, b \in V$, такие, что $a \not\triangledown b$ (возможность такого выбора вытекает из условия $V \not\triangledown V$). Заменив a на $(1/\sigma(a,b))a$, можно добиться выполнения равенства $\sigma(a,b) = 1$. Кроме того, $\sigma(a,a) = \sigma(b,b) = 0$ ввиду условия (20). Таким образом, в данном случае $\langle a, b\rangle$ является подпространством типа 2 пространства V. ■

4.3. $\triangledown$-ортогональные дополнения к конечномерным подпространствам S_1

Первое утверждение следующей леммы хорошо известно в случае, когда V конечномерно и $\lozenge = V \times V$.

Лемма 4.4. Пусть U — произвольное конечномерное подпространство S_1, такое, что билинейная функция σ невырождена на U (другими словами, $\mathsf{C}_U^{\triangledown}(U) = \{0\}$). Тогда $V = U \oplus \mathsf{C}_V^{\triangledown}(U)$. Кроме того, если $V = U + W$ — $\triangledown$-ортогональное разложение пространства V, то $W = \mathsf{C}_V^{\triangledown}(U)$.

Доказательство. Пусть $v \in V$. Тогда $x \mapsto \sigma(v,x)$, где $x \in U$, является линейной функцией из U в F. Ввиду конечномерности U и невырожденности σ на U всякая такая функция имеет вид $x \mapsto \sigma(u,x)$ (где $x \in U$) для некоторого $u \in U$. Следовательно,

можно выбрать $u \in U$, удовлетворяющий равенству $\sigma(v,x) = \sigma(u,x)$ для любого $x \in U$. Тогда $v = u + (v-u) \in U + \mathsf{C}_V^\triangledown(U)$, так как $v - u \triangledown U$. Таким образом, $V = U + \mathsf{C}_V^\triangledown(U)$, причем эта сумма является прямой ввиду того, что $U \cap \mathsf{C}_V^\triangledown(U) = \mathsf{C}_U^\triangledown(U) = \{0\}$.

Предположим теперь, что $V = U + W$ — $\triangledown$-ортогональное разложение пространства V. Тогда равенство $W = \mathsf{C}_V^\triangledown(U)$ легко вытекает из очевидного включения $W \subseteq \mathsf{C}_V^\triangledown(U)$ и уже доказанного равенства $V = U \oplus \mathsf{C}_V^\triangledown(U)$. ■

Лемма 4.5.

1. Пусть $\operatorname{char} F \neq 2$, $u \in S_1$ и $t \in T$, причем $u \not\triangledown u$. Тогда $(\mathsf{C}_V^\triangledown(u);\sigma) \cong (\mathsf{C}_V^\triangledown(u+t);\sigma)$.
2. Пусть $u_1, u_2 \in S_1$ и $t_1, t_2 \in T$, причем $\sigma(u_1,u_1) = \sigma(u_2,u_2) = 0$ и $\sigma(u_1,u_2) = 1$. Тогда $(\mathsf{C}_V^\triangledown(u_1,u_2);\sigma) \cong (\mathsf{C}_V^\triangledown(u_1+t_1,u_2+t_2);\sigma)$.

Доказательство. Докажем сначала утверждение 1. Легко видеть, что отображение

$$x \mapsto x - \frac{\sigma(x,t)}{\sigma(u,u)}\left(\frac{1}{2}t + u\right), \quad x \in \mathsf{C}_V^\triangledown(u),$$

является изоморфизмом $(\mathsf{C}_V^\triangledown(u);\sigma)$ на $(\mathsf{C}_V^\triangledown(u+t);\sigma)$ (обратным к этому отображению является $y \mapsto y + (\sigma(y,t)/\sigma(u,u))((1/2)t+u)$, где $y \in \mathsf{C}_V^\triangledown(u+t)$).

Утверждение 2 доказывается аналогично, а именно непосредственно проверяется, что отображение

$$x \mapsto x - \sigma(x,t_1)(u_2+t_2) - \sigma(x,t_2)u_1, \quad x \in \mathsf{C}_V^\triangledown(u_1,u_2),$$

является изоморфизмом $(\mathsf{C}_V^\triangledown(u_1,u_2);\sigma)$ на $(\mathsf{C}_V^\triangledown(u_1+t_1,u_2+t_2);\sigma)$ (обратным к этому отображению является $y \mapsto y + \sigma(y,t_1)(u_2+t_2) + \sigma(y,t_2)u_1$, где $y \in \mathsf{C}_V^\triangledown(u_1+t_1,u_2+t_2)$). Лемма доказана. ■

Лемма 4.6. Предположим, что $(V;\lozenge)$ удовлетворяет условию (13) и S_1/T конечномерно. Пусть также $x \triangledown x$ для всех $x \in S_1$ в случае, когда $\operatorname{char} F = 2$. Тогда если U и W — прямые дополнения к T в S_1, то $(\mathsf{C}_V^\triangledown(U);\sigma) \cong (\mathsf{C}_V^\triangledown(W);\sigma)$.

Доказательство. Используем индукцию по $\dim(S_1/T) = \dim U = \dim W$. Если $\dim(S_1/T) = 0$, то $U = W = \{0\}$ и утверждение леммы тривиально. Пусть теперь $\dim(S_1/T) \geqslant 1$. Очевидно, что $U \not\triangledown U$, поэтому ввиду леммы 4.3 существует подпространство $U_0 \subseteq U$, имеющее тип 1, если $\operatorname{char} F \neq 2$, или тип 2, если $\operatorname{char} F = 2$ (в последнем случае U не содержит подпространств типа 1, так как $x \triangledown x$ для всех $x \in U$). Выберем базис $u_1, \ldots, u_k$ пространства U_0 (здесь $k \in \{1,2\}$), такой, что если $\operatorname{char} F = 2$, то σ в этом базисе имеет матрицу $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Пусть также $t_1, \ldots, t_k$ — некоторые элементы T, для которых $u_1 + t_1, \ldots, u_k + t_k \in W$. Тогда $u_1 + t_1, \ldots, u_k + t_k$ линейно независимы. Положим $W_0 = \langle u_1 + t_1, \ldots, u_k + t_k \rangle$. Очевидно, что σ в базисе $u_1 + t_1, \ldots, u_k + t_k$ имеет ту же матрицу, что и в базисе $u_1, \ldots, u_k$. Это показывает, что $(U_0;\sigma) \cong (W_0;\sigma)$. Кроме того, из леммы 4.5 вытекает, что $(\mathsf{C}_V^\triangledown(U_0);\sigma) \cong (\mathsf{C}_V^\triangledown(W_0);\sigma)$. Поэтому ввиду леммы 4.4 и утверждения 1 леммы 4.1 можно выбрать $\alpha \in \operatorname{Aut}(V;\sigma)$, для которого $\alpha(U_0) = W_0$ и $\alpha\left(\mathsf{C}_V^\triangledown(U_0)\right) = \mathsf{C}_V^\triangledown(W_0)$.

Положим для краткости $V' = \mathsf{C}_V^\triangledown(W_0)$ и $S_1' = \mathsf{C}_{V'}^\lozenge(V')$. Тогда $V = W_0 \oplus V'$ по лемме 4.4. Следовательно, $S_1 = W_0 \oplus S_1'$ и $T = \mathsf{C}_{S_1'}^\triangledown(S_1')$ ввиду леммы 3.1, так как $\mathsf{C}_{W_0}^\lozenge(W_0) = W_0$ и $\mathsf{C}_{W_0}^\triangledown(W_0) = \{0\}$. С другой стороны, из леммы 4.4 вытекает, что

$$S_1 = W \oplus T = W_0 \oplus \mathsf{C}_W^\triangledown(W_0) \oplus T,$$

где $\mathsf{C}^{\triangledown}_W(W_0)\oplus T\subseteq S_1\cap V'=S'_1$ (см. лемму 3.1). Поэтому $\mathsf{C}^{\triangledown}_W(W_0)\oplus T=S'_1$. Аналогично,

$$\mathsf{C}^{\triangledown}_U(U_0)\oplus T=\mathsf{C}^{\Diamond}_{\mathsf{C}^{\triangledown}_V(U_0)}\left(\mathsf{C}^{\triangledown}_V(U_0)\right).$$

Применив к обеим частям последнего равенства автоморфизм α и воспользовавшись очевидным равенством $\alpha(T)=T$, получим, что $\mathsf{C}^{\triangledown}_{\alpha(U)}(W_0)\oplus T=S'_1$. По предположению индукции, примененному к пространству V' и прямым дополнениям $\mathsf{C}^{\triangledown}_{\alpha(U)}(W_0)$ и $\mathsf{C}^{\triangledown}_W(W_0)$ к подпространству T в S'_1, имеет место изоморфизм

$$\left(\mathsf{C}^{\triangledown}_{V'}\left(\mathsf{C}^{\triangledown}_{\alpha(U)}(W_0)\right);\sigma\right)\cong\left(\mathsf{C}^{\triangledown}_{V'}\left(\mathsf{C}^{\triangledown}_W(W_0)\right);\sigma\right). \tag{21}$$

Здесь применение предположения индукции возможно, так как $\dim(S'_1/T)=\dim(S_1/T)-k$. Кроме того, $\alpha(U)=W_0\oplus\mathsf{C}^{\triangledown}_{\alpha(U)}(W_0)$ (так как $U=U_0\oplus\mathsf{C}^{\triangledown}_U(U_0)$ по лемме 4.4) и, следовательно,

$$\mathsf{C}^{\triangledown}_V(\alpha(U))=V'\cap\mathsf{C}^{\triangledown}_V\left(\mathsf{C}^{\triangledown}_{\alpha(U)}(W_0)\right)=\mathsf{C}^{\triangledown}_{V'}\left(\mathsf{C}^{\triangledown}_{\alpha(U)}(W_0)\right). \tag{22}$$

Аналогично, $W=W_0\oplus\mathsf{C}^{\triangledown}_W(W_0)$ и

$$\mathsf{C}^{\triangledown}_V(W)=V'\cap\mathsf{C}^{\triangledown}_V\left(\mathsf{C}^{\triangledown}_W(W_0)\right)=\mathsf{C}^{\triangledown}_{V'}\left(\mathsf{C}^{\triangledown}_W(W_0)\right). \tag{23}$$

Утверждение леммы следует теперь из (21)–(23) и очевидного изоморфизма $(\mathsf{C}^{\triangledown}_V(U);\sigma)\cong(\mathsf{C}^{\triangledown}_V(\alpha(U));\sigma)$. ■

4.4. Некоторые $\triangledown$-ортогональные разложения пространства V в случае, когда σ всюду определена и V/S_0 конечномерно

Будем считать, что σ определена на всем $V\times V$ (т. е. $\Diamond=V\times V$) и пространство V/S_0 конечномерно.

Пусть R — какое-либо прямое дополнение к S_0 в V. Тогда $(R;\sigma)\cong(V/S_0;\sigma)$ (см. замечание 4.3) и, следовательно, R конечномерно и σ невырождена на R. Применив к R известные результаты линейной алгебры (см. теоремы IV.6 и IV.7 из [4]), получим следующую лемму.

Лемма 4.7. Существует $\triangledown$-ортогональное разложение

$$V=\left(\bigoplus\mathfrak{A}\right)\oplus S_0 \tag{24}$$

пространства V, в котором множество $\mathfrak{A}$ состоит из подпространств типов 1 и 2.

Очевидно, что множество $\mathfrak{A}$ в любом разложении вида (24) конечно.

Замечание 4.5. Разложение (24) из леммы 4.7 строится с помощью следующего хорошо известного процесса:

1) положить $\mathfrak{A}=\varnothing$ и $U=V$;
2) если $U\triangledown U$, то закончить процесс;
3) если $U\not\triangledown U$, то выбрать некоторое подпространство $W\subseteq U$ типа 1 или 2 (это возможно ввиду леммы 4.3), добавить W к множеству $\mathfrak{A}$, заменить U на $\mathsf{C}^{\triangledown}_U(W)$ и перейти к шагу 2.

Из леммы 4.4 вытекает, что перед каждым выполнением шага 2 имеет место равенство

$$V=\left(\bigoplus\mathfrak{A}\right)\oplus U, \tag{25}$$

дающее ∇-ортогональное разложение пространства V. При этом $S_0 \subseteq U$ и $\dim(U/S_0)$ после каждого выполнения шага 3 уменьшается на 1 или 2. Следовательно, шаг 3 может выполниться не более $\dim(V/S_0)$ раз, после чего процесс закончится на шаге 2. После окончания процесса (на шаге 2) получим, что $U = S_0$ (ввиду равенства (25) и леммы 3.1), поэтому (25) дает требуемое разложение. Кроме того, легко видеть (используя лемму 4.4), что указанный процесс позволяет построить все разложения (24), удовлетворяющие условиям леммы 4.7.

Выясним теперь, когда разложение (24) из леммы 4.7 единственно.

Лемма 4.8. Разложение (24) пространства V, удовлетворяющее условиям леммы 4.7, является единственным разложением такого вида тогда и только тогда, когда выполнено хотя бы одно из следующих условий:

1) $\mathfrak{A} = \varnothing$ (т. е. $V = S_0$);
2) $S_0 = \{0\}$ и всякое подпространство пространства V, имеющее тип 1 или 2, принадлежит $\mathfrak{A}$.

Доказательство. Пусть указанное разложение единственно и $\mathfrak{A} \neq \varnothing$. Предположим, что $S_0 \neq \{0\}$, и выберем некоторый элемент $s \in S_0 \setminus \{0\}$, а также некоторое подпространство $W \in \mathfrak{A}$ и представим его в виде $\langle w\rangle \oplus W_0$, где $w \in W \setminus \{0\}$ и W_0 — подпространство W. Тогда $V = (\langle w + s\rangle \oplus W_0) \oplus (\bigoplus(\mathfrak{A} \setminus \{W\})) \oplus S_0$ — разложение пространства V, удовлетворяющее условиям леммы 4.7 и отличное от исходного. Полученное противоречие показывает, что $S_0 = \{0\}$.

Пусть подпространство $U \subseteq V$ имеет тип 1 или 2. По лемме 4.4 $V = U \oplus \mathsf{C}_V^{\nabla}(U)$, так как σ невырождена на U. Из леммы 4.7 вытекает, что существует ∇-ортогональное разложение $\mathsf{C}_V^{\nabla}(U) = \bigoplus \mathfrak{A}'$ подпространства $\mathsf{C}_V^{\nabla}(U)$, состоящее из подпространств типов 1 и 2 (очевидно, что σ невырождена на $\mathsf{C}_V^{\nabla}(U)$). Тогда разложение $V = U \oplus (\bigoplus \mathfrak{A}')$ пространства V удовлетворяет условиям леммы 4.7. Следовательно, $U \in \mathfrak{A}$. Таким образом, выполнено условие 2. Обратно, если выполнено условие 1 или условие 2, то очевидно, что рассматриваемое разложение единственно. ■

5. Полные ∇-ортогональные разложения пространства V

Предположим, что пространство с ортогональностью $(V; \lozenge)$ удовлетворяет условию (13). Опишем полные ∇-ортогональные разложения пространства V при условии конечномерности пространства V/T или, что эквивалентно, пространства V/S_0 (см. замечание 4.1). Это будет сделано в два этапа. Сначала (в п. 5.1), предполагая, что пространство S_1/T конечномерно, сведём задачу описания полных ∇-ортогональных разложений пространства V к задаче описания таких разложений пространств некоторого специального вида (теорема 5.1). Затем (в п. 5.2) решим последнюю из указанных задач в предположении конечномерности пространства V/T (леммы 5.12 и 5.13). Основной результат, описывающий полные ∇-ортогональные разложения пространства V, приводится в п. 5.3 (теорема 5.2); при этом нужно предполагать конечномерность V/T.

5.1. Сведение задачи описания полных ∇-ортогональных разложений пространства V к некоторому частному случаю

Пусть пространство S_1/T конечномерно. Обозначим через $\mathcal{A}$ множество всех множеств $\mathfrak{A}$, состоящих из подпространств S_1 типов 1 и 2 и таких, что

$$S_1 = (\bigoplus \mathfrak{A}) \oplus T$$

дает ∇-ортогональное разложение пространства S_1. Очевидно, что все множества из $\mathcal{A}$ конечны. По лемме 4.7, примененной к пространству S_1, множество $\mathcal{A}$ непусто. Для произвольного $\mathfrak{A} \in \mathcal{A}$ положим

$$G(\mathfrak{A}) = \mathsf{C}_V^{\nabla}\left(\bigoplus \mathfrak{A}\right).$$

Тогда ввиду леммы 4.4

$$V = \left(\bigoplus \mathfrak{A}\right) \oplus G(\mathfrak{A}), \tag{26}$$

так как $\bigoplus \mathfrak{A}$ конечномерно и билинейная функция σ невырождена на $\bigoplus \mathfrak{A}$. Кроме того, используя равенство (26) и лемму 3.1, легко видеть, что

$$S_0 = \mathsf{C}_{G(\mathfrak{A})}^{\nabla}(G(\mathfrak{A})) \quad \text{и} \quad T = S_1 \cap G(\mathfrak{A}) = \mathsf{C}_{G(\mathfrak{A})}^{\diamondsuit}(V) = \mathsf{C}_{G(\mathfrak{A})}^{\diamondsuit}(G(\mathfrak{A})) \tag{27}$$

для любого $\mathfrak{A} \in \mathcal{A}$.

Через $\mathcal{B}$ обозначим множество всех множеств $\mathfrak{B}$, состоящих из одномерных подпространств S_0 и таких, что $S_0 = \bigoplus \mathfrak{B}$. Если U — подпространство V, содержащее S_0, то через $\mathfrak{R}(U)$ будет обозначаться множество всех прямых дополнений к S_0 в U. Отметим, что $\mathcal{B} \neq \varnothing$ и $\mathfrak{R}(U) \neq \varnothing$ для любого подпространства $U \subseteq V$, содержащего S_0. Обозначим через $\mathcal{P}$ множество всех четверок вида $(\mathfrak{A}, \mathfrak{B}, R, \mathfrak{C})$, в которых $\mathfrak{A} \in \mathcal{A}$, $\mathfrak{B} \in \mathcal{B}$, $R \in \mathfrak{R}(G(\mathfrak{A}))$, а $\mathfrak{C}$ — такое множество подпространств $G(\mathfrak{A})/S_0$, что $G(\mathfrak{A})/S_0 = \bigoplus \mathfrak{C}$ является полным ∇-ортогональным разложением пространства $G(\mathfrak{A})/S_0$.

Лемма 5.1. Пусть $V = \bigoplus \mathfrak{S}$ — полное ∇-ортогональное разложение пространства V. Тогда существует единственное подмножество $\mathfrak{A} \subseteq \mathfrak{S}$, принадлежащее $\mathcal{A}$. Это подмножество $\mathfrak{A}$ определяется формулами

$$\mathfrak{A} = \left\{U \in \mathfrak{S} \;\middle|\; \mathsf{C}_U^{\diamondsuit}(U) \not\nabla \mathsf{C}_U^{\diamondsuit}(U)\right\} = \left\{U \in \mathfrak{S} \,|\, U \subseteq S_1,\ \mathsf{C}_U^{\nabla}(U) = \{0\}\right\}$$

и удовлетворяет равенству $G(\mathfrak{A}) = \bigoplus(\mathfrak{S} \setminus \mathfrak{A})$.

Доказательство. Положим $\mathfrak{A} = \left\{U \in \mathfrak{S} \;\middle|\; \mathsf{C}_U^{\diamondsuit}(U) \not\nabla \mathsf{C}_U^{\diamondsuit}(U)\right\}$. Рассмотрим произвольное подпространство $U \in \mathfrak{A}$. Выберем подпространство $W \subseteq \mathsf{C}_U^{\diamondsuit}(U)$, имеющее тип 1 или 2 (это возможно по лемме 4.3). Тогда $U = W \oplus \mathsf{C}_U^{\nabla}(W)$ согласно лемме 4.4. Ввиду ∇-неразложимости U получаем, что $U = W = \mathsf{C}_U^{\diamondsuit}(U)$ и, следовательно, $U = S_1 \cap U \subseteq S_1$ (см. лемму 3.1) и $\mathsf{C}_U^{\nabla}(U) = \mathsf{C}_W^{\nabla}(W) = \{0\}$. Это показывает, что $\mathfrak{A}$ состоит из подпространств типов 1 и 2 и содержится в $\left\{U \in \mathfrak{S} \,|\, U \subseteq S_1,\ \mathsf{C}_U^{\nabla}(U) = \{0\}\right\}$. Пусть теперь U — произвольный элемент последнего множества. Тогда $U = \mathsf{C}_U^{\diamondsuit}(U)$ (так как $U \subseteq S_1$) и $U \not\nabla U$ (в противном случае $U = \mathsf{C}_U^{\nabla}(U) = \{0\}$, а это несовместимо с вхождением U в $\mathfrak{S}$). Поэтому $U \in \mathfrak{A}$. Таким образом,

$$\mathfrak{A} = \left\{U \in \mathfrak{S} \,|\, U \subseteq S_1,\ \mathsf{C}_U^{\nabla}(U) = \{0\}\right\}. \tag{28}$$

Из леммы 3.1 и доказанного выше равенства $\mathsf{C}_U^{\diamondsuit}(U) = U$ для любого $U \in \mathfrak{A}$ вытекает, что

$$S_1 = \left(\bigoplus \mathfrak{A}\right) \oplus T', \tag{29}$$

где

$$T' = \bigoplus \left\{\mathsf{C}_U^{\diamondsuit}(U) \;\middle|\; U \in \mathfrak{S},\ \mathsf{C}_U^{\diamondsuit}(U) \nabla \mathsf{C}_U^{\diamondsuit}(U)\right\}.$$

Применив к разложению (29) пространства S_1 лемму 3.1 и воспользовавшись равенством (28), получим равенство $T = T'$. Поэтому из равенства (29) следует, что $\mathfrak{A} \in \mathcal{A}$ (как мы уже видели выше, $\mathfrak{A}$ состоит из подпространств типов 1 и 2).

Пусть теперь подмножество $\mathfrak{A}' \subseteq \mathfrak{S}$ принадлежит $\mathcal{A}$. Тогда $\mathfrak{A}' \subseteq \mathfrak{A}$ ввиду равенства (28). Поэтому $\mathfrak{A}' = \mathfrak{A}$, так как $S_1 = \left(\bigoplus \mathfrak{A}'\right) \oplus T = \left(\bigoplus \mathfrak{A}\right) \oplus T$. Равенство $G(\mathfrak{A}) = \bigoplus(\mathfrak{S} \setminus \mathfrak{A})$ непосредственно вытекает из леммы 4.4. ■

Теорема 5.1. Отображение

$$(\mathfrak{A},\mathfrak{B},R,\mathfrak{C})\mapsto \mathfrak{A}\sqcup\mathfrak{B}\sqcup\nu^{-1}_{R,S_0}(\mathfrak{C}),\quad (\mathfrak{A},\mathfrak{B},R,\mathfrak{C})\in\mathcal{P},\tag{30}$$

является биекцией $\mathcal{P}$ на множество всех $\mathfrak{S}$, таких, что $V=\bigoplus\mathfrak{S}$ — полное $\triangledown$-ортогональное разложение пространства V. Если $\mathfrak{S}$ — образ четверки $(\mathfrak{A},\mathfrak{B},R,\mathfrak{C})\in\mathcal{P}$ при отображении (30), то эти $\mathfrak{A}$, $\mathfrak{B}$, R и $\mathfrak{C}$ могут быть найдены по $\mathfrak{S}$ следующим образом:

$$\begin{aligned}
&\mathfrak{A}=\big\{U\in\mathfrak{S}\;\big|\;\mathsf{C}^{\diamondsuit}_U(U)\not\Subset\mathsf{C}^{\diamondsuit}_U(U)\big\}=\{U\in\mathfrak{S}\,|\,U\subseteq S_1,\,\mathsf{C}^{\triangledown}_U(U)=\{0\}\}\,;\\
&\mathfrak{B}=\{U\in\mathfrak{S}\,|\,U\cap S_0\neq\{0\}\}=\{U\in\mathfrak{S}\,|\,\mathsf{C}^{\triangledown}_U(U)\neq\{0\}\}=\{U\in\mathfrak{S}\,|\,U\subseteq S_0\};\\
&R=\bigoplus(\mathfrak{S}\setminus(\mathfrak{A}\sqcup\mathfrak{B}))=\bigoplus\big\{U\in\mathfrak{S}\;\big|\;\mathsf{C}^{\diamondsuit}_U(U)\triangledown\mathsf{C}^{\diamondsuit}_U(U),\,\mathsf{C}^{\triangledown}_U(U)=\{0\}\big\}\,;\\
&\mathfrak{C}=\nu_{R,S_0}(\mathfrak{S}\setminus(\mathfrak{A}\sqcup\mathfrak{B})).
\end{aligned}$$

Доказательство. Из равенства (26) и лемм 4.2 и 3.2 следует, что если $\mathfrak{S}$ — образ произвольной четверки $(\mathfrak{A},\mathfrak{B},R,\mathfrak{C})\in\mathcal{P}$ при отображении (30), то $V=\bigoplus\mathfrak{S}$ является полным $\triangledown$-ортогональным разложением пространства V. Пусть теперь $V=\bigoplus\mathfrak{S}$ — полное $\triangledown$-ортогональное разложение пространства V. Обозначим через $\mathfrak{A}$ подмножество $\mathfrak{S}$, принадлежащее $\mathcal{A}$; согласно лемме 5.1, такое подмножество существует, единственно и определяется формулами, указанными в формулировке теоремы. Кроме того, $G(\mathfrak{A})=\bigoplus(\mathfrak{S}\setminus\mathfrak{A})$ по той же лемме 5.1; очевидно, что это равенство дает полное $\triangledown$-ортогональное разложение пространства $G(\mathfrak{A})$. Ввиду леммы 3.2 (примененной к пространству $G(\mathfrak{A})$) и равенства для S_0 в (27) множество $\mathfrak{B}$, пространство R и множество $\mathfrak{C}$, такие, что

$$(\mathfrak{A},\mathfrak{B},R,\mathfrak{C})\in\mathcal{P}\;\;\&\;\;\mathfrak{S}=\mathfrak{A}\sqcup\mathfrak{B}\sqcup\nu^{-1}_{R,S_0}(\mathfrak{C}),\tag{31}$$

существуют, единственны и определяются формулами, указанными в формулировке теоремы. Здесь мы воспользовались тем, что условие (31) выполняется тогда и только тогда, когда $\mathfrak{B}\in\mathcal{B}$, $R\in\mathfrak{R}(G(\mathfrak{A}))$, $G(\mathfrak{A})/S_0=\bigoplus\mathfrak{C}$ — полное $\triangledown$-ортогональное разложение пространства $G(\mathfrak{A})/S_0$ и $\mathfrak{S}\setminus\mathfrak{A}=\mathfrak{B}\sqcup\nu^{-1}_{R,S_0}(\mathfrak{C})$. Таким образом, $\mathfrak{S}$ имеет единственный прообраз при отображении (30), причем компоненты этого прообраза могут быть найдены по формулам, указанным в формулировке теоремы. ■

5.2. Полные $\triangledown$-ортогональные разложения пространства $G(\mathfrak{A})/S_0$ для $\mathfrak{A}\in\mathcal{A}$

Предположим, что пространство V/T конечномерно. Согласно замечанию 4.1, это предположение эквивалентно предположению о конечномерности пространства V/S_0. При указанном предположении пространство S_1/T также конечномерно, поэтому здесь применимы результаты п. 5.1.

Фиксируем произвольное множество $\mathfrak{A}\in\mathcal{A}$ и полагаем $G=G(\mathfrak{A})=\mathsf{C}^{\triangledown}_V(\bigoplus\mathfrak{A})$ (см. п. 5.1). Многие объекты, определяемые и рассматриваемые в данном пункте, зависят от G. Однако здесь не будем явно указывать на их связь с подпространством G, так как последнее фиксировано. Некоторые из этих объектов понадобятся и далее, где будем указывать соответствующее подпространство G после объекта в круглых скобках (например, $\mathcal{K}(G)$).

Очевидно, что $S_0\subseteq T\subseteq G$. Из (27) и (6) вытекает, что

$$\mathsf{C}^{\triangledown}_{G/S_0}(G/S_0)=\{0\}\quad\text{и}\quad\mathsf{C}^{\diamondsuit}_{G/T}(G/T)=\{0\}.\tag{32}$$

Поэтому все $\triangledown$-ортогональные разложения пространства G/S_0 и $\lozenge$-ортогональные разложения пространства G/T являются прямыми ввиду замечания 3.3. Кроме того,

$$T/S_0 \triangledown T/S_0, \tag{33}$$

так как $T \triangledown T$.

Замечание 5.1. Из равенства $T = S_1 \cap G$ (см. (27)) следует, что отображение $g + T \mapsto g + S_1$, где $g \in G$, является изоморфизмом G/T на $(G + S_1)/S_1 = V/S_1$ (равенство $G + S_1 = V$ вытекает из (26)). Более того, ввиду эквивалентности (4) это отображение представляет собой изоморфизм $(G/T; \lozenge)$ на $(V/S_1; \lozenge)$. Таким образом, $(G/T; \lozenge) \cong (V/S_1; \lozenge)$. Это вместе с замечанием 3.6 показывает, что пространство с ортогональностью $(G/T; \lozenge)$ удовлетворяет условию (13).

Опишем все полные $\triangledown$-ортогональные разложения пространства G/S_0.

Множества I и $\mathfrak{I}$

Пусть I — множество всех ненулевых $\lozenge$-неразложимых элементов пространства G/T. Это значит, что I — множество всех $a \in (G/T) \setminus \{0\}$, которые не могут быть представлены в виде $a_1 + a_2$, где $a_i \in G/T$ и $\mathsf{C}^{\lozenge}_{G/T}(a_i) \supset \mathsf{C}^{\lozenge}_{G/T}(a)$ при $i = 1, 2$ (см. п. 3.3, а также [1, § 3]). По лемме 2.1 (см. также лемму 5 из [1]) множество I порождает пространство G/T. Здесь в качестве ω из леммы 2.1 используется отображение $a \mapsto \mathsf{C}^{\lozenge}_{G/T}(a)$ (где $a \in G/T$) пространства G/T в частично упорядоченное множество всех подпространств конечномерного пространства G/T относительно отношения включения.

Из леммы 3.4, которая применима здесь ввиду замечания 5.1 и второго из равенств (32), непосредственно вытекает

Следствие 5.1. Пусть $G/T = \bigoplus\limits_{m \in M} W_m$ — $\lozenge$-ортогональное разложение пространства G/T. Тогда $I = \bigsqcup\limits_{m \in M} (I \cap W_m)$.

Через $\mathfrak{I}$ будет обозначаться множество всех связных компонент графа с множеством вершин I, в котором две различные вершины a и b смежны тогда и только тогда, когда a и b не $\lozenge$-ортогональны. Лемма 3.5, которая применима здесь ввиду замечания 5.1 и второго из равенств (32), приводит к следующему выводу:

Следствие 5.2. Имеет место равенство $G/T = \bigoplus\limits_{X \in \mathfrak{I}} \langle X \rangle$, которое дает единственное полное $\lozenge$-ортогональное разложение пространства G/T.

Из равенства следствия 5.2 вытекает неравенство

$$|\mathfrak{I}| \leqslant \dim(G/T). \tag{34}$$

В частности, множество $\mathfrak{I}$ конечно.

Множества $\Xi_X(a)$

Для $a \in T/S_0$ и $X \subseteq I$ положим

$$\Xi_X(a) = \left\{ x \in X \;\middle|\; a \not\perp \rho^{-1}(x) \right\} = \left\{ x \in X \;\middle|\; \forall\, b \in \rho^{-1}(x)\ a \not\perp b \right\}.$$

Совпадение двух последних множеств следует из (33).

Лемма 5.2. Пусть $G/S_0 = \bigoplus\limits_{m\in M} U_m$ — $\triangledown$-ортогональное разложение пространства G/S_0. Пусть также $u = \sum\limits_{m\in M} u_m$, где $u_m \in (T/S_0)\cap U_m$ при всех $m\in M$, причем $\{m\in M \,|\, u_m \neq 0\}$ конечно. Тогда

$$\langle \Xi_I(u)\rangle = \bigoplus_{m\in M} \left\langle \Xi_{I\cap\rho(U_m)}(u)\right\rangle, \tag{35}$$

причем $\Xi_{I\cap\rho(U_m)}(u) = \Xi_{I\cap\rho(U_m)}(u_m) = \Xi_I(u_m)$ для любого $m\in M$.

Доказательство. Согласно следствию 5.1, примененному к разложению $G/T = \bigoplus\limits_{m\in M}\rho(U_m)$ (которое является $\diamondsuit$-ортогональным ввиду эквивалентности (18)),

$$I = \bigsqcup_{m\in M} (I\cap\rho(U_m)). \tag{36}$$

Поэтому $\Xi_I(u) = \bigsqcup\limits_{m\in M} \Xi_{I\cap\rho(U_m)}(u)$, откуда непосредственно вытекает требуемое равенство (35), так как сумма подпространств $\rho(U_m)$ по всем $m\in M$ прямая. Последнее утверждение следует из того, что разложение $G/T = \sum\limits_{m\in M}\rho(U_m)$ $\diamondsuit$-ортогонально (ввиду эквивалентности (18)).

Пусть теперь $m\in M$. Легко видеть, что если $a\in U_m$, то $u\triangledown a$ тогда и только тогда, когда $u_m \triangledown a$. Поэтому $\Xi_{I\cap\rho(U_m)}(u) = \Xi_{I\cap\rho(U_m)}(u_m)$. Равенство $\Xi_{I\cap\rho(U_m)}(u_m) = \Xi_I(u_m)$ вытекает из включения $\Xi_I(u_m)\subseteq I\cap\rho(U_m)$, которое, в свою очередь, следует из (36) и $\triangledown$-ортогональности разложения $G/S_0 = \bigoplus\limits_{m\in M} U_m$. ■

Лемма 5.3. Пусть $u\in T/S_0$. Тогда

$$\Xi_I(u) = \varnothing \iff u = 0.$$

Доказательство. Равенство $\Xi_I(0) = \varnothing$ очевидно. Предположим теперь, что $\Xi_I(u) = \varnothing$. Тогда $u\triangledown\rho^{-1}(I)$. Кроме того, $u\triangledown T/S_0$ ввиду (33). Из равенств (19) и $\langle I\rangle = G/T$ вытекает, что $\langle\rho^{-1}(I)\rangle + (T/S_0) = G/S_0$. Поэтому $u\triangledown G/S_0$ и, следовательно, $u=0$ ввиду первого из равенств (32). ■

Множество J

Обозначим через J множество всех $a\in(T/S_0)\setminus\{0\}$, которые не могут быть представлены в виде a_1+a_2, где $a_i\in T/S_0$ и $\langle\Xi_I(a_i)\rangle\subset\langle\Xi_I(a)\rangle$ при $i=1,2$. Ввиду леммы 2.1 множество J порождает пространство T/S_0. Здесь в качестве ω из леммы 2.1 используется отображение $a\mapsto\langle\Xi_I(a)\rangle$ (где $a\in T/S_0$) пространства T/S_0 в частично упорядоченное множество, двойственное частично упорядоченному множеству всех подпространств конечномерного пространства G/T относительно отношения включения.

Лемма 5.4. Пусть $G/S_0 = \bigoplus\limits_{m\in M} U_m$ — $\triangledown$-ортогональное разложение пространства G/S_0. Тогда $J\subseteq\bigcup\limits_{m\in M} U_m$ или, что эквивалентно, $J = \bigsqcup\limits_{m\in M}(J\cap U_m)$.

Доказательство. Пусть $a\in J$. Представим a в виде $\sum\limits_{m\in M} a_m$, где $a_m\in(T/S_0)\cap$ $\cap U_m = \mathsf{C}^{\diamondsuit}_{U_m}(G/S_0)$ при всех $m\in M$, причем $\{m\in M\,|\,a_m\neq 0\}$ конечно. Такое представление существует согласно лемме 3.1, так как $T/S_0 = \mathsf{C}^{\diamondsuit}_{G/S_0}(G/S_0)$ ввиду равенств $T = \mathsf{C}^{\diamondsuit}_G(G)$ (см. (27)) и (5). Выберем индекс $k\in M$, для которого $a_k\neq 0$

(это возможно, так как $a \neq 0$), и положим $b = a - a_k = \sum\limits_{m\in M\setminus\{k\}} a_m$. Из леммы 5.2 (примененной к разложению $G/S_0 = U_k \oplus W$, где $W = \bigoplus\limits_{m\in M\setminus\{k\}} U_m$) следует, что $\langle \Xi_I(a)\rangle = \langle \Xi_I(a_k)\rangle \oplus \langle \Xi_I(b)\rangle$. Кроме того, $\langle \Xi_I(a_k)\rangle \neq \{0\}$, так как $\Xi_I(a_k) \neq \varnothing$ (по лемме 5.3) и $\Xi_I(a_k) \subseteq I \subseteq (G/T)\setminus\{0\}$. Если $b \neq 0$, то аналогично показывается, что $\langle \Xi_I(b)\rangle \neq \{0\}$. Но тогда получаем противоречие с тем, что $a \in J$. Отсюда следует, что $b = 0$ и $a = a_k \in U_k \subseteq \bigcup\limits_{m\in M} U_m$. ■

Граф Γ, множества $\mathcal{I}$ и $\mathfrak{K}$ и подпространства H_X

Обозначим через Γ граф с множеством вершин $\mathfrak{I}$, в котором две различные вершины X и Y смежны тогда и только тогда, когда существует $a \in J$, такой, что $\Xi_X(a) \neq \varnothing$ и $\Xi_Y(a) \neq \varnothing$ (очевидно, что это отношение смежности симметрично). Пусть $\mathcal{I}$ — множество всех связных компонент графа Γ, а

$$\mathfrak{K} = \{\bigsqcup \mathfrak{X} \,|\, \mathfrak{X} \in \mathcal{I}\}\,.$$

Из неравенства (34) следует, что $|\mathfrak{K}| = |\mathcal{I}| \leqslant |\mathfrak{I}| \leqslant \dim(G/T)$. Отметим, что все множества из $\mathfrak{K}$ непусты.

Замечание 5.2. Из следствия 5.2 вытекает, что $G/T = \bigoplus\limits_{X\in\mathfrak{K}} \langle X\rangle$ — $\Diamond$-ортогональное разложение пространства G/T. Кроме того, это разложение нетривиально, так как $X \neq \varnothing$ и $X \subseteq I \subseteq (G/T)\setminus\{0\}$ при любом $X \in \mathfrak{K}$.

Для каждого $X \in \mathfrak{K}$ положим

$$H_X = \mathsf{C}^{\triangledown}_{T/S_0}\left(\rho^{-1}(\langle I\setminus X\rangle)\right) = \mathsf{C}^{\triangledown}_{T/S_0}\left(\rho^{-1}(I\setminus X)\right).$$

Совпадение двух последних множеств следует из (19), (33) и того, что $\mathsf{C}^{\triangledown}_{T/S_0}(\langle Y\rangle) = \mathsf{C}^{\triangledown}_{T/S_0}(Y)$ для произвольного подмножества $Y \subseteq V/S_0$. Легко также видеть, что

$$H_X = \{a \in T/S_0 \,|\, \Xi_I(a) \subseteq X\} \tag{37}$$

для любого $X \in \mathfrak{K}$.

Пусть a — произвольный элемент J. Тогда ввиду определения $\mathfrak{K}$ множество $\mathfrak{L} = \{X \in \mathfrak{K} \,|\, \Xi_X(a) \neq \varnothing\}$ не может содержать более одного элемента. С другой стороны, $\Xi_I(a) = \bigsqcup\limits_{X\in\mathfrak{K}} \Xi_X(a)$ (так как $I = \bigsqcup\mathfrak{K}$), причем $\Xi_I(a) \neq \varnothing$ по лемме 5.3. Это показывает, что $\mathfrak{L} \neq \varnothing$ и, следовательно, $|\mathfrak{L}| = 1$. Поэтому

$$J = \bigsqcup_{X\in\mathfrak{K}} (J \cap H_X) \tag{38}$$

ввиду равенства (37) и леммы 5.3.

Лемма 5.5. Имеет место равенство

$$T/S_0 = \bigoplus_{X\in\mathfrak{K}} H_X,$$

причем $H_X = \langle J \cap H_X\rangle$ для всех $X \in \mathfrak{K}$.

Доказательство. Выше было отмечено, что $T/S_0 = \langle J \rangle$. Поэтому $T/S_0 = \sum\limits_{X \in \mathfrak{K}} \langle J \cap H_X \rangle$ ввиду равенства (38). Предположим теперь, что $a \in H_Y \cap \sum\limits_{X \in \mathfrak{K} \setminus \{Y\}} H_X$ для некоторого $Y \in \mathfrak{K}$. Тогда $\Xi_I(a) \subseteq Y$ согласно равенству (37). С другой стороны, $a \triangledown \rho^{-1}(Y)$, так как $Y \subseteq I \setminus X$ для всех $X \in \mathfrak{K} \setminus \{Y\}$. Поэтому $\Xi_I(a) \subseteq I \setminus Y$. Следовательно, $\Xi_I(a) = \varnothing$ и $a = 0$ по лемме 5.3. Это показывает, что сумма подпространств H_X по всем $X \in \mathfrak{K}$ прямая. Таким образом,

$$T/S_0 = \bigoplus_{X \in \mathfrak{K}} \langle J \cap H_X \rangle = \bigoplus_{X \in \mathfrak{K}} H_X. \tag{39}$$

Равенство $H_X = \langle J \cap H_X \rangle$ для любого $X \in \mathfrak{K}$ легко вытекает из (39) и включения $\langle J \cap H_X \rangle \subseteq H_X$. ■

Согласно лемме 5.5, всякий элемент $u \in T/S_0$ единственным образом представляется в виде $\sum\limits_{X \in \mathfrak{K}} u_X$, где $u_X \in H_X$; этот элемент u_X будем называть *X-координатой* элемента u ($X \in \mathfrak{K}$). Указанная сумма определена, так как множество $\mathfrak{K}$ конечно (см. начало п. 5.2).

Лемма 5.6. Пусть $a \in \rho^{-1}(\langle X \rangle)$ и $b \in \rho^{-1}(\langle Y \rangle)$, где $X, Y \in \mathfrak{K}$, $X \neq Y$. Пусть также $u, w \in T/S_0$. Тогда $a + u \triangledown b + w$, если и только если $\sigma(a, w_X) + \sigma(b, u_Y) = -\sigma(a, b)$, где w_X — X-координата элемента w, а u_Y — Y-координата элемента u.

Доказательство. Из эквивалентности (18) вытекает, что $a \lozenge b$, так как $\rho(a) \lozenge \rho(b)$. Поэтому

$$a + u \triangledown b + w \iff 0 = \sigma(a + u, b + w) = \sigma(a, b) + \sigma(a, w) + \sigma(b, u) + \sigma(u, w). \tag{40}$$

Для любого $Z \in \mathfrak{K} \setminus \{X\}$ из определения H_Z следует, что $\rho^{-1}(\langle X \rangle) \triangledown H_Z$. Поэтому

$$\sigma(a, w) = \sigma(a, w_X) \tag{41}$$

ввиду леммы 5.5. Аналогично,

$$\sigma(b, u) = \sigma(b, u_Y). \tag{42}$$

Кроме того, из (33) вытекает равенство

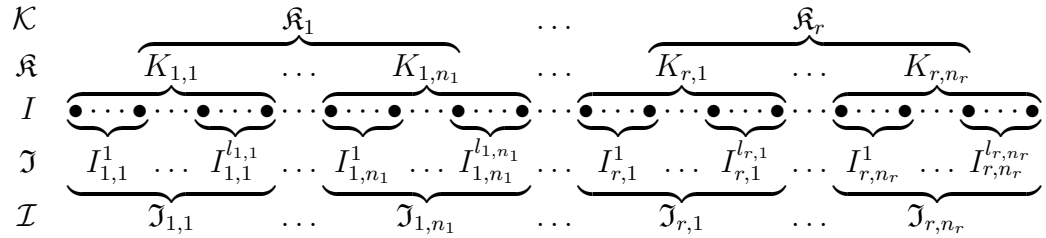
$$\sigma(u, w) = 0. \tag{43}$$

Утверждение леммы следует теперь из эквивалентности (40) и равенств (41)–(43). ■

Граф Δ, множество $\mathcal{K}$ и схема взаимных связей множеств I, $\mathfrak{I}$, $\mathcal{I}$, $\mathfrak{K}$ и $\mathcal{K}$

Обозначим через Δ граф с множеством вершин $\mathfrak{K}$, в котором две различные вершины X и Y смежны тогда и только тогда, когда не существует прямого дополнения Q_X к T/S_0 в $\rho^{-1}(\langle X \rangle)$ и прямого дополнения Q_Y к T/S_0 в $\rho^{-1}(\langle Y \rangle)$, таких, что $Q_X \triangledown Q_Y$ (очевидно, что это отношение смежности симметрично). Пусть также $\mathcal{K}$ — множество всех связных компонент графа Δ. Из неравенства (34) следует, что $|\mathcal{K}| \leqslant |\mathfrak{K}| = |\mathcal{I}| \leqslant |\mathfrak{I}| \leqslant \dim(G/T)$.

Взаимные связи множеств I, $\mathfrak{I}$, $\mathcal{I}$, $\mathfrak{K}$ и $\mathcal{K}$ могут быть изображены с помощью следующей схемы:



На этой схеме черные кружки обозначают элементы множества I. Кроме того,

$$\mathfrak{I} = \left\{ I^l_{m,n} \,|\, m \in \{1,\dots,r\},\, n \in \{1,\dots,n_m\},\, l \in \{1,\dots l_{m,n}\}\right\},$$
$$\mathfrak{I}_{m,n} = \left\{ I^1_{m,n},\dots,I^{l_{m,n}}_{m,n}\right\}, \quad \mathcal{I} = \{\mathfrak{I}_{m,n} \,|\, m \in \{1,\dots,r\},\, n \in \{1,\dots,n_m\}\},$$
$$K_{m,n} = \bigsqcup \mathfrak{I}_{m,n}, \quad \mathfrak{K} = \{K_{m,n} \,|\, m \in \{1,\dots,r\},\, n \in \{1,\dots,n_m\}\},$$
$$\mathfrak{K}_m = \{K_{m,1},\dots,K_{m,n_m}\}, \quad \mathcal{K} = \{\mathfrak{K}_1,\dots,\mathfrak{K}_r\}.$$

Критерий смежности вершин в графе Δ

Пусть X — произвольное множество из $\mathfrak{K}$. Выберем и зафиксируем некоторое линейно независимое множество D_X, такое, что $\langle D_X\rangle$ — прямое дополнение к T/S_0 в $\rho^{-1}(\langle X\rangle)$. Тогда из замечания 4.2 вытекает, что $|D_X| = \dim\langle X\rangle$ и, следовательно, множество D_X конечно. Легко также видеть, что $D_X \cap D_Y = \varnothing$ для любых различных $X, Y \in \mathfrak{K}$.

Лемма 5.7. Пусть $X, Y \in \mathfrak{K}$, $X \neq Y$. Тогда X и Y смежны в графе Δ, если и только если система уравнений

$$\sigma(a, u_{b,X}) + \sigma(b, u_{a,Y}) = -\sigma(a,b), \quad a \in D_X,\, b \in D_Y, \tag{44}$$

неразрешима относительно $u_{a,Y} \in H_Y$ $(a \in D_X)$ и $u_{b,X} \in H_X$ $(b \in D_Y)$.

Доказательство. Предположим сначала, что X и Y не смежны в графе Δ. Пусть Q_X и Q_Y — прямые дополнения к T/S_0 в $\rho^{-1}(\langle X\rangle)$ и $\rho^{-1}(\langle Y\rangle)$ соответственно, удовлетворяющие условию $Q_X \triangledown Q_Y$. Для каждого $a \in D_X$ выберем $u_a \in T/S_0$, такой, что $a + u_a \in Q_X$. Аналогично, пусть для всякого $b \in D_Y$ элемент $u_b \in T/S_0$ удовлетворяет условию $b + u_b \in Q_Y$. Тогда $a + u_a \triangledown b + u_b$. Из последнего условия и леммы 5.6 следует, что $\sigma(a, u_{b,X}) + \sigma(b, u_{a,Y}) = -\sigma(a,b)$ для любых $a \in D_X$ и $b \in D_Y$, где $u_{a,Y}$ — Y-координата элемента u_a, а $u_{b,X}$ — X-координата элемента u_b. Таким образом, система (44) разрешима.

Пусть теперь система (44) разрешима и $(u_{a,Y}, u_{b,X} \,|\, a \in D_X,\, b \in D_Y)$ — некоторое ее решение. Положим $Q_X = \langle a + u_{a,Y} \,|\, a \in D_X\rangle$ и $Q_Y = \langle b + u_{b,X} \,|\, b \in D_Y\rangle$. Тогда очевидно, что $\rho^{-1}(\langle X\rangle) = Q_X \oplus (T/S_0)$ и $\rho^{-1}(\langle Y\rangle) = Q_Y \oplus (T/S_0)$. Кроме того, ввиду леммы 5.6 $a + u_{a,Y} \triangledown b + u_{b,X}$ для всех $a \in D_X$ и $b \in D_Y$. Поэтому $Q_X \triangledown Q_Y$. Таким образом, X и Y не смежны в графе Δ. ■

Замечание 5.3. Пусть X и Y — различные множества из $\mathfrak{K}$, а Z_X и Z_Y — некоторые базисы пространств H_X и H_Y соответственно (ввиду замечания 4.1 Z_X и Z_Y конечны). Тогда очевидно, что $(u_{a,Y}, u_{b,X} \,|\, a \in D_X,\, b \in D_Y)$ — решение системы (44), если и только если координаты элементов $u_{a,Y}$ в базисе Z_Y и элементов $u_{b,X}$ в базисе Z_X образуют решение системы уравнений

$$\sum_{z_X \in Z_X} \sigma(a, z_X) u_{b,X,z_X} + \sum_{z_Y \in Z_Y} \sigma(b, z_Y) u_{a,Y,z_Y} = -\sigma(a,b), \quad a \in D_X,\, b \in D_Y, \tag{45}$$

относительно $u_{a,Y,z_Y}, u_{b,X,z_X} \in F$ $(a \in D_X,\, b \in D_Y,\, z_X \in Z_X,\, z_Y \in Z_Y)$, где u_{a,Y,z_Y} — координата $u_{a,Y}$ при z_Y, а u_{b,X,z_X} — координата $u_{b,X}$ при z_X. Таким образом, нахождение решений системы (44) эквивалентно нахождению решений конечной системы (45) линейных уравнений над полем F, которая состоит из

$$|D_X||D_Y| = \dim\langle X\rangle \dim\langle Y\rangle$$

уравнений и содержит

$$|D_X||Z_Y| + |D_Y||Z_X| = \dim\langle X\rangle \dim H_Y + \dim\langle Y\rangle \dim H_X$$

переменных, принимающих значения в F (здесь использовано то, что $|D_X| = \dim\langle X\rangle$ и $|D_Y| = \dim\langle Y\rangle$ ввиду замечания 4.2).

Множество $\mathcal{Q}$

Обозначим через $\mathcal{Q}$ множество всех семейств $(Q_X \,|\, X \in \mathfrak{K})$, удовлетворяющих следующим условиям:

1) для любого $X \in \mathfrak{K}$ семейство Q_X является прямым дополнением к T/S_0 в $\rho^{-1}(\langle X\rangle)$;
2) если $X, Y \in \mathfrak{K}$, причем X и Y принадлежат различным связным компонентам графа Δ, то $Q_X \triangledown Q_Y$.

Из утверждения 2 следующей леммы, в частности, вытекает, что $\mathcal{Q} \neq \varnothing$.

Лемма 5.8. Для каждого двухэлементного подмножества $\{X, Y\} \subseteq \mathfrak{K}$, в котором X и Y принадлежат различным связным компонентам графа Δ, выберем некоторое решение $(u_{a,Y}, u_{b,X} \,|\, a \in D_X,\, b \in D_Y)$ системы уравнений (44) (такое решение существует ввиду леммы 5.7, так как X и Y не смежны в графе Δ). Выберем также произвольный элемент $u_{a,Y} \in H_Y$ для всех $a \in D_X$ ($X \in \mathfrak{K}$) и $Y \in \mathfrak{K}$, таких, что X и Y принадлежат одной связной компоненте графа Δ. Для каждого $X \in \mathfrak{K}$ положим

$$E_X = \left\{ a + \sum_{Y \in \mathfrak{K}} u_{a,Y} \;\middle|\; a \in D_X \right\}.$$

Тогда

1) при любом $X \in \mathfrak{K}$ множество E_X линейно независимо;
2) семейство $(\langle E_X\rangle \,|\, X \in \mathfrak{K})$ принадлежит $\mathcal{Q}$.

Доказательство. Утверждение 1 очевидно. Ясно также, что $\rho^{-1}(\langle X\rangle) = \langle E_X\rangle \oplus (T/S_0)$ для любого $X \in \mathfrak{K}$. Пусть теперь $X, Y \in \mathfrak{K}$, причем X и Y принадлежат различным связным компонентам графа Δ. Тогда из леммы 5.6 непосредственно вытекает, что $E_X \triangledown E_Y$ и, следовательно, $\langle E_X\rangle \triangledown \langle E_Y\rangle$. Таким образом, утверждение 2 доказано. ■

Замечание 5.4. Пусть $(Q_X \,|\, X \in \mathfrak{K}) \in \mathcal{Q}$. Пусть также X — произвольное множество из $\mathfrak{K}$. Для каждого $a \in D_X$ выберем элемент $u_a \in T/S_0$, такой, что $a + u_a \in Q_X$. Положим $E_X = \{a + u_a \,|\, a \in D_X\}$. Очевидно, что множество E_X линейно независимо и, следовательно, $\dim\langle E_X\rangle = |D_X| = \dim\langle D_X\rangle = \dim\langle X\rangle = \dim Q_X$ (см. замечание 4.2), причем размерности в этих равенствах конечны. Поэтому из очевидного включения $\langle E_X\rangle \subseteq Q_X$ вытекает равенство $\langle E_X\rangle = Q_X$.

Для произвольных $a \in \bigsqcup\limits_{X \in \mathfrak{K}} D_X$ и $Y \in \mathfrak{K}$ обозначим через $u_{a,Y}$ Y-координату элемента u_a. Предположим, что X и Y (где $X, Y \in \mathfrak{K}$) принадлежат различным связным компонентам графа Δ. Тогда $Q_X \triangledown Q_Y$ и, следовательно, $E_X \triangledown E_Y$. Поэтому из леммы 5.6 вытекает, что $(u_{a,Y}, u_{b,X} \,|\, a \in D_X,\, b \in D_Y)$ является решением системы уравнений (44). Таким образом, конструкция, описанная в лемме 5.8, позволяет построить все семейства из $\mathcal{Q}$.

Множества $\mathcal{K}_U$

Пусть U — произвольное подпространство G/S_0, входящее в некоторое $\triangledown$-ортогональное разложение $G/S_0 = U \oplus W$ пространства G/S_0. Ввиду замечания 3.2 (которое применимо здесь согласно первому из равенств (32)) для произвольного такого

разложения W совпадает с $\mathsf{C}_V^{\perp}(U)$. Следовательно, это $\triangledown$-ортогональное разложение при фиксированном U единственно. Обозначения U и W будут иметь вышеуказанный смысл в пределах всего п. 5.2.

Применив к разложению $G/S_0 = U \oplus W$ гомоморфизм ρ, получим разложение $G/T = \rho(U) \oplus \rho(W)$ пространства G/T, которое является $\diamondsuit$-ортогональным (и, следовательно, прямым) ввиду эквивалентности (18). Поставим в соответствие подпространству U подмножество $\mathcal{K}_U \subseteq \mathcal{K}$, такое, что

$$\rho(U) = \bigoplus_{X \in \bigsqcup \mathcal{K}_U} \langle X \rangle. \tag{46}$$

Корректность этого определения непосредственно вытекает из следующей леммы.

Лемма 5.9. Подмножество $\mathcal{K}_U \subseteq \mathcal{K}$, удовлетворяющее равенству (46), существует и единственно (при фиксированном U).

Доказательство. Из следствия 5.2 вытекает, что $\rho(U) = \bigoplus\limits_{X \in \mathfrak{I}_U} \langle X \rangle$ и $\rho(W) = \bigoplus\limits_{Y \in \mathfrak{I} \setminus \mathfrak{I}_U} \langle Y \rangle$, где $\mathfrak{I}_U$ — некоторое однозначно определенное подмножество $\mathfrak{I}$. Предположим, что некоторые $X \in \mathfrak{I}_U$ и $Y \in \mathfrak{I} \setminus \mathfrak{I}_U$ смежны в графе Γ. Тогда существуют $u \in U$, $w \in W$ и $z \in J$, такие, что $u \not\triangledown z$ и $w \not\triangledown z$. С другой стороны, $z \in U$ или $z \in W$ ввиду леммы 5.4. Но тогда соответственно $w \triangledown z$ или $u \triangledown z$. Полученное противоречие показывает, что $\mathfrak{I}_U = \bigsqcup \mathcal{I}_U$, где $\mathcal{I}_U$ — некоторое однозначно определенное подмножество $\mathcal{I}$.

Положим $\mathfrak{K}_U = \{\bigsqcup \mathfrak{X} \,|\, \mathfrak{X} \in \mathcal{I}_U\}$. Тогда легко видеть, что $\rho(U) = \bigoplus\limits_{X \in \mathfrak{K}_U} \langle X \rangle$ и $\rho(W) = \bigoplus\limits_{Y \in \mathfrak{K} \setminus \mathfrak{K}_U} \langle Y \rangle$, так как $\mathfrak{I} \setminus \mathfrak{I}_U = \bigsqcup(\mathcal{I} \setminus \mathcal{I}_U)$ и $\mathfrak{K} \setminus \mathfrak{K}_U = \{\bigsqcup \mathfrak{Y} \,|\, \mathfrak{Y} \in \mathcal{I} \setminus \mathcal{I}_U\}$. Предположим, что некоторые $X \in \mathfrak{K}_U$ и $Y \in \mathfrak{K} \setminus \mathfrak{K}_U$ смежны в графе Δ. Тогда легко видеть, что $\rho^{-1}(\langle X \rangle) \subseteq U + (T/S_0)$, поэтому $\rho^{-1}(\langle X \rangle) = Q_X \oplus (T/S_0)$ для некоторого подпространства $Q_X \subseteq U$. Аналогично, $\rho^{-1}(\langle Y \rangle) \subseteq W + (T/S_0)$ и $\rho^{-1}(\langle Y \rangle) = Q_Y \oplus (T/S_0)$ для некоторого подпространства $Q_Y \subseteq W$. Кроме того, такие подпространства Q_X и Q_Y $\triangledown$-ортогональны ввиду $\triangledown$-ортогональности U и W. Но это означает, что X и Y не смежны в графе Δ. Полученное противоречие показывает, что $\mathfrak{K}_U = \bigsqcup \mathcal{K}_U$, где $\mathcal{K}_U$ — некоторое однозначно определенное подмножество $\mathcal{K}$. Таким образом, существование требуемого множества $\mathcal{K}_U$ доказано, а единственность этого множества следует из замечания 5.2. ■

Замечание 5.5. Очевидно, что $\mathcal{K}_{G/S_0} = \mathcal{K}$. Непосредственно проверяется также, что если $U = \bigoplus\limits_{m \in M} U_m$ — $\triangledown$-ортогональное разложение пространства U, то $\mathcal{K}_U = \bigsqcup\limits_{m \in M} \mathcal{K}_{U_m}$.

Из определения множества $\mathcal{K}_U$ вытекает, что $\rho(U) = \langle \bigsqcup\bigsqcup \mathcal{K}_U \rangle$. Следовательно, $\bigsqcup\bigsqcup \mathcal{K}_U \subseteq I \cap \rho(U)$. Аналогично, $\bigsqcup\bigsqcup \mathcal{K}_W \subseteq I \cap \rho(W)$. Кроме того, $(\bigsqcup\bigsqcup \mathcal{K}_U) \sqcup \sqcup (\bigsqcup\bigsqcup \mathcal{K}_W) = I$ (так как $\mathcal{K}_U \sqcup \mathcal{K}_W = \mathcal{K}$ ввиду замечания 5.5) и $(I \cap \rho(U)) \sqcup (I \cap \rho(W)) = I$ (согласно следствию 5.1). Поэтому

$$I \cap \rho(U) = \bigsqcup\bigsqcup \mathcal{K}_U. \tag{47}$$

Кроме того, легко видеть, что произвольное подмножество $\mathcal{L} \subseteq \mathcal{K}$ определяется по $\bigsqcup\bigsqcup \mathcal{L}$ однозначно (так как $\mathfrak{K}$ состоит из непустых попарно не пересекающихся множеств). Это показывает, что $\mathcal{K}_U$ эквивалентным образом можно определить как подмножество $\mathcal{K}$, удовлетворяющее равенству (47).

Лемма 5.10. Имеет место равенство

$$(T/S_0)\cap U=\bigoplus_{X\in\bigsqcup\mathcal{K}_U}H_X. \tag{48}$$

Доказательство. Пусть X — произвольное множество из $\bigsqcup\mathcal{K}_U$. Предположим, что существует $a\in J\cap H_X\cap W$. Тогда, с одной стороны, $\Xi_I(a)\subseteq X$ (см. равенство (37)). С другой стороны, очевидно, что $\rho^{-1}(X)\subseteq U+(T/S_0)$. Кроме того, $a\triangledown U$ (так как $W\triangledown U$) и $a\triangledown T/S_0$ (ввиду (33)). Поэтому $a\triangledown\rho^{-1}(X)$ и $\Xi_I(a)\subseteq I\setminus X$. Но тогда $\Xi_I(a)=\varnothing$ и, следовательно, $a=0$ (см. лемму 5.3), что противоречит принадлежности a множеству J. Полученное противоречие вместе с равенством

$$J=(J\cap U)\sqcup(J\cap W), \tag{49}$$

имеющим место ввиду леммы 5.4, показывают, что

$$\bigsqcup_{X\in\bigsqcup\mathcal{K}_U}(J\cap H_X)\subseteq J\cap U. \tag{50}$$

Аналогично,

$$\bigsqcup_{Y\in\bigsqcup\mathcal{K}_W}(J\cap H_Y)\subseteq J\cap W. \tag{51}$$

Из (38) и (49)–(51) вытекает равенство

$$J\cap U=\bigsqcup_{X\in\bigsqcup\mathcal{K}_U}(J\cap H_X),$$

так как $\mathcal{K}_U\sqcup\mathcal{K}_W=\mathcal{K}$ (ввиду замечания 5.5) и $(\bigsqcup\mathcal{K}_U)\sqcup(\bigsqcup\mathcal{K}_W)=\bigsqcup\mathcal{K}=\mathfrak{K}$. Поэтому

$$\langle J\cap U\rangle=\bigoplus_{X\in\bigsqcup\mathcal{K}_U}\langle J\cap H_X\rangle=\bigoplus_{X\in\bigsqcup\mathcal{K}_U}H_X \tag{52}$$

(см. лемму 5.5). Кроме того,

$$T/S_0=\langle J\rangle=\langle J\cap U\rangle\oplus\langle J\cap W\rangle$$

ввиду равенства (49), где $\langle J\cap U\rangle\subseteq(T/S_0)\cap U$ и $\langle J\cap W\rangle\subseteq(T/S_0)\cap W$. Следовательно, $\langle J\cap U\rangle=(T/S_0)\cap U$. Требуемое равенство (48) непосредственно вытекает из последнего равенства и (52). ■

Лемма 5.11. Следующие условия эквивалентны:

1) $U=\{0\}$;
2) $\rho(U)=\{0\}$;
3) $\mathcal{K}_U=\varnothing$.

Доказательство. Импликация $1\Longrightarrow 2$ и эквивалентность $2\Longleftrightarrow 3$ очевидны (см. замечание 5.2). Пусть теперь выполнены эквивалентные условия 2 и 3. Тогда $U\subseteq T/S_0$ (ввиду условия 2) и $(T/S_0)\cap U=\{0\}$ (ввиду условия 3 и леммы 5.10). Поэтому $U=(T/S_0)\cap U=\{0\}$, т. е. выполнено условие 1. ■

Из эквивалентности (18) и эквивалентности условий 1 и 2 леммы 5.11 вытекает

Следствие 5.3. Пусть пространство G/T $\Diamond$-неразложимо или, что эквивалентно, $\mathfrak{I}=\{I\}$ (см. следствие 5.2). Тогда пространство G/S_0 $\triangledown$-неразложимо.

Подпространства $C^{\mathfrak{Q}}_{\mathfrak{X}}$

Для произвольных $\mathfrak{Q} = (Q_X \mid X \in \mathfrak{K}) \in \mathcal{Q}$ и $\mathfrak{X} \in \mathcal{K}$ положим

$$C^{\mathfrak{Q}}_{\mathfrak{X}} = \bigoplus_{X \in \mathfrak{X}} (Q_X \oplus H_X).$$

Из замечания 5.2 и леммы 5.5 следует, что эта сумма прямая.

Замечание 5.6. Пусть $\mathfrak{Q} \in \mathcal{Q}$. Тогда легко видеть, что

$$\rho\left(C^{\mathfrak{Q}}_{\mathfrak{X}}\right) = \bigoplus_{X \in \mathfrak{X}} \langle X \rangle$$

при всех $\mathfrak{X} \in \mathcal{K}$. Из этого ввиду замечания 5.2 вытекает, что $\rho\left(C^{\mathfrak{Q}}_{\mathfrak{X}}\right) \neq \rho\left(C^{\mathfrak{Q}}_{\mathfrak{Y}}\right)$ (и, следовательно, $C^{\mathfrak{Q}}_{\mathfrak{X}} \neq C^{\mathfrak{Q}}_{\mathfrak{Y}}$) для любых различных $\mathfrak{X}, \mathfrak{Y} \in \mathcal{K}$.

Лемма 5.12. Для любого $\mathfrak{Q} \in \mathcal{Q}$ имеет место равенство

$$G/S_0 = \bigoplus_{\mathfrak{X} \in \mathcal{K}} C^{\mathfrak{Q}}_{\mathfrak{X}}, \tag{53}$$

которое дает полное $\triangledown$-ортогональное разложение пространства G/S_0.

Доказательство. Пусть $\mathfrak{Q} = (Q_X \mid X \in \mathfrak{K})$. Непосредственно проверяется (с использованием замечания 5.2 и леммы 5.5), что

$$\begin{aligned} G/S_0 = \rho^{-1}(G/T) = \left(\sum_{X \in \mathfrak{K}} \rho^{-1}(\langle X \rangle)\right) + (T/S_0) = \left(\bigoplus_{X \in \mathfrak{K}} Q_X\right) \oplus (T/S_0) = \\ = \bigoplus_{X \in \mathfrak{K}} (Q_X \oplus H_X) = \bigoplus_{\mathfrak{X} \in \mathcal{K}} \bigoplus_{X \in \mathfrak{X}} (Q_X \oplus H_X) = \bigoplus_{\mathfrak{X} \in \mathcal{K}} C^{\mathfrak{Q}}_{\mathfrak{X}}. \end{aligned}$$

Таким образом, равенство (53) выполнено.

Выберем произвольное множество $\mathfrak{X} \in \mathcal{K}$. Пусть также $\mathfrak{Y} \in \mathcal{K} \setminus \{\mathfrak{X}\}$, $X \in \mathfrak{X}$ и $Y \in \mathfrak{Y}$. Тогда

— $Q_X \triangledown Q_Y$, так как $\mathfrak{Q} \in \mathcal{Q}$;
— $Q_X \triangledown H_Y$ и $Q_Y \triangledown H_X$, так как $Q_X \subseteq \rho^{-1}(\langle X \rangle)$ и $Q_Y \subseteq \rho^{-1}(\langle Y \rangle)$;
— $H_X \triangledown H_Y$ ввиду (33).

Поэтому $C^{\mathfrak{Q}}_{\mathfrak{X}} \triangledown C^{\mathfrak{Q}}_{\mathfrak{Y}}$. Таким образом, разложение (53) $\triangledown$-ортогонально.

Так как $\mathfrak{X} \neq \varnothing$, из замечаний 5.6 и 5.2 следует, что $C^{\mathfrak{Q}}_{\mathfrak{X}} \neq \{0\}$. Предположим теперь, что $C^{\mathfrak{Q}}_{\mathfrak{X}} = U \oplus W$ — нетривиальное $\triangledown$-ортогональное разложение пространства $C^{\mathfrak{Q}}_{\mathfrak{X}}$. Тогда $\mathcal{K}_{C^{\mathfrak{Q}}_{\mathfrak{X}}} = \mathcal{K}_U \sqcup \mathcal{K}_W$ согласно замечанию 5.5. Но множество в левой части этого равенства одноэлементно (оно равно $\{\mathfrak{X}\}$ ввиду замечания 5.6), а множество в правой части содержит не менее двух элементов (так как $\mathcal{K}_U \neq \varnothing$ и $\mathcal{K}_W \neq \varnothing$ по лемме 5.11). Полученное противоречие показывает, что пространство $C^{\mathfrak{Q}}_{\mathfrak{X}}$ $\triangledown$-неразложимо. ■

Лемма 5.13. Пусть $G/S_0 = \bigoplus \mathfrak{C}$ — полное $\triangledown$-ортогональное разложение пространства G/S_0. Тогда существует семейство $\mathfrak{Q} \in \mathcal{Q}$, такое, что

$$\mathfrak{C} = \left\{ C^{\mathfrak{Q}}_{\mathfrak{X}} \;\middle|\; \mathfrak{X} \in \mathcal{K} \right\}. \tag{54}$$

Доказательство. Согласно замечанию 5.5,

$$\bigsqcup_{U \in \mathfrak{C}} \mathcal{K}_U = \mathcal{K}. \tag{55}$$

Пусть X — произвольное множество из $\mathfrak{K}$. Тогда ввиду равенства (55) существует единственное подпространство $U \in \mathfrak{C}$, такое, что $X \in \bigsqcup \mathcal{K}_U$. Легко видеть, что $\rho^{-1}(\langle X\rangle) \subseteq U + (T/S_0)$. Поэтому можно выбрать линейно независимое множество $D_X \subseteq U$, удовлетворяющее равенству $\rho^{-1}(\langle X\rangle) = \langle D_X\rangle \oplus (T/S_0)$.

Построим теперь семейство $\mathfrak{Q} = (\langle E_X\rangle \,|\, X \in \mathfrak{K}) \in \mathcal{Q}$ в соответствии с леммой 5.8, используя выбранные в предыдущем абзаце множества D_X. При этом для любого двухэлементного множества $\{X, Y\}$, в котором $X \in \bigsqcup \mathcal{K}_U$ и $Y \in \bigsqcup \mathcal{K}_W$, где U и W — различные подпространства из $\mathfrak{C}$, выберем решение $(u_{a,Y}, u_{b,X} \,|\, a \in D_X,\, b \in D_Y)$ системы уравнений (44), состоящее из одних нулей. Это допустимо, так как $\sigma(a, b) = 0$ при всех $a \in D_X$ и $b \in D_Y$ ввиду того, что $D_X \subseteq U$, $D_Y \subseteq W$ и $U \triangledown W$.

Пусть W — произвольное подпространство из $\mathfrak{C}$. Тогда из леммы 5.10 вытекает, что $\bigoplus\limits_{\mathfrak{X}\in\mathcal{K}_W} C^{\mathfrak{Q}}_{\mathfrak{X}} \subseteq W$. В то же время $V/S_0 = \bigoplus\limits_{U\in\mathfrak{C}} \left(\bigoplus_{\mathfrak{X}\in\mathcal{K}_U} C^{\mathfrak{Q}}_{\mathfrak{X}}\right)$ ввиду равенств (53) (см. лемму 5.12) и (55). Кроме того, $V/S_0 = \bigoplus\limits_{U\in\mathfrak{C}} U$. Следовательно, $\bigoplus\limits_{\mathfrak{X}\in\mathcal{K}_W} C^{\mathfrak{Q}}_{\mathfrak{X}} = W$. Так как W $\triangledown$-неразложимо, из последнего равенства и леммы 5.12 вытекает, что $\mathcal{K}_W = \{\mathfrak{Z}_W\}$ и

$$C^{\mathfrak{Q}}_{\mathfrak{Z}_W} = W \tag{56}$$

для некоторого $\mathfrak{Z}_W \in \mathcal{K}$. Требуемое равенство (54) следует теперь из равенства (56) (верного при любом $W \in \mathfrak{C}$) и того, что $W \mapsto \mathfrak{Z}_W$ — биекция $\mathfrak{C}$ на $\mathcal{K}$ (это вытекает из равенства (55)). ■

Леммы 5.12 и 5.13 показывают, что полные $\triangledown$-ортогональные разложения пространства G/S_0 — это в точности разложения вида $G/S_0 = \bigoplus\limits_{\mathfrak{X}\in\mathcal{K}} C^{\mathfrak{Q}}_{\mathfrak{X}}$, где $\mathfrak{Q} \in \mathcal{Q}$.

Лемма 5.14. Пусть $\mathfrak{Q}, \mathfrak{Q}' \in \mathcal{Q}$ и $\mathfrak{X} \in \mathcal{K}$. Тогда

$$\left(C^{\mathfrak{Q}}_{\mathfrak{X}}; \sigma\right) \cong \left(C^{\mathfrak{Q}'}_{\mathfrak{X}}; \sigma\right). \tag{57}$$

Доказательство. Положим для краткости $U = \sum\limits_{X\in\mathfrak{X}} \rho^{-1}(\langle X\rangle)$ и $W = \bigoplus\limits_{X\in\mathfrak{K}\setminus\mathfrak{X}} H_X$. Тогда легко видеть, что $C^{\mathfrak{Q}}_{\mathfrak{X}} \oplus W = U$ (см. лемму 5.5). Из леммы 5.12 вытекает, что $C^{\mathfrak{Q}}_{\mathfrak{Y}} \triangledown C^{\mathfrak{Q}}_{\mathfrak{X}}$ при любом $\mathfrak{Y} \in \mathcal{K} \setminus \{\mathfrak{X}\}$. Следовательно, $W \triangledown C^{\mathfrak{Q}}_{\mathfrak{X}}$, так как $W \subseteq \bigoplus\limits_{\mathfrak{Y}\in\mathcal{K}\setminus\{\mathfrak{X}\}} C^{\mathfrak{Q}}_{\mathfrak{Y}}$. Кроме того, $W \triangledown W$ ввиду (33). Поэтому $W \subseteq \mathsf{C}^{\triangledown}_U(U)$ и

$$\left(C^{\mathfrak{Q}}_{\mathfrak{X}}; \sigma\right) \cong (U/W; \sigma) \tag{58}$$

согласно замечанию 4.3. Аналогично,

$$\left(C^{\mathfrak{Q}'}_{\mathfrak{X}}; \sigma\right) \cong (U/W; \sigma). \tag{59}$$

Требуемый изоморфизм (57) вытекает теперь из (58) и (59). ■

5.3. Основной результат

Из теоремы 5.1, лемм 5.12 и 5.13 и замечания 5.6 непосредственно вытекает следующий основной результат настоящей работы. В его формулировке напоминаются предположения, которые сделаны выше.

Теорема 5.2. Предположим, что пространство с ортогональностью $(V; \lozenge)$ удовлетворяет условию (13) и что пространство V/T конечномерно. Тогда полные $\triangledown$-ортогональные разложения пространства V — это в точности разложения вида

$$V = \left(\bigoplus \mathfrak{A}\right) \oplus \left(\bigoplus \mathfrak{B}\right) \oplus \bigoplus_{\mathfrak{X}\in\mathcal{K}(G(\mathfrak{A}))} \nu^{-1}_{R,S_0}\left(C^{\mathfrak{Q}}_{\mathfrak{X}}(G(\mathfrak{A}))\right), \tag{60}$$

где $\mathfrak{A} \in \mathcal{A}$, $\mathfrak{B} \in \mathcal{B}$, $R \in \mathfrak{R}(G(\mathfrak{A}))$ и $\mathfrak{Q} \in \mathcal{Q}(G(\mathfrak{A}))$. Множества $\mathfrak{A}$ и $\mathfrak{B}$ и подпространства R и $C^{\mathfrak{Q}}_{\mathfrak{X}}(G(\mathfrak{A}))$ для каждого $\mathfrak{X} \in \mathcal{K}(G(\mathfrak{A}))$ определяются произвольным разложением (60) однозначно. Кроме того, если $\mathfrak{S}$ — множество всех подпространств, входящих в это разложение, т. е.

$$\mathfrak{S} = \mathfrak{A} \sqcup \mathfrak{B} \sqcup \left\{ \nu^{-1}_{R,S_0}\left(C^{\mathfrak{Q}}_{\mathfrak{X}}(G(\mathfrak{A}))\right) \,\middle|\, \mathfrak{X} \in \mathcal{K}(G(\mathfrak{A}))\right\},$$

то $\mathfrak{A}$, $\mathfrak{B}$ и R могут быть найдены по $\mathfrak{S}$ с помощью формул из теоремы 5.1, а $C^{\mathfrak{Q}}_{\mathfrak{X}}(G(\mathfrak{A}))$ для произвольного $\mathfrak{X} \in \mathcal{K}(G(\mathfrak{A}))$ — это единственное подпространство

$$U \in \nu_{R,S_0}(\mathfrak{S} \setminus (\mathfrak{A} \sqcup \mathfrak{B})),$$

такое, что $\rho(U) = \bigoplus\limits_{X \in \mathfrak{X}} \langle X \rangle$.

6. Приложения

Напомним, что здесь, как и в п. 5.2 и 5.3, делаются следующие предположения:

— пространство с ортогональностью $(V; \Diamond)$ удовлетворяет условию (13);
— пространство V/T конечномерно.

6.1. Критерий ∇-неразложимости

Теорема 6.1. Пространство V ∇-неразложимо тогда и только тогда, когда выполнено одно из следующих взаимоисключающих условий:

1) V — пространство типа 1 или 2;
2) V одномерно и σ тождественно равна 0 на $V \times V$;
3) $S_0 = \{0\}$, σ тождественно равна 0 на $S_1 \times S_1$, граф $\Delta(V)$ имеет непустое множество вершин и связен (это условие для графа $\Delta(V)$ может быть записано в виде равенства $\mathcal{K}(V) = \{\mathfrak{K}(V)\}$).

Доказательство. Из лемм 4.2 и 5.12 следует, что при выполнении любого из условий теоремы пространство V ∇-неразложимо (очевидно, что если выполнено условие 3, то $T = S_1$, $\mathcal{A} = \{\varnothing\}$ и $V = G(\varnothing)$; здесь используется также то, что $\mathcal{Q}(V) \neq \varnothing$ ввиду п. 2 леммы 5.8). Пусть теперь пространство V ∇-неразложимо. По теореме 5.1 существует единственная четверка $(\mathfrak{A}, \mathfrak{B}, R, \mathfrak{C}) \in \mathcal{P}$, такая, что

$$\{V\} = \mathfrak{A} \sqcup \mathfrak{B} \sqcup \nu^{-1}_{R,S_0}(\mathfrak{C}).$$

Возможны три случая:

1) $\mathfrak{A} = \{V\}$, $\mathfrak{B} = \varnothing$, $\mathfrak{C} = \varnothing$;
2) $\mathfrak{A} = \varnothing$, $\mathfrak{B} = \{V\}$, $\mathfrak{C} = \varnothing$;
3) $\mathfrak{A} = \varnothing$, $\mathfrak{B} = \varnothing$, $\nu^{-1}_{R,S_0}(\mathfrak{C}) = \{V\}$.

Легко заметить, что в каждом из этих случаев выполняется условие леммы с тем же номером (в случае 3 для этого нужно использовать лемму 5.13 и замечание 5.6). ■

Следующий пример приводится как для иллюстрации приведенного выше материала, так и для описания конструкции, которая понадобится нам в дальнейшем.

Пример 6.1. Предположим, что $\operatorname{char} F = 2$ и V конечномерно, причем $\dim V \geqslant 3$. Пусть U — подпространство V, имеющее размерность не более $(\dim V) - 3$, а τ — билинейная функция из $U \times V$ в F, удовлетворяющая следующим условиям:

— τ тождественно равна 0 на $U \times U$;
— для любого $u \in U \setminus \{0\}$ существует $v \in V$, такой, что $\tau(u, v) \neq 0$.

Очевидно, что такая билинейная функция τ существует тогда и только тогда, когда $\dim U \leqslant (\dim V)/2$.

Определим подмножество $\lozenge \subseteq V \times V$ и функцию $\sigma\colon \lozenge \to F$ следующим образом. Пусть $x, y \in V$. Тогда считаем, что $x \lozenge y$, если и только если $x+U$ и $y+U$ линейно зависимы (в V/U). Предположим теперь, что последнее условие выполнено. Если $x \in U$, то полагаем $\sigma(x,y) = \tau(x,y)$. Если же $x \notin U$, то y может быть единственным образом представлен в виде $\lambda x + u$, где $\lambda \in F$ и $u \in U$. Тогда полагаем $\sigma(x,y) = \tau(u,x)$. Непосредственно проверяется, что σ — частичная симметричная билинейная функция на V с областью определения $\lozenge$, причем $0 \lozenge V$ и

$$\mathsf{C}_V^{\lozenge}(x) = \begin{cases} V, & \text{если } x \in U, \\ \langle x \rangle \oplus U, & \text{если } x \in V \setminus U. \end{cases}$$

Используя эту формулу и неравенство $\dim V \geqslant (\dim U) + 3$, легко видеть, что

— пространство с ортогональностью $(V; \lozenge)$ удовлетворяет условию (13);
— $S_0 = \{0\}$ и $S_1 = T = U$;
— $\mathsf{C}_{V/U}^{\lozenge}(a) = \langle a \rangle$ при всех $a \in (V/U) \setminus \{0\}$ (для доказательства этого утверждения используется также эквивалентность (4)).

Поэтому $\mathcal{A} = \{\varnothing\}$, $\mathcal{B} = \{\varnothing\}$, $\mathfrak{R}(V) = \{V\}$, $I(V) = (V/U) \setminus \{0\}$, $\mathfrak{I}(V) = \{I(V)\}$, $\mathcal{I}(V) = \{\mathfrak{I}(V)\}$, $\mathfrak{K}(V) = \{I(V)\}$ и $\mathcal{K}(V) = \{\mathfrak{K}(V)\}$, где $V = G(\varnothing)$. Согласно следствию 5.3, пространство V $\triangledown$-неразложимо. Тот же самый вывод можно сделать, используя теорему 6.1 (для V выполняется условие 3 этой теоремы). Кроме того, $x \triangledown x$ для любого $x \in V$.

6.2. Об эквивалентности полных $\triangledown$-ортогональных разложений

Замечание 6.1. Пусть $\mathfrak{B}, \mathfrak{B}' \in \mathcal{B}$. Тогда $|\mathfrak{B}| = |\mathfrak{B}'| = \dim S_0$. Кроме того, если U и W — произвольные изоморфные (или, что эквивалентно, имеющие одинаковую размерность) подпространства S_0, то всякий изоморфизм U на W является изоморфизмом $(U;\sigma)$ на $(W;\sigma)$. Поэтому биекции $U \mapsto W_U$ множества $\mathfrak{B}$ на множество $\mathfrak{B}'$ существуют и удовлетворяют условию $(U;\sigma) \cong (W_U;\sigma)$ для всех $U \in \mathfrak{B}$.

Лемма 6.1. Пусть $(\mathfrak{A}, \mathfrak{B}, R, \mathfrak{C}), (\mathfrak{A}', \mathfrak{B}', R', \mathfrak{C}') \in \mathcal{P}$. Предположим, что $(G(\mathfrak{A});\sigma) \cong (G(\mathfrak{A}');\sigma)$. Тогда существует биекция $U \mapsto W_U$ множества $\nu_{R,S_0}^{-1}(\mathfrak{C})$ на множество $\nu_{R',S_0}^{-1}(\mathfrak{C}')$, такая, что $(U;\sigma) \cong (W_U;\sigma)$ для любого $U \in \nu_{R,S_0}^{-1}(\mathfrak{C})$.

Доказательство. Согласно замечанию 4.3, $(\bigoplus \mathfrak{A};\sigma) \cong (S_1/T;\sigma) \cong (\bigoplus \mathfrak{A}';\sigma)$. Поэтому ввиду равенства (26) и п. 1 леммы 4.1 можем выбрать $\alpha \in \mathrm{Aut}(V;\sigma)$, для которого $\alpha(\bigoplus \mathfrak{A}') = \bigoplus \mathfrak{A}$. Применив α к элементам множества $\mathfrak{A}' \sqcup \mathfrak{B}' \sqcup \nu_{R',S_0}^{-1}(\mathfrak{C}')$, считаем, что $\bigoplus \mathfrak{A} = \bigoplus \mathfrak{A}'$ (очевидно, что утверждение леммы до этой замены равносильно таковому после нее). Тогда $G(\mathfrak{A}) = G(\mathfrak{A}')$; обе части последнего равенства для краткости обозначим через G.

По лемме 5.13 существуют $\mathfrak{Q}, \mathfrak{Q}' \in \mathcal{Q}(G)$, такие, что

$$\mathfrak{C} = \left\{ C_{\mathfrak{X}}^{\mathfrak{Q}}(G) \,\middle|\, \mathfrak{X} \in \mathcal{K}(G) \right\} \quad \text{и} \quad \mathfrak{C}' = \left\{ C_{\mathfrak{X}}^{\mathfrak{Q}'}(G) \,\middle|\, \mathfrak{X} \in \mathcal{K}(G) \right\}.$$

Ввиду замечания 5.6 для доказательства леммы достаточно показать, что

$$\left(\nu_{R,S_0}^{-1}\left(C_{\mathfrak{X}}^{\mathfrak{Q}}(G)\right);\sigma\right) \cong \left(\nu_{R',S_0}^{-1}\left(C_{\mathfrak{X}}^{\mathfrak{Q}'}(G)\right);\sigma\right)$$

при любом $\mathfrak{X} \in \mathcal{K}(G)$. Но это непосредственно следует из того, что для произвольного $\mathfrak{X} \in \mathcal{K}(G)$

$$\left(C^{\mathfrak{Q}}_{\mathfrak{X}}(G);\sigma\right) \cong \left(C^{\mathfrak{Q}'}_{\mathfrak{X}}(G);\sigma\right)$$

по лемме 5.14 и

$$\left(\nu^{-1}_{R,S_0}\left(C^{\mathfrak{Q}}_{\mathfrak{X}}(G)\right);\sigma\right) \cong \left(C^{\mathfrak{Q}}_{\mathfrak{X}}(G);\sigma\right), \quad \left(\nu^{-1}_{R',S_0}\left(C^{\mathfrak{Q}'}_{\mathfrak{X}}(G)\right);\sigma\right) \cong \left(C^{\mathfrak{Q}'}_{\mathfrak{X}}(G);\sigma\right)$$

согласно замечанию 4.3. ■

Напомним, что ввиду сделанных выше предположений и п. 2 леммы 4.1 произвольные ∇-ортогональные разложения пространства V σ-эквивалентны тогда и только тогда, когда они локально σ-эквивалентны.

Теорема 6.2. Пусть $V = \bigoplus \mathfrak{S}$ и $V = \bigoplus \mathfrak{S}'$ — полные ∇-ортогональные разложения пространства V. Тогда

1) если $\operatorname{char} F \neq 2$, то указанные разложения пространства V σ-эквивалентны в том и только в том случае, когда

$$\forall\, x \in F^*/(F^*)^2 \ \left|\{U \in \mathfrak{A} \mid \varkappa(U;\sigma) = x\}\right| = \left|\{U \in \mathfrak{A}' \mid \varkappa(U;\sigma) = x\}\right|, \qquad (61)$$

   где $\mathfrak{A}$ и $\mathfrak{A}'$ — первые компоненты прообразов множеств $\mathfrak{S}$ и $\mathfrak{S}'$ соответственно при биекции (30) из теоремы 5.1 (в данном случае $\mathfrak{A}$ и $\mathfrak{A}'$ состоят лишь из подпространств типа 1);
2) если $\operatorname{char} F = 2$ и $x \nabla x$ для всех $x \in S_1$, то указанные разложения пространства V σ-эквивалентны.

Доказательство. Выберем $(\mathfrak{A}, \mathfrak{B}, R, \mathfrak{C}), (\mathfrak{A}', \mathfrak{B}', R', \mathfrak{C}') \in \mathcal{P}$, такие, что

$$\mathfrak{S} = \mathfrak{A} \sqcup \mathfrak{B} \sqcup \nu^{-1}_{R,S_0}(\mathfrak{C}) \quad \text{и} \quad \mathfrak{S}' = \mathfrak{A}' \sqcup \mathfrak{B}' \sqcup \nu^{-1}_{R',S_0}(\mathfrak{C}'). \qquad (62)$$

Эти четверки существуют и единственны по теореме 5.1. Предположим, что имеет место один из следующих случаев:

1) $\operatorname{char} F \neq 2$ и выполнено условие (61);
2) выполнено условие п. 2, т. е. $\operatorname{char} F = 2$ и $x \nabla x$ для всех $x \in S_1$.

Тогда существует биекция $U \mapsto W_U$ множества $\mathfrak{A}$ на множество $\mathfrak{A}'$, такая, что

$$(U;\sigma) \cong (W_U;\sigma) \qquad (63)$$

для всех $U \in \mathfrak{A}$. В случае 1 из замечания 4.4 вытекает, что в качестве такой биекции можно взять произвольную биекцию $U \mapsto W_U$ множества $\mathfrak{A}$ на множество $\mathfrak{A}'$, удовлетворяющую равенству $\varkappa(U;\sigma) = \varkappa(W_U;\sigma)$ для всех $U \in \mathfrak{A}$ (эти биекции существуют ввиду (61)). В случае 2 $\mathfrak{A}$ и $\mathfrak{A}'$ состоят лишь из подпространств типа 2. Кроме того, очевидно, что $(U;\sigma) \cong (W;\sigma)$ для любых подпространств $U, W \subseteq S_1$ типа 2. Поэтому в качестве такой биекции можно взять произвольную биекцию $\mathfrak{A}$ на $\mathfrak{A}'$ (эти биекции существуют, так как $|\mathfrak{A}| = |\mathfrak{A}'| = \dim(S_1/T)/2$).

Выберем некоторую биекцию $U \mapsto W_U$ множества $\mathfrak{A}$ на множество $\mathfrak{A}'$, удовлетворяющую условию (63) для всех $U \in \mathfrak{A}$. Замечание 6.1 и лемма 6.1 (которая применима ввиду леммы 4.6) позволяют продолжить эту биекцию до биекции $U \mapsto W_U$ множества $\mathfrak{S}$ на множество $\mathfrak{S}'$, удовлетворяющей условию (63) при любом $U \in \mathfrak{S}$ (см. равенства (62)). Таким образом, рассматриваемые разложения пространства V локально σ-эквивалентны и, следовательно, σ-эквивалентны (ввиду п. 2 леммы 4.1 — см. выше).

Пусть теперь $\operatorname{char} F \neq 2$ и рассматриваемые разложения пространства V σ-эквивалентны. Выберем некоторый автоморфизм $\alpha \in \operatorname{Aut}(V;\sigma)$, такой, что $\alpha(\mathfrak{S}) = \mathfrak{S}'$. Очевидно, что $\alpha(S_1) = S_1$ и $\alpha(T) = T$, поэтому $\alpha(\mathfrak{A}) \in \mathcal{A}$. Из равенств (62) вытекает, что $\alpha(\mathfrak{A})$ и $\mathfrak{A}'$ — подмножества $\mathfrak{S}'$, принадлежащие $\mathcal{A}$. Поэтому $\alpha(\mathfrak{A}) = \mathfrak{A}'$ ввиду утверждения о единственности из леммы 5.1. Условие (61) следует теперь из последнего равенства и замечания 4.4. ■

Замечание 6.2. Очевидно, что $(T;\sigma)$ не изоморфно никакому пространству типа 1. Поэтому из замечания 4.4 следует, что если $\mathfrak{A}$ и $\mathfrak{A}'$ принадлежат $\mathcal{A}$ и состоят лишь из подпространств типа 1 ($\operatorname{char} F$ может быть произвольной), то условие (61) (см. п. 1 теоремы 6.2) выполняется тогда и только тогда, когда разложения $S_1 = (\bigoplus \mathfrak{A}) \oplus T$ и $S_1 = (\bigoplus \mathfrak{A}') \oplus T$ пространства S_1 локально σ-эквивалентны или, что в данном случае равносильно, σ-эквивалентны (см. п. 2 леммы 4.1).

Из п. 2 теоремы 6.2 непосредственно вытекает

Следствие 6.1. Если $\operatorname{char} F = 2$ и ортогональность ∇ ассоциирована с некоторой функцией из V в F, то все полные ∇-ортогональные разложения пространства V σ-эквивалентны.

Следующий пример показывает, что при $\operatorname{char} F \neq 2$ аналог следствия 6.1 неверен.

Пример 6.2. Пусть F конечно, $\operatorname{char} F \neq 2$, $\dim V = 3$, а σ определена на всем $V \times V$ и имеет в некотором базисе a_1, a_2, a_3 пространства V единичную матрицу. В этом случае σ ассоциирована с функцией φ, заданной формулой

$$\varphi(x_1a_1 + x_2a_2 + x_3a_3) = \frac{1}{2}\left(x_1^2 + x_2^2 + x_3^2\right), \quad x_1, x_2, x_3 \in F.$$

Выберем произвольные элементы $\theta \in F^* \setminus (F^*)^2$ и $\xi, \eta \in F$, удовлетворяющие равенству $\xi^2 + \eta^2 = \theta$ (такие ξ и η существуют для любого $\theta \in F$ ввиду леммы IV.2 из [4]). Положим $b_1 = a_1$, $b_2 = \xi a_2 + \eta a_3$ и $b_3 = -\eta a_2 + \xi a_3$. Тогда σ в базисе b_1, b_2, b_3 имеет матрицу

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \theta & 0 \\ 0 & 0 & \theta \end{pmatrix}$$

(см. лемму IV.3 из [4]). Поэтому $V = \langle a_1\rangle \oplus \langle a_2\rangle \oplus \langle a_3\rangle$ и $V = \langle b_1\rangle \oplus \langle b_2\rangle \oplus \langle b_3\rangle$ — полные ∇-ортогональные разложения пространства V. Покажем, что эти разложения не только не σ-эквивалентны (это вытекает из замечания 4.4), но и не ∇-эквивалентны. Пусть, напротив, некоторый автоморфизм $\alpha \in \operatorname{Aut}(V;\nabla)$ удовлетворяет равенству

$$\alpha(\{\langle a_1\rangle, \langle a_2\rangle, \langle a_3\rangle\}) = \{\langle b_1\rangle, \langle b_2\rangle, \langle b_3\rangle\}.$$

Пусть также перестановка β множества $\{1,2,3\}$ такова, что $\alpha\left(\left\langle a_{\beta(i)}\right\rangle\right) = \langle b_i\rangle$ для всех $i \in \{1,2,3\}$. Тогда $\alpha\left(a_{\beta(i)}\right) = \lambda_i b_i$ при некоторых $\lambda_i \in F^*$ $(i = 1,2,3)$. Очевидно, что $a_{\beta(1)} + a_{\beta(2)} \nabla -a_{\beta(1)} + a_{\beta(2)}$ и, следовательно, $\alpha\left(a_{\beta(1)} + a_{\beta(2)}\right) \nabla \alpha\left(-a_{\beta(1)} + a_{\beta(2)}\right)$. В то же время

$$\alpha\left(a_{\beta(1)} + a_{\beta(2)}\right) \nabla \alpha\left(-a_{\beta(1)} + a_{\beta(2)}\right) \iff -\lambda_1^2 + \theta\lambda_2^2 = 0 \iff \theta = \left(\frac{\lambda_1}{\lambda_2}\right)^2,$$

а последнее равенство противоречит выбору θ. Таким образом, рассматриваемые разложения не ∇-эквивалентны.

Пример 6.2 показывает, что при $\operatorname{char} F \neq 2$ полные $\triangledown$-ортогональные разложения пространства V могут не быть даже $\triangledown$-эквивалентными. Следующая теорема утверждает, что в данном случае для любых двух таких разложений имеет место локальная $(\diamondsuit, \triangledown)$-эквивалентность.

Теорема 6.3. Если $\operatorname{char} F \neq 2$, то все полные $\triangledown$-ортогональные разложения пространства V локально $(\diamondsuit, \triangledown)$-эквивалентны.

Доказательство. Пусть $V = \bigoplus \mathfrak{S}$ и $V = \bigoplus \mathfrak{S}'$ — произвольные полные $\triangledown$-ортогональные разложения пространства V. Как и в доказательстве теоремы 6.2, выберем $(\mathfrak{A}, \mathfrak{B}, R, \mathfrak{C}), (\mathfrak{A}', \mathfrak{B}', R', \mathfrak{C}') \in \mathcal{P}$, такие, что

$$\mathfrak{S} = \mathfrak{A} \sqcup \mathfrak{B} \sqcup \nu^{-1}_{R,S_0}(\mathfrak{C}) \quad \text{и} \quad \mathfrak{S}' = \mathfrak{A}' \sqcup \mathfrak{B}' \sqcup \nu^{-1}_{R',S_0}(\mathfrak{C}'). \tag{64}$$

Эти четверки существуют и единственны по теореме 5.1. Множества $\mathfrak{A}$ и $\mathfrak{A}'$ состоят лишь из подпространств типа 1. Кроме того, легко видеть, что $(U; \diamondsuit, \triangledown) \cong (W; \diamondsuit, \triangledown)$ для любых подпространств $U, W \subseteq S_1$ типа 1. Поэтому биекции $U \mapsto W_U$ множества $\mathfrak{A}$ на множество $\mathfrak{A}'$ существуют (так как $|\mathfrak{A}| = |\mathfrak{A}'| = \dim(S_1/T)$) и удовлетворяют условию

$$(U; \diamondsuit, \triangledown) \cong (W_U; \diamondsuit, \triangledown) \tag{65}$$

для всех $U \in \mathfrak{A}$. Замечание 6.1 и лемма 6.1 (которая применима ввиду леммы 4.6) позволяют продолжить произвольную такую биекцию до биекции $U \mapsto W_U$ множества $\mathfrak{S}$ на множество $\mathfrak{S}'$, удовлетворяющей условию (65) при любом $U \in \mathfrak{S}$ (см. равенства (64)). Таким образом, рассматриваемые разложения пространства V локально $(\diamondsuit, \triangledown)$-эквивалентны. ■

Из теоремы 6.3 вытекает, что при $\operatorname{char} F \neq 2$ мультимножество размерностей подпространств, входящих в полное $\triangledown$-ортогональное разложение пространства V, одно и то же для всех таких разложений. Следующий пример показывает, что при $\operatorname{char} F = 2$ это не всегда верно.

Пример 6.3. Пусть $\operatorname{char} F = 2$, $\dim V = 3$, а σ определена на всем $V \times V$ и имеет в некотором базисе a_1, a_2, a_3 пространства V единичную матрицу. Положим $b_1 = a_1 + a_2 + a_3$, $b_2 = a_1 + a_3$ и $b_3 = a_2 + a_3$. Тогда σ в базисе b_1, b_2, b_3 имеет матрицу

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

(см. лемму IV.5 из [4]). Поэтому $V = \langle a_1 \rangle \oplus \langle a_2 \rangle \oplus \langle a_3 \rangle$ и $V = \langle b_1 \rangle \oplus \langle b_2, b_3 \rangle$ — полные $\triangledown$-ортогональные разложения пространства V (см. лемму 4.2; здесь $\langle b_2, b_3 \rangle$ имеет тип 2).

Следующий пример показывает, что при $\operatorname{char} F = 2$ утверждение, аналогичное п. 1 теоремы 6.2 (в предположении, что $\mathfrak{A}$ и $\mathfrak{A}'$ состоят лишь из подпространств типа 1), неверно.

Пример 6.4. Предположим, что $\operatorname{char} F = 2$. Пусть V_1 — одномерное векторное пространство, a — некоторый порождающий элемент этого пространства, а симметричная билинейная функция $\sigma_1 \colon V_1 \times V_1 \to F$ определена равенством $\sigma_1(a, a) = 1$. Обозначим через $(V_2; \sigma_2)$ некоторое пространство с частичной симметричной билинейной функцией, построенное в соответствии с конструкцией примера 6.1, где дополнительно предполагаем, что $U \neq \{0\}$. Пусть теперь $(V; \sigma)$ — прямое ортогональное произведение

семейства $((V_i;\sigma_i)\,|\,i\in\{1,2\})$ пространств с частичными симметричными билинейными функциями. Ввиду замечания 3.4 пространство с ортогональностью $(V;\lozenge)$ удовлетворяет условию (13). Кроме того, $0\lozenge V$. Будем считать V_1 и V_2 подпространствами V, отождествляя v_1 с $(v_1,0)$ для любого $v_1\in V_1$ и v_2 с $(0,v_2)$ для любого $v_2\in V_2$. Тогда $V=V_1\oplus V_2$ — $\triangledown$-ортогональное разложение пространства V. Из леммы 3.1 вытекает, что

$$S_0=\{0\},\quad S_1=V_1\oplus U\quad\text{и}\quad T=U.$$

Выберем произвольный элемент $u\in U\setminus\{0\}$ и положим $V_1'=\langle a+u\rangle$. Очевидно, что $\{V_1\},\{V_1'\}\in\mathcal{A}$, причем V_1 и V_1' имеют тип 1 и $\varkappa(V_1;\sigma)=\varkappa(V_1';\sigma)=1$ в $F^*/(F^*)^2$ (так как $\sigma(a,a)=\sigma(a+u,a+u)=1$). Легко видеть, что $G(\{V_1\})=V_2$ (см. лемму 4.4) и $G(\{V_1'\})=\{w-\sigma(w,u)a\,|\,w\in V_2\}$; последнее подпространство обозначим для краткости через V_2'. Непосредственно проверяется, что отображение $w\mapsto w-\sigma(w,u)a$, где $w\in V_2$, является изоморфизмом $(V_2;\lozenge)$ на $(V_2';\lozenge)$. Кроме того, всякий изоморфизм $(V_2;\lozenge)$ на $(V_2';\lozenge)$ отображает

$$T=\mathsf{C}^{\lozenge}_{V_2}(V_2)=\mathsf{C}^{\lozenge}_{V_2'}(V_2')$$

(см. (27)) на себя. Поэтому $(V_2/T;\lozenge)\cong(V_2'/T;\lozenge)$ и пространство V_2'/T $\lozenge$-неразложимо, так как $\lozenge$-неразложимо пространство V_2/T (последнее условие ввиду следствия 5.2 эквивалентно равенству $\mathfrak{I}(V_2)=\{I(V_2)\}$). Следствие 5.3 показывает теперь, что V_2', как и V_2, $\triangledown$-неразложимо. Таким образом, $V=V_1\oplus V_2$ и $V=V_1'\oplus V_2'$ — полные $\triangledown$-ортогональные разложения пространства V (см. равенство (26)), причем первые компоненты прообразов множеств $\{V_1,V_2\}$ и $\{V_1',V_2'\}$ при биекции (30) из теоремы 5.1 равны соответственно $\{V_1\}$ и $\{V_1'\}$ (это вытекает из леммы 5.1). В то же время $(V_2;\triangledown)$ и $(V_2';\triangledown)$ не изоморфны, так как $x\triangledown x$ для любого $x\in V_2$, но существует $y\in V_2'$, такой, что $y\not\triangledown y$ (в качестве такого y можно взять произвольный элемент $V_2'\setminus V_2$). Следовательно, указанные разложения пространства V не локально $\triangledown$-эквивалентны и тем более не σ-эквивалентны.

6.3. Критерий единственности полного $\triangledown$-ортогонального разложения

Теорема 6.4. Полное $\triangledown$-ортогональное разложение пространства V единственно тогда и только тогда, когда выполнено одно из следующих взаимоисключающих условий:

1) S_1 — пространство типа 1 или 2;
2) $F=\mathrm{GF}(2)$ и σ в некотором базисе пространства S_1 имеет матрицу $\begin{pmatrix}1&0\\0&1\end{pmatrix}$;
3) $F=\mathrm{GF}(3)$ и σ в некотором базисе пространства S_1 имеет матрицу $\begin{pmatrix}1&0\\0&-1\end{pmatrix}$;
4) V одномерно и σ тождественно равна 0 на $V\times V$;
5) $S_0=\{0\}$, σ тождественно равна 0 на $S_1\times S_1$ и

$$\left|\left\{\mathfrak{X}\in\mathcal{K}(V)\;\middle|\;\bigoplus_{X\in\mathfrak{X}}H_X(V)\neq\{0\}\right\}\right|\leqslant 1. \tag{66}$$

Доказательство. Из теоремы 5.2 вытекает, что полное ∇-ортогональное разложение пространства V единственно тогда и только тогда, когда

$$|\mathcal{A}| = 1; \tag{67}$$

$$|\mathcal{B}| = 1; \tag{68}$$

$$|\mathfrak{R}(G(\mathfrak{A}))| = 1; \tag{69}$$

$$\forall\, \mathfrak{Q}, \mathfrak{Q}' \in \mathcal{Q}(G(\mathfrak{A}))\ \forall\, \mathfrak{X} \in \mathcal{K}(G(\mathfrak{A}))\ C^{\mathfrak{Q}}_{\mathfrak{X}}(G(\mathfrak{A})) = C^{\mathfrak{Q}'}_{\mathfrak{X}}(G(\mathfrak{A})), \tag{70}$$

где $\mathfrak{A}$ — единственный элемент множества $\mathcal{A}$ (см. условие (67)). Поэтому для доказательства теоремы достаточно показать, что условия (67)–(70) выполнены, если и только если выполнено одно из условий, указанных в формулировке теоремы.

I. Предположим, что выполнены условия (67)–(70); при этом через $\mathfrak{A}$ будем обозначать единственный элемент множества $\mathcal{A}$. Пусть сначала $\mathfrak{A} \neq \varnothing$. Тогда ввиду равенства (67) и леммы 4.8, примененной к пространству S_1, получаем, что $T = \{0\}$ и всякое подпространство пространства S_1, имеющее тип 1 или 2, принадлежит $\mathfrak{A}$. Если $|\mathfrak{A}| = 1$, то выполнено условие 1. Считаем теперь, что $|\mathfrak{A}| \geqslant 2$. Предположим, что $\mathfrak{A}$ содержит подпространство U типа 2. Выберем базис u_1, u_2 подпространства U, в котором σ имеет матрицу $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Так как $|\mathfrak{A}| \geqslant 2$, можно выбрать подпространство $W \in \mathfrak{A} \setminus \{U\}$. Рассмотрим два случая.

1. W имеет тип 1, т. е. $W = \langle w \rangle$, где $w \not\perp w$. Тогда $\langle w + u_1 \rangle$ — подпространство S_1, имеющее тип 1 и не принадлежащее $\mathfrak{A}$.
2. W имеет тип 2, т. е. матрица симметричной билинейной функции σ в некотором базисе w_1, w_2 этого подпространства равна $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Тогда $\langle u_1 + w_1, u_2 \rangle$ — подпространство S_1, имеющее тип 2 и не принадлежащее $\mathfrak{A}$.

В обоих случаях получаем противоречие с доказанным выше свойством. Это показывает, что $\mathfrak{A}$ состоит только из подпространств типа 1.

Предположим, что существуют $u, w \in S_1$ и $\lambda \in F^*$, такие, что $\langle u \rangle$ и $\langle w \rangle$ — различные подпространства из $\mathfrak{A}$ и $\lambda^2 \neq -\sigma(u,u)/\sigma(w,w)$. Тогда $\langle u + \lambda w \rangle$ — подпространство S_1, имеющее тип 1 и не принадлежащее $\mathfrak{A}$. Полученное противоречие показывает, что $\lambda^2 = -\sigma(u,u)/\sigma(w,w)$ для любых $u, w \in S_1$, порождающих различные подпространства из $\mathfrak{A}$, и любого $\lambda \in F^*$. Следовательно,

$$\forall\, \lambda \in F^*\ \lambda^2 = 1; \tag{71}$$

$$\forall\, u, w \in S_1\ (\langle u \rangle, \langle w \rangle \in \mathfrak{A}\ \ \&\ \ \langle u \rangle \neq \langle w \rangle\ \implies\ \sigma(u,u) = -\sigma(w,w)). \tag{72}$$

Из (71) вытекает, что $F = \mathrm{GF}(2)$ или $F = \mathrm{GF}(3)$. Если $F = \mathrm{GF}(2)$, то σ в некотором базисе пространства S_1 имеет единичную матрицу (размера $|\mathfrak{A}| \times |\mathfrak{A}|$). При этом $|\mathfrak{A}| \leqslant 2$ (иначе конструкция примера 6.3 дает противоречие с равенством (67)). Поэтому $|\mathfrak{A}| = 2$ и выполняется условие 2. Если же $F = \mathrm{GF}(3)$, то из (72) следует, что для любого $\varepsilon \in \{-1, 1\}$ множество $\mathfrak{A}$ содержит не более одного подпространства $\langle u \rangle$, удовлетворяющего равенству $\sigma(u,u) = \varepsilon$ (здесь $\sigma(u,u)$ не зависит от выбора порождающего u ввиду (71)). Это вместе с условием $|\mathfrak{A}| \geqslant 2$ показывает, что в данном случае выполняется условие 3.

Считаем теперь, что $\mathfrak{A} = \varnothing$. Тогда σ тождественно равна 0 на $S_1 \times S_1$ и $G(\mathfrak{A}) = V$. Если $\dim S_0 \geqslant 2$, то получаем противоречие с равенством (68). Поэтому $\dim S_0 \leqslant 1$. Пусть сначала $\dim S_0 = 1$. Если $S_0 \neq G(\mathfrak{A})$, то имеет место противоречие с равенством (69). Таким образом, в данном случае $S_0 = G(\mathfrak{A}) = V$ и выполняется условие 4.

Рассмотрим теперь случай, когда $\dim S_0 = 0$, т. е. $S_0 = \{0\}$. Предположим, что неравенство (66) не выполняется. Тогда можем выбрать $\mathfrak{Z}_i \in \mathcal{K}(V)$, $Z_i \in \mathfrak{Z}_i$ и $w_i \in H_{Z_i}(V) \setminus \{0\}$ для $i = 1, 2$, где $\mathfrak{Z}_1 \neq \mathfrak{Z}_2$. Выберем также какое-либо семейство $\mathfrak{Q} = (Q_X \,|\, X \in \mathfrak{K}(V)) \in \mathcal{Q}(V)$ (из п. 2 леммы 5.8 вытекает, что $\mathcal{Q}(V) \neq \varnothing$). Пусть $i \in \{1, 2\}$. Очевидно, что $x \mapsto \sigma(x, w_i)$ (где $x \in Q_{Z_i}$) — линейная функция из Q_{Z_i} в F. Эта функция отлична от тождественного нуля на Q_{Z_i}, так как $w_i \not\nabla \rho^{-1}(\langle Z_i \rangle) = Q_{Z_i} \oplus T$ (ввиду леммы 5.3 и равенства (37)) и $w_i \nabla T$ (согласно (33)). Поэтому можем выбрать базис D_{Z_i} пространства Q_{Z_i}, такой, что $\sigma(c_i, w_i) = 1$ для некоторого $c_i \in D_{Z_i}$ и $\sigma(a, w_i) = 0$ при всех $a \in D_{Z_i} \setminus \{c_i\}$. Для любого $X \in \mathfrak{K}(V) \setminus \{Z_1, Z_2\}$ выберем произвольное линейно независимое множество D_X, удовлетворяющее равенству $\rho^{-1}(\langle X \rangle) = \langle D_X \rangle \oplus T$. Очевидно, что если $\{X, Y\} = \{Z_1, Z_2\}$, то система уравнений (44) приобретает вид

$$\sigma(a, u_{b,Z_1}) + \sigma(b, u_{a,Z_2}) = 0, \quad a \in D_{Z_1},\ b \in D_{Z_2}. \tag{73}$$

Построим теперь семейство $\mathfrak{Q}' = (\langle E_X \rangle \,|\, X \in \mathfrak{K}(V)) \in \mathcal{Q}(V)$ в соответствии с леммой 5.8, используя выбранные выше множества D_X. При этом в качестве решения системы (73) выбираем $(u_{a,Z_2}, u_{b,Z_1} \,|\, a \in D_{Z_1},\ b \in D_{Z_2})$, где

$$u_{a,Z_2} = \begin{cases} w_2, & \text{если } a = c_1, \\ 0, & \text{если } a \in D_{Z_1} \setminus \{c_1\}; \end{cases} \qquad u_{b,Z_1} = \begin{cases} -w_1, & \text{если } b = c_2, \\ 0, & \text{если } b \in D_{Z_2} \setminus \{c_2\}. \end{cases}$$

Непосредственно проверяется, что эти элементы действительно образуют решение системы (73). Выбор всех остальных элементов, требуемых в лемме 5.8, несуществен. Положим теперь $v = c_1 + \sum\limits_{Y \in \mathfrak{K}(V)} u_{c_1,Y}$. Тогда $v \in E_{Z_1} \subseteq C^{\mathfrak{Q}'}_{\mathfrak{Z}_1}(V)$. В то же время если $v \in C^{\mathfrak{Q}}_{\mathfrak{Z}_1}(V)$, то $u_{c_1,Y} = 0$ для всех $Y \in \mathfrak{K}(V) \setminus \mathfrak{Z}_1$ ввиду того, что сумма Q_X и $H_X(V)$ по всем $X \in \mathfrak{K}(V)$ прямая (см. доказательство леммы 5.12). В частности, в этом случае $u_{c_1,Y} = 0$ при любом $Y \in \mathfrak{Z}_2$, что неверно, так как $u_{c_1,Z_2} = w_2 \neq 0$. Таким образом, $v \notin C^{\mathfrak{Q}}_{\mathfrak{Z}_1}(V)$ и $C^{\mathfrak{Q}'}_{\mathfrak{Z}_1}(V) \neq C^{\mathfrak{Q}}_{\mathfrak{Z}_1}(V)$.

II. Предположим теперь, что выполнено одно из условий 1–3. Тогда $T = \{0\}$, так как σ невырождена на S_1. Если выполнено условие 1, то $\mathcal{A} = \{\mathfrak{A}\}$, где $\mathfrak{A} = \{S_1\}$ (см. лемму 4.2). Если же выполнено условие 2 или 3, то легко видеть, что $\mathcal{A} = \{\mathfrak{A}\}$, где $\mathfrak{A}$ — множество всех подпространств S_1, имеющих тип 1. Отметим, что если a, b — произвольный базис S_1, в котором σ имеет указанную в условии 2 или 3 матрицу, то $\mathfrak{A} = \{\langle a \rangle, \langle b \rangle\}$ (последнее множество не зависит от выбора указанного базиса a, b). Из включения $S_0 \subseteq T$ и равенства $T = \{0\}$ следует равенство $S_0 = \{0\}$. Поэтому $\mathcal{B} = \{\varnothing\}$ и $\mathfrak{R}(G(\mathfrak{A})) = \{G(\mathfrak{A})\}$. Кроме того, очевидно, что $\mathcal{Q}(G(\mathfrak{A})) = \{(\langle X \rangle \,|\, X \in \mathfrak{K}(G(\mathfrak{A})))\}$. Таким образом, в данном случае условия (67)–(70) выполнены.

Если выполнено условие 4, то $S_0 = T = S_1 = V$, $\mathcal{A} = \{\varnothing\}$, $\mathcal{B} = \{\{V\}\}$, $\mathfrak{R}(V) = \{\{0\}\}$, а множества $I(V)$, $\mathfrak{I}(V)$, $\mathcal{I}(V)$, $\mathfrak{K}(V)$ и $\mathcal{K}(V)$ пусты (здесь $V = G(\varnothing)$). Поэтому в данном случае условия (67)–(70) выполнены. Впрочем, в случае выполнения условия 4 единственность полного ∇-ортогонального разложения пространства V тривиальна. Пусть теперь выполнено условие 5. Тогда $\mathcal{A} = \{\varnothing\}$, $\mathcal{B} = \{\varnothing\}$ и $\mathfrak{R}(V) = \{V\}$, где $V = G(\varnothing)$. Это показывает, что выполнены условия (67)–(69). Осталось доказать, что условие (70) также выполнено. Если $T = \{0\}$, то это вытекает из легко проверяемого равенства $\mathcal{Q}(V) = \{(\langle X \rangle \,|\, X \in \mathfrak{K}(V))\}$. Считаем теперь, что $T \neq \{0\}$. Обозначим через $\mathfrak{L}$ элемент множества $\mathcal{K}(V)$, удовлетворяющий равенству $T = \bigoplus\limits_{X \in \mathfrak{L}} H_X(V)$

(существование и единственность такого элемента следуют из леммы 5.5). Выберем произвольные семейства $\mathfrak{Q} = (Q_X \mid X \in \mathfrak{K}(V))$ и $\mathfrak{Q}' = (Q'_X \mid X \in \mathfrak{K}(V))$, принадлежащие $\mathcal{Q}(V)$, и покажем, что

$$C^{\mathfrak{Q}}_{\mathfrak{X}}(V) = C^{\mathfrak{Q}'}_{\mathfrak{X}}(V) \tag{74}$$

для любого $\mathfrak{X} \in \mathcal{K}(V)$. Если $\mathfrak{X} = \mathfrak{L}$, то равенство (74) верно ввиду того, что обе его части равны $\sum\limits_{X \in \mathfrak{L}} \rho^{-1}(\langle X \rangle)$. Пусть теперь $\mathfrak{X} \in \mathcal{K}(V) \setminus \{\mathfrak{L}\}$. Для каждого $X \in \mathfrak{K}(V)$ выберем некоторый базис D_X пространства Q_X. Воспользовавшись замечанием 5.4, выберем также в конструкции леммы 5.8 элементы $u_{a,Y} \in H_Y(V)$ ($a \in D_X$, $X, Y \in \mathfrak{K}(V)$) таким образом, что

$$Q'_X = \left\langle a + \sum_{Y \in \mathfrak{K}(V)} u_{a,Y} \;\middle|\; a \in D_X \right\rangle = \left\langle a + \sum_{Y \in \mathfrak{L}} u_{a,Y} \;\middle|\; a \in D_X \right\rangle \tag{75}$$

для любого $X \in \mathfrak{K}(V)$ (здесь последнее равенство вытекает из того, что $u_{a,Y} = 0$ при $Y \in \mathfrak{K}(V) \setminus \mathfrak{L}$). Пусть $a \in D_X$, где $X \in \mathfrak{X}$, и $Y \in \mathfrak{L}$. Тогда

$$\sigma(b, u_{a,Y}) = \sigma(a, u_{b,X}) + \sigma(b, u_{a,Y}) = -\sigma(a, b) = 0$$

для любого $b \in D_Y$, так как $u_{b,X} = 0$ и $\sigma(a, b) = 0$. Поэтому $u_{a,Y} \triangledown D_Y$ и, следовательно, $u_{a,Y} \triangledown \rho^{-1}(Y)$, так как $u_{a,Y} \triangledown T$ (см. (33)) и $\rho^{-1}(Y) \subseteq \rho^{-1}(\langle Y \rangle) = \langle D_Y \rangle \oplus T$. С другой стороны, $u_{a,Y} \triangledown \rho^{-1}(I(V) \setminus Y)$, так как $u_{a,Y} \in H_Y(V)$. Это показывает, что $\Xi_{I(V)}(u_{a,Y}) = \varnothing$. Поэтому $u_{a,Y} = 0$ ввиду леммы 5.3. Из (75) вытекает теперь, что $Q'_X = Q_X$ для любого $X \in \mathfrak{X}$, откуда непосредственно следует требуемое равенство (74). Таким образом, условие (70) выполнено и в случае выполнения условия 5. ■

Замечание 6.3. Из леммы 5.5 следует, что если в условии 5 теоремы 6.4 заменить неравенство (66) на дизъюнкцию

$$T = \{0\} \vee \exists\, \mathfrak{L} \in \mathcal{K}(V)\ T = \bigoplus_{X \in \mathfrak{L}} H_X(V) \neq \{0\},$$

то полученный вариант условия 5 будет эквивалентен исходному.

6.4. Вычислительные аспекты

Будем считать, что поле F конечно, а пространство V конечномерно (и, следовательно, конечно). Носители F и V предполагаются содержащимися в множестве всех конечных строк, состоящих из нулей и единиц. Поле F подается на вход алгоритмам в виде набора, состоящего из носителя этого поля и таблиц сложения и умножения в F. Аналогично, пространство V задается для алгоритмов с помощью носителя этого пространства вместе с таблицами сложения в V и умножения элементов V на элементы поля F. Частичная симметричная билинейная функция σ подается на вход алгоритмам в виде таблицы своих значений; эта же таблица задает область определения $\lozenge$ функции σ и ортогональность $\triangledown$. Кроме того, произвольная функция $\varphi\colon V \to F$ также подается на вход алгоритмам в виде таблицы своих значений.

Замечание 6.4. Очевидно, что существует полиномиальный детерминированный алгоритм, который по произвольным F, V и функции $\varphi\colon V \to F$ вычисляет таблицу значений частичной функции σ_φ, определенной по φ согласно (1).

Теорема 6.5. Существует полиномиальный недетерминированный алгоритм, множество всех выходных значений которого при вычислении на произвольном входе вида (F, V, σ) совпадает с множеством всех $\mathfrak{S}$, таких, что $V = \bigoplus \mathfrak{S}$ — полное $\triangledown$-ортогональное разложение пространства V.

Доказательство. Работа искомого алгоритма заключается в выборе произвольных $\mathfrak{A} \in \mathcal{A}$, $\mathfrak{B} \in \mathcal{B}$, $R \in \mathfrak{R}(G(\mathfrak{A}))$, $\mathfrak{Q} \in \mathcal{Q}(G(\mathfrak{A}))$ и возвращении на выходе множества

$$\mathfrak{A} \sqcup \mathfrak{B} \sqcup \left\{ \nu_{R,S_0}^{-1}\left(C_{\mathfrak{X}}^{\mathfrak{Q}}(G(\mathfrak{A}))\right) \mid \mathfrak{X} \in \mathcal{K}(G(\mathfrak{A}))\right\}$$

(см. теорему 5.2). При этом для построения $\mathfrak{A}$ и $\mathfrak{Q}$ следует воспользоваться методами, описанными в замечании 4.5 и лемме 5.8 (вместе с замечанием 5.3) соответственно. (Согласно замечаниям 4.5 и 5.4, эти методы позволяют построить все множества $\mathfrak{A} \in \mathcal{A}$ и все семейства $\mathfrak{Q} \in \mathcal{Q}(G(\mathfrak{A}))$.) Легко видеть, что все действия, требуемые в приведенном выше описании алгоритма, могут быть выполнены за полиномиальное время. (Смежность вершин в графе $\Delta(G(\mathfrak{A}))$ проверяема за полиномиальное время ввиду леммы 5.7 и замечания 5.3.) ■

Приведем теперь две теоремы, первая из которых (теорема 6.6) непосредственно следует из теорем 6.1 и 6.4, а вторая (теорема 6.7) — из п. 1 теоремы 6.2.

Теорема 6.6. Следующие условия могут быть проверены с помощью полиномиального детерминированного алгоритма, на вход которому подается произвольная тройка вида (F, V, σ):

1) ∇-неразложимость V;
2) единственность полного ∇-ортогонального разложения пространства V.

Теорема 6.7. Существует полиномиальный детерминированный алгоритм, который для произвольного входа вида $(F, V, \sigma, \mathfrak{S}, \mathfrak{S}')$, такого, что $\operatorname{char} F \neq 2$, а $V = \bigoplus \mathfrak{S}$ и $V = \bigoplus \mathfrak{S}'$ — полные ∇-ортогональные разложения пространства V, проверяет, являются ли эти разложения σ-эквивалентными.

Отметим, что в замечании 6.4 и теоремах 6.5–6.7 полиномиальность (полиномиальное время работы) алгоритмов означает, что время их работы на указанных входах не превосходит $\mathrm{O}((l(|F| + |V|))^c)$ для некоторой константы $c > 0$, где l — максимальная длина строк из $F \cup V$.

Автор благодарит О. А. Логачёва, А. А. Сальникова и В. В. Ященко за постановку первоначальной проблемы (см. задачу 2.17 из книги [3]) и признателен математикам Института проблем информационной безопасности МГУ им. М. В. Ломоносова (в котором он имеет честь работать), а также руководителям и участникам семинара «Кольца и модули» кафедры высшей алгебры механико-математического факультета указанного университета за полезные обсуждения настоящей работы.

ЛИТЕРАТУРА

1. *Анохин М. И.* О расщепляемости p-ичных функций // Матем. сб. 2007. Т. 198. № 7. С. 31–44.
2. *Логачёв О. А., Сальников А. А., Ященко В. В.* Некоторые характеристики «нелинейности» групповых отображений // Дискрет. анализ и исслед. опер. 2001. Т. 8. № 1. С. 40–54.
3. *Логачёв О. А., Сальников А. А., Ященко В. В.* Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2004.
4. *Newman M.* Integral matrices. New York and London: Academic Press, 1972.

DOI 10.17223/20710410/15/3
УДК 519.6

О РАЗВЕТВЛЕНИЯХ КРИПТОГРАФИЧЕСКИХ ФУНКЦИЙ НА ПРЕОБРАЗОВАНИЯ С ЗАДАННЫМ ПРИЗНАКОМ

К. Г. Когос, В. М. Фомичев

Национальный исследовательский ядерный университет (МИФИ), г. Москва, Россия

E-mail: fomichev@nm.ru

Понятийный аппарат, связанный с разветвлением функций на линейные подфункции, обобщен для разветвлений функций специального вида на преобразования, имеющие заданный признак. Получены оценки характеристик для разветвлений функций, реализуемых генераторами «δ–τ шагов» и с перемежающимся шагом.

Ключевые слова: *степень разветвления, признак в полугруппе, генераторы гаммы с неравномерным движением, генератор с перемежающимся шагом.*

Анализ криптографических систем алгебраическими методами состоит в разработке методов решения систем алгебраических уравнений (САУ) и оценке вычислительной сложности их решения. Известно, что от решения нелинейной САУ можно перейти к решению нескольких линейных систем (СЛАУ) путем фиксации константами нескольких переменных. Вместе с тем не только линеаризация систем уравнений представляет интерес для анализа криптографических свойств, важно свести САУ к системам уравнений определенного вида, левая часть которых задает преобразования, имеющие заданный признак [1, гл. 9].

Целью работы является обобщение понятийного аппарата, связанного с разветвлением функций на линейные подфункции [2], для разветвлений функций специального вида на преобразования, имеющие заданный признак, и исследование криптографических характеристик функций, реализуемых в некоторых генераторах гаммы с неравномерным шагом.

При натуральных n, m рассмотрим множество $\Phi_{n,m}$ всех отображений $V^n \to V^m$, $n > m$, $V = \{0, 1\}$. Пусть функция $g \in \Phi_{n,m}$ задается системой координатных функций $\{g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)\}$. Определим умножение булева монома $\mu = x_{i_1} \cdot \ldots \cdot x_{i_p}$ степени p на отображение $g \in \Phi_{n,m}$:

$$\mu \cdot g(x_1, \ldots, x_n) = \{\mu \cdot g_1(x_1, \ldots, x_n), \ldots, \mu \cdot g_m(x_1, \ldots, x_n)\}.$$

Обозначим через $g^{\alpha_1,\ldots,\alpha_k}_{x_1,\ldots,x_k}(x_{k+1}, \ldots, x_n)$ ограничение функции g на множество $(\alpha_1, \ldots, \alpha_k, x_{k+1}, \ldots, x_n)$. Здесь $\alpha_1, \ldots, \alpha_k$ — фиксированные значения; $x_{k+1}, \ldots, x_n$ — булевы переменные. Сформулируем обобщение теоремы о разложении булевой функции по переменным на отображения из множества $\Phi_{n,m}$.

Теорема 1. Для любой функции $g \in \Phi_{n,m}$ выполнено равенство

$$g(x_1, \ldots, x_n) = \bigoplus_{(\sigma_1,\ldots,\sigma_k) \in V^k} x_1^{\sigma_1} \cdot \ldots \cdot x_k^{\sigma_k} \cdot g^{\sigma_1,\ldots,\sigma_k}_{x_1,\ldots,x_k}(x_{k+1}, \ldots, x_n).$$

Итак, множество подфункций $\{g(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n) : (\sigma_1, \ldots, \sigma_k) \in V^k\}$ однозначно задает отображение $g : V^n \to V^m$ при любом k и любом наборе фиксируемых

координат. Будем говорить, что отображение $g \in \Phi_{n,m}$ разветвляется на отображения $g^{\alpha_1,\ldots,\alpha_k}_{x_1,\ldots,x_k}(x_{k+1},\ldots,x_n)$.

Рассмотрим случай, когда фиксируется $k = n - m$ переменных. Тогда отображение g разветвляется на преобразования множества V^m. Набор номеров фиксируемых переменных обозначим I, то есть $I = \{i_1,\ldots,i_{n-m}\}$, где $\{i_1,\ldots,i_{n-m}\} \subseteq \{1,\ldots,n\}$, а набор фиксируемых переменных обозначим $x(I)$, то есть $x(I) = \{x_{i_1},\ldots,x_{i_{n-m}}\}$.

Определение 1. *Степенью I-разветвления отображения $g \in \Phi_{n,m}$* называется число различных преобразований $V^m \to V^m$, полученных при фиксации всевозможными наборами двоичных констант набора переменных $x(I)$; обозначается $æ_{I(g)}$.

Напомним основные определения теории признаков в полугруппах [1, гл. 9].

Пусть G — конечная полугруппа, ее систему образующих обозначим через $S = \{s_1,\ldots,s_p\} \subset G$, $p \in N$, $G = \langle S\rangle$. Известно, что произвольный элемент g полугруппы G представляется словом в алфавите S. Признаком H (H-признаком) в полугруппе G называется подмножество H полугруппы G, состоящее из всех элементов G, обладающих определенным свойством. Элемент g полугруппы G имеет признак H, если $g \in H$. Непустое множество $R \subseteq G$ имеет H-признак, если $R \cap H \neq \varnothing$; R не имеет H-признака или имеет пустой H-признак, если $R \cap H = \varnothing$. Признак H в R называется тривиальным, если $|R \cap H| = 1$ (нетривиальным, если $|R \cap H| > 1$).

Определение 2. Пусть при всех фиксациях набора переменных $x(I)$ функция $g \in \Phi_{n,m}$ разветвляется только на преобразования, имеющие признак H. Тогда набор $x(I)$ назовем *вполне H-сводящим функцию g*, а число различных преобразований $V^m \to V^m$, полученных при фиксации всевозможными наборами двоичных констант набора переменных $x(I)$ и имеющих признак H, назовем *степенью (I,H)-разветвления функции g* и обозначим $æ_{I(g,H)}$.

Определение 3. Пусть при всех фиксациях набора переменных $x(I)$ функция $g \in \Phi_{n,m}$ разветвляется только на линейные преобразования. Тогда набор $x(I)$ назовем *вполне линеаризующим функцию g*, а число различных линейных преобразований $V^m \to V^m$, полученных при фиксации всевозможными наборами двоичных констант набора переменных $x(I)$, назовем *степенью линейного I-разветвления функции g* и обозначим $\lambda_I(g)$.

Пусть $N_n = \{1,\ldots,n\}$, где n — натуральное число.

Определение 4. *Степенью разветвления функции g* называется $\min\limits_{I\subseteq N_n}\{æ_I(g)\}$, где минимум берется по всем наборам $x(I)$ порядка $n - m$, эта величина обозначается $æ(g)$.

Определение 5. *Степенью H-разветвления функции g* называется $\min\limits_{I\subseteq N_n}\{æ_I(g,H)\}$, где минимум берется по всем вполне H-сводящим наборам $x(I)$ порядка $n - m$, эта величина обозначается $æ(g,H)$.

Определение 6. *Степенью линейного разветвления функции g* называется $\min\limits_{I\subseteq N_n}\{\lambda_I(g)\}$, где минимум берется по всем наборам $x(I)$ порядка $n - m$, вполне линеаризующим функцию g, эта величина обозначается $\lambda(g)$.

Заметим, что если H есть моноид всех преобразований множества V^m, то $æ_I(g,H) = æ_I(g)$, $æ(g,H) = æ(g)$. Если H есть полугруппа всех линейных преобразований множества V^m, то $æ_I(g,H) = \lambda_I(g)$, $æ(g,H) = \lambda(g)$.

Далее без потери общности считаем, что фиксируются первые $n - m$ переменных, то есть $x_1,\ldots,x_{n-m}$.

Для любой функции $g \in \Phi_{n,m}$, которая разветвляется только на преобразования, имеющие признак H, справедлива универсальная оценка степени (I,H)-разветвления:

$$1 \leqslant æ_I(g,H) \leqslant \min(2^{n-m}, |H|).$$

Отметим, что нижние и верхние границы достижимы:

1) $æ_I(g,H) = 1$, если функция g при любом опробовании сводится к единственному преобразованию из множества H;
2) $æ_I(g,H) = 2^{n-m}$, если $2^{n-m} \leqslant |H|$ и различны все преобразования из множества H, на которые разветвляется функция g;
3) $æ_H(g) = |H|$, если $2^{n-m} \geqslant |H|$ и каждое преобразование из множества H соответствует хотя бы одному опробованию набора переменных $x(I)$.

В частности, справедливы тривиальные оценки степени I-разветвления и степени линейного I-разветвления:

$$1 \leqslant æ_I(g) \leqslant \min(2^{n-m}, 2^{m\cdot 2^m}), \tag{1}$$

$$1 \leqslant \lambda_I(g) \leqslant \min(2^{n-m}, 2^{m^2}).$$

Для криптографических приложений представляет интерес описание класса функций из $\Phi_{n,m}$, для которых оценка степени (I,H)-разветвления существенно ниже, чем в (1).

Введенное понятие степени разветвления функций может быть использовано для анализа некоторых генераторов гаммы с неравномерным движением. Неравномерное движение информации в регистрах генератора используется для введения нелинейной зависимости знаков выходной гаммы от ключевых элементов генератора, в качестве которого фигурирует начальное заполнение регистров. В этом случае степень линейного разветвления характеризует сложность определения начального заполнения генератора с помощью линеаризации уравнений.

Рассмотрим генератор «δ–τ шагов» [1, с. 316], представляющий собой двухблочную схему, где управляющий автомат есть фильтрующий генератор на базе ЛРС-1 (линейного регистра сдвига) длины n с фильтрующей булевой функцией $f(x)$, а генерирующий автомат есть ЛРС-2 длины m (с одной из ячеек которого снимается гамма), продвижение информации в котором зависит от управляющего знака, поступающего от ЛРС-1.

Обозначим $x_i = (x_{i,n}, \ldots, x_{i,1})$, $y_i = (y_{i,m}, \ldots, y_{i,1})$ состояния ЛРС-1 и ЛРС-2 соответственно в i-м такте, $i = 1, 2, \ldots$; $x_1 = x$, $y_1 = y$ — их начальные состояния. Уравнения гаммообразования генератора имеют вид

$$\gamma_i = y_{\sigma(i,x),1},$$

где $\sigma(i,x)$ — суммарная «глубина продвижения» ЛРС-2 после первых i тактов функционирования:

$$\sigma(i,x) = \sum_{j=1}^{i} (\delta(f(x_j) \oplus 1) + \tau \cdot f(x_j)). \tag{2}$$

Ключом генератора являются начальные состояния ЛРС-1, ЛРС-2, а также функция усложнения f. Заметим, что состояние ЛРС-2 после l тактов функционирования зависит от поступившей от ЛРС-1 управляющей последовательности длины l и начального состояния ЛРС-2.

Зададим отображение ψ следующим образом: $\psi \in \Phi_{m+l,m}$, а именно ψ отображает множество управляющих последовательностей длины l и множество начальных состояний ЛРС-2 во множество состояний ЛРС-2 после l тактов функционирования генератора. Зафиксируем множество управляющих последовательностей и рассмотрим преобразования, на которые разветвляется отображение ψ. Таким образом, $I = \{1, \ldots, l\}$, далее индекс I опускаем.

Теорема 2. Справедлива оценка степени линейного разветвления отображения ψ:

$$\lambda(\psi) \leqslant l + 1. \tag{3}$$

Доказательство. Пусть g — линейная подстановка, реализуемая за один такт генерирующим ЛРС-2. При фиксации начального состояния x ЛРС-1 соответствующая управляющая последовательность длины l определяется однозначно и функция ψ совпадает с линейным преобразованием $g^{\sigma(l,x)}$ множества V^m. Следовательно, степень линейного разветвления функции ψ не превосходит числа различных значений, которые принимает величина $\sigma(l, x)$. В соответствии с (2) значение $\sigma(l, x)$ зависит только от веса управляющей последовательности длины l. Таким образом, $\lambda(\psi) \leqslant l + 1$. ■

Данный результат в общем случае допускает уточнение, так как в теореме не учитываются структурные свойства управляющей последовательности, а именно то, что она может быть, например, линейной рекуррентной последовательностью. С целью уточнения полученных результатов реализован вычислительный эксперимент. На языке C++ написана программа, реализующая фильтрующий генератор (управляющий блок генератора «δ–τ шагов») на базе ЛРС-1 длины 17 с функцией обратной связи, соответствующей примитивному многочлену $1 \oplus x^3 \oplus x^{17}$, что обеспечивает максимальную длину периода ЛРС-1. Фильтрующая функция генератора $f(x) = x_1 x_2 \oplus x_{17}$ сбалансирована в силу линейности по x_{17}. В результате выполнения программы для ЛРС-1 получено, что функция «число единиц в выходной последовательности длины 2^{10}» при различных ненулевых начальных состояниях принимает всего 157 значений. Заметим, что данное значение существенно ниже верхней границы неравенства (3).

Усложнением генератора «δ–τ шагов» является генератор с перемежающимся шагом [1, с. 318]. Его гамма образуется как сумма гамм двух генераторов «стоп — вперед» с генерирующими ЛРС-2 и ЛРС-3 длин m и r соответственно. Оба генерирующих регистра имеют общий управляющий блок — фильтрующий генератор с фильтрующей булевой функцией $f(x)$ на базе ЛРС-1 длины n.

Обозначим $x_i = (x_{i,n}, \ldots, x_{i,1})$, $y_i = (y_{i,m}, \ldots, y_{i,1})$, $z_i = (z_{i,r}, \ldots, z_{i,1})$ — состояния ЛРС-1, ЛРС-2 и ЛРС-3 соответственно в i-м такте, $i = 1, 2, \ldots$; $x_1 = x$, $y_1 = y$, $z_1 = z$ — начальные состояния регистров. Уравнения гаммообразования генератора имеют вид

$$\gamma_i = y_{\sigma(i,x),1} \oplus z_{i-\sigma(i,x),1},$$

где $\sigma(i, x)$ — суммарная «глубина продвижения» ЛРС-2 после первых i тактов функционирования:

$$\sigma(i, x) = \sum_{j=1}^{i} f(x_j).$$

Ключом генератора являются начальные состояния ЛРС-1, ЛРС-2, ЛРС-3, а также функция усложнения f. Состояния ЛРС-2, ЛРС-3 после l тактов функционирования зависят от поступившей от ЛРС-1 управляющей последовательности длины l и начальных состояний ЛРС-2, ЛРС-3.

Зададим отображение $\pi \in \Phi_{m+r+l,m+r}$ следующим образом: π отображает множество начальных состояний ЛРС-2, ЛРС-3 и множество управляющих последовательностей длины l во множество состояний ЛРС-2, ЛРС-3 после l тактов функционирования генератора. Зафиксируем множество управляющих последовательностей и рассмотрим преобразования, на которые разветвляется отображение π. Аналогично теореме 2 доказывается следующая теорема.

Теорема 3. Справедлива оценка степени линейного разветвления отображения π:

$$\lambda(\pi) \leqslant l + 1.$$

Полученные результаты могут быть использованы для криптографического анализа криптосистем, построенных на основе некоторых генераторов гаммы с неравномерным движением. Перспективным направлением представляется исследование разветвлений криптографических функций на преобразования, имеющие признак H не только линейный.

ЛИТЕРАТУРА

1. *Фомичёв В. М.* Методы дискретной математики в криптологии. М.: ДИАЛОГ-МИФИ, 2010. 424 с.
2. *Логачёв О. А., Сальников А. А., Ященко В. В.* Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2004. 470 с.

2012 Теоретические основы прикладной дискретной математики №1(15)

DOI 10.17223/20710410/15/4

УДК 511+519.1

О СВОЙСТВАХ КОЭФФИЦИЕНТОВ СУПЕРПОЗИЦИИ НЕКОТОРЫХ ПРОИЗВОДЯЩИХ ФУНКЦИЙ

Д. В. Кручинин

Томский государственный университет систем управления и радиоэлектроники, г. Томск, Россия

E-mail: KruchininDm@gmail.com

Получены свойства коэффициентов суперпозиции производящих функций $\ln\left((1-F(x))^{-1}\right)$, где $F(x)$ — обыкновенная производящая функция с целыми коэффициентами, позволяющие построить вероятностные проверки натурального числа на простоту. Приведена связь этих свойств с существующими проверками на простоту. Получены новые свойства чисел Люка и биномиальных коэффициентов вида $\binom{2n-1}{n-1}$.

Ключевые слова: *производящие функции, суперпозиция производящих функций, композиция натурального числа, тесты на простоту.*

Производящие функции являются мощным инструментом решения задач теории чисел, комбинаторики, алгебры, теории вероятностей. Одним из направлений в теории производящих функций является исследование коэффициентов их суперпозиции. В [1] показаны возможные пути вычисления таких коэффициентов. Данная работа продолжает исследования коэффициентов суперпозиции, при условии, что внешней функцией является производящая функция $\ln(1/(1-x))$, а внутренней — произвольная обыкновенная производящая функция с целыми коэффициентами.

Рассмотрим функции $f(n)$ и $r(n)$ и их производящие функции $F(x)=\sum\limits_{n\geqslant 1} f(n)x^n$ и $R(x)=\sum\limits_{n\geqslant 0} r(n)x^n$ соответственно. Тогда для вычисления функции $g(n)$ коэффициентов суперпозиции производящих функций $G(x)=R\left(F(x)\right)$ справедлива формула

$$g(0)=r(0),$$

$$g(n)=\sum_{k=1}^{n}\sum_{\substack{\lambda_i>0,\\ \lambda_1+\lambda_2+\ldots+\lambda_k=n}} f(\lambda_1)f(\lambda_2)\ldots f(\lambda_k)r(k). \qquad (1)$$

Следует отметить, что суммирование ведется по всем композициям натурального числа n, имеющим ровно k частей.

Определение 1. Композитой производящей функции $F(x)=\sum\limits_{n\geqslant 1} f(n)x^n$ называется функция

$$F^{\Delta}(n,k)=\sum_{\substack{\lambda_i>0,\\ \lambda_1+\lambda_2+\ldots+\lambda_k=n}} f(\lambda_1)f(\lambda_2)\ldots f(\lambda_k). \qquad (2)$$

Тогда из формулы (1) следует, что для вычисления суперпозиции $G(x)=R\left(F(x)\right)$ справедлива формула [1]

$$g(n)=\sum_{k=1}^{n} F^{\Delta}(n,k)r(k). \qquad (3)$$

Очевидно, что если $f(n)$, $r(n)$ — целочисленные функции, то и $g(n)$ является целочисленной функцией.

Рассмотрим суперпозицию производящих функций $G(x) = \ln((1 - F(x))^{-1})$ и докажем следующую теорему.

Теорема 1. Пусть имеется обыкновенная производящая функция с целыми коэффициентами $F(x) = \sum\limits_{n\geqslant 1} f(n)x^n$ и производящая функция $G(x) = \sum\limits_{n\geqslant 1} g(n)x^n$, где $G(x) = \ln((1 - F(x))^{-1})$. Тогда для любых значений $n > 0$ значения выражения $ng(n)$ являются целыми числами.

Доказательство. Согласно (3), для суперпозиции производящих функций $G(x) = \ln((1 - F(x))^{-1})$ функция коэффициентов имеет следующее выражение:

$$g(n) = \sum_{k=1}^{n} F^{\Delta}(n, k)/k.$$

Рассмотрим производную $G'(x) = [\ln((1 - F(x))^{-1})]'$ и получим выражение

$$\left(\frac{F'(x)}{1 - F(x)}\right) = g_1 + 2g_2x^1 + \ldots + ng_nx^{n-1} + \ldots$$

Рассмотрим левую часть как произведение производящих функций $F'(x)$ и $H(x) = (1 - F(x))^{-1}$. Коэффициенты для $F'(x)$ являются целыми числами. Коэффициенты суперпозиции $H(x)$ также являются целыми числами, поскольку $F(x)$ и $R(x) = (1 - x)^{-1}$ — производящие функции с целыми коэффициентами.

Произведение функций с целыми коэффициентами также имеет целые коэффициенты. Следовательно, значения выражения

$$ng(n) = n\sum_{k=1}^{n} F^{\Delta}(n, k)/k \tag{4}$$

являются целыми числами. ■

Следствие 1. Для любой целочисленной последовательности $a_1, a_2, \ldots, a_n$ сумма

$$\sum_{k=1}^{n} \frac{n}{k} \sum_{\substack{\lambda_i>0,\\ \lambda_1+\lambda_2+\ldots+\lambda_k=n}} a_{\lambda_1}a_{\lambda_2}\ldots a_{\lambda_k}$$

является целым числом.

Доказательство. Построим производящую функцию $F(x) = a_1x + a_2x^2 + \ldots + a_nx^n$. Тогда композитой этой функции по определению [1] будет

$$F^{\Delta}(n, k) = \sum_{\substack{\lambda_i>0,\\ \lambda_1+\lambda_2+\ldots+\lambda_k=n}} a_{\lambda_1}a_{\lambda_2}\ldots a_{\lambda_k}.$$

А значение выражения $\sum\limits_{k=1}^{n} \frac{n}{k} F^{\Delta}(n, k)$ является целым числом. ■

Следствие 2. Пусть имеется обыкновенная производящая функция с целыми коэффициентами $F(x) = \sum\limits_{n\geqslant 1} f(n)x^n$. Тогда для любого простого n значения выражения

$$\frac{ng(n) - f(1)^n}{n} \tag{5}$$

являются целыми числами, где $g(n)$ — функция коэффициентов суперпозиции производящих функций $\ln((1 - F(x))^{-1})$.

Доказательство. Рассмотрим выражение (4). Значение n-го слагаемого в сумме (4) равно целому значению $f(1)^n$ [1], т. е. значение выражения

$$ng(n) - f(1)^n = n\sum_{k=1}^{n-1} F^{\Delta}(n,k)/k$$

является целым числом.

Поскольку n простое, то оно не имеет общих делителей с $k < n$. Следовательно, значение выражения (5) является числом целым. ■

Таким образом, значение выражения

$$\sum_{k=1}^{n-1} F^{\Delta}(n,k)/k, \tag{6}$$

тождественного выражению (5), является целым числом для всех простых n. Для составных n значение (6) может быть как целым, так и не целым. Но если оно не целое, то n однозначно составное.

Полученный результат позволяет строить алгоритмы вероятностных проверок натурального числа на простоту на основе суперпозиции производящих функций $\ln\left((1-F(x))^{-1}\right)$, где $F(x)$ — обыкновенная производящая функция с целыми коэффициентами. Вероятностные проверки позволяют точно установить, что число является составным (если оно не удовлетворяет некоторым условиям). Если же число удовлетворяет всем условиям проверки, то это еще не означает, что оно простое; в этом случае говорят, что «n — простое число с некоторой вероятностью».

Рассмотрим ряд примеров.

Пример 1. Пусть задана производящая функция $F(x) = \beta x/(1-\alpha x)$, где α и β — любые целые числа, не равные 0, и ее композита, согласно [1]:

$$F^{\Delta}(n,k,\alpha) = \binom{n-1}{k-1}\alpha^{n-k}\beta^k.$$

Тогда коэффициенты суперпозиции

$$G(x) = \ln\left((1-\beta x/(1-\alpha x))^{-1}\right) = \ln\left((1-\alpha x)/(1-(\alpha+\beta)x)\right)$$

имеют следующий вид:

$$g(n) = \sum_{k=1}^{n}\binom{n-1}{k-1}\frac{\alpha^{n-k}\beta^k}{k}.$$

Отсюда

$$ng(n) = \sum_{k=1}^{n}\frac{n}{k}\binom{n-1}{k-1}\alpha^{n-k}\beta^k = (\alpha+\beta)^n - \alpha^n.$$

Таким образом, значение выражения

$$\sum_{k=1}^{n-1}\frac{\alpha^{n-k}\beta^k}{k}\binom{n-1}{k-1} = \frac{(\alpha+\beta)^n - \alpha^n - \beta^n}{n}$$

для простых n является целым числом.

Если рассматривать β как некоторую переменную X, а $\alpha = -a$, где a — любое целое число, то с учётом равенства $a^n \equiv a \pmod{n}$ получим следующее тождество:

$$(X - a)^n \equiv (X^n - a) \pmod{n}.$$

Это тождество является основой для детерминированного полиномиального алгоритма проверки чисел на простоту AKS [3].

Пример 2. Пусть задана производящая функция $F(x) = x + x^2$ и её композита, согласно [1], $F^{\Delta}(n,k) = \binom{k}{n-k}$. Тогда коэффициенты суперпозиции $\ln\left((1 - x - x^2)^{-1}\right)$ имеют вид

$$g(n) = \sum_{k=1}^{n} \binom{k}{n-k} \frac{1}{k}.$$

Начальные элементы последовательности $ng(n)$, $n = 1, 2, \ldots$, равны $1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843, 1364, 2207, 3571, \ldots$ и совпадают с соответствующими элементами последовательности чисел Люка (A000032 в [2]), что дает основание предполагать, что эти две последовательности совпадают. В этом случае значение выражения $(L_n - 1)/n = (ng(n) - 1)/n$ является целым для простых n, где L_n — числа Люка:

$$L_n = \left(\frac{1 + \sqrt{5}}{2}\right)^n + \left(\frac{1 - \sqrt{5}}{2}\right)^n.$$

Пример 3. Пусть задана производящая функция для чисел Каталана $F(x) = \dfrac{1 - \sqrt{1 - 4x}}{2x}$ и ее композита, согласно [1], $F^{\Delta}(n,k) = \dfrac{k}{n}\binom{2n-k-1}{n-1}$. Тогда коэффициенты суперпозиции $\ln\left((1 - F(x))^{-1}\right)$ имеют следующее выражение:

$$g(n) = \sum_{k=1}^{n} \frac{k}{n}\binom{2n-k-1}{n-1}\frac{1}{k} = \frac{1}{n}\sum_{k=1}^{n}\binom{2n-k-1}{n-1}.$$

Эта формула порождает последовательность целых чисел $ng(n)$, $n = 1, 2, \ldots$,

$$1, 3, 10, 35, 126, 462, 1716, 6435, 24310, 92378, \ldots,$$

которая совпадает с A001700 [2]; поэтому

$$ng(n) = \binom{2n-1}{n-1}.$$

Таким образом, по следствию 2 значение выражения

$$\frac{1}{n}\left(\binom{2n-1}{n-1} - 1\right)$$

является целым для простых n.

Таким образом, получены свойства коэффициентов суперпозиции производящих функций $\ln\left((1 - F(x))^{-1}\right)$, где $F(x)$ — обыкновенная производящая функция с целыми коэффициентами. Эти свойства могут служить основой для построения алгоритмов вероятностных проверок натурального числа на простоту. Получены также новые свойства чисел Люка и биномиальных коэффициентов вида $\binom{2n-1}{n-1}$.

ЛИТЕРАТУРА

1. *Кручинин В. В.* Комбинаторика композиций и её приложения. Томск: В-Спектр, 2010. 156 с.
2. `www.oeis.org` — J. A. Sloane. The On-Line Encyclopedia of Integer Sequences. 2011.
3. *Agrawal M., Kayal N., and Saxena N.* PRIMES is in P // Ann. Math. 2004. V. 160. No. 2. P. 781–793.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ СТЕГАНОГРАФИИ

DOI 10.17223/20710410/15/5

УДК 681.511:3

МНОЖЕСТВА ИДЕНТИФИКАЦИОННЫХ НОМЕРОВ, УСТОЙЧИВЫЕ К АТАКЕ СГОВОРОМ

Т. М. Соловьёв, Р. И. Черняк

Национальный исследовательский Томский государственный университет, г. Томск, Россия

E-mail: tm.solovev@gmail.com, r.chernyack@gmail.com

Исследуются множества идентификационных номеров, обеспечивающие устойчивость стегосистемы к атаке сговором. Изучается структура таких множеств, различные способы представления, предлагается алгоритм их построения по ключу. Рассматривается задача идентификации участников атаки сговором.

Ключевые слова: *цифровые водяные знаки, стегосистемы идентификационных номеров, коллективные атаки, атака сговором.*

Введение

В настоящее время для защиты медиаконтента от несанкционированного использования широко применяются цифровые водяные знаки (ЦВЗ). В общем случае ЦВЗ — это некоторые метки, внедряемые в исходный материал. Они могут содержать информацию о времени и месте производства контента, а также о пользователе, которому этот контент предназначался. Метки последнего типа выделяются в отдельный класс и называются идентификационными номерами (ИН).

В случае применения ИН в контейнер, предназначенный каждому пользователю, внедряется персональный номер, позволяющий контролировать дальнейший путь этого контейнера. Такие системы в работе [1] называются *стегосистемами идентификационных номеров*. Если пользователь окажется медиапиратом и начнет незаконное распространение контента, то ИН позволит быстро определить злоумышленника. При этом предполагается, что все контейнеры одинаковы, то есть выданные копии отличаются только в тех позициях, в которые внедрены идентификаторы. Одной из основных отличительных черт данных систем является их потенциальная уязвимость к *коллективным атакам*, или *атакам сговором*. При реализации таких атак используются особенности стегосистем ИН, поэтому обеспечение защиты с помощью классических для ЦВЗ методик невозможно.

В работах [2, 3] рассматривались проблемы создания множеств ИН, устойчивых к коллективным атакам. Настоящая работа обобщает и расширяет полученные ранее результаты. Подробно исследуется структура множеств ИН, приводятся их основные характеристики. Обсуждается проблема идентификации группы злоумышленников при помощи ИН, полученного в результате атаки сговором. В заключение приводится алгоритм построения множества ИН по ключу.

1. Основные определения

Рассмотрим стегосистему ИН с множеством идентификаторов W. Пусть среди пользователей системы существует группа злоумышленников, обладающих копиями контента с идентификаторами из множества P.

Определение 1. *Атакой сговором* называется следующая последовательность действий:

1) определение части позиций, в которые внедрен ИН, посредством побитного сравнения копий, имеющихся у злоумышленников;
2) создание новой копии путем изменения значений бит с номерами, определенными на шаге 1.

Идентификатор, содержащийся в копии, полученной в результате атаки сговором, далее будем называть *ложным*.

Возможны следующие угрозы:

а) создание копии с ИН, не идентифицирующим никого;
б) создание копии с ИН из множества $W \setminus P$, то есть с идентификатором одного из пользователей, непричастных к атаке.

Из определения 1 непосредственно следует, что атака сговором более эффективна в случае, если ИН в разных копиях внедрен в разные позиции. Поэтому с точки зрения защиты целесообразно внедрение идентификаторов в одинаковые позиции во всех копиях. Далее рассматриваются только такие стегосистемы.

Замечание 1. Ложный ИН, независимо от фиксации бит на шаге 2 атаки сговором, находится внутри наименьшего интервала, покрывающего множество идентификаторов использованных копий.

Последний факт является важной особенностью атаки сговором, и на его основе представляется возможным разработать эффективное противодействие данной атаке.

Определение 2. Множество идентификационных номеров $W_n \subseteq \{0,1\}^n$ (или W, если длина ИН не существенна) называется *допустимым*, если каждому его непустому подмножеству P взаимно-однозначно сопоставляется наименьший интервал $I(P)$, покрывающий это подмножество.

Далее, если не оговорено другое, будем рассматривать только собственные подмножества P множества W_n. Интервал $I(P)$ будем называть *соответствующим* множеству P и обозначать троичным вектором с компонентами из множества $\{0,1,-\}$. При этом внутренние компоненты интервала обозначаются символом «$-$», а внешние — соответствующими константами из $\{0,1\}$.

В рамках решаемой задачи достаточно ограничиться рассмотрением таких допустимых множеств W_n, для которых $I(W_n) = \underbrace{-\,-\,\ldots\,-}_{n}$. В противном случае все ИН и, как следствие, все покрывающие интервалы содержат одну и ту же константу в некоторой компоненте, что никак не помогает при идентификации сговорившихся, а лишь вносит избыточность во внедряемую метку.

2. Свойства допустимых множеств

Свойство 1. Множество W допустимо, если и только если для любого его подмножества P интервал $I(P)$ не содержит векторов из $W \setminus P$.

Доказательство. Необходимость. Если вектор w из $W \setminus P$ принадлежит интервалу $I(P)$, то $I(P) = I(P \cup \{w\})$, что противоречит допустимости множества W, поскольку $P \neq P \cup \{w\}$.

Достаточность[1]. Пусть W не является допустимым, т. е. существуют два его различных подмножества P_1 и P_2, такие, что $I(P_1) = I(P_2)$. Пусть для определенности $P_1 \setminus P_2$ не пусто. Возьмём вектор w из $P_1 \setminus P_2$. Он принадлежит P_1, следовательно, принадлежит и $I(P_1) = I(P_2)$. Таким образом, интервал $I(P_2)$ содержит вектор из $P_1 \setminus P_2 \subseteq W \setminus P_2$. ■

Свойство 1 означает, что если множество идентификаторов стегосистемы ИН является допустимым, то злоумышленники при проведении атаки сговором никогда не смогут «подставить» невиновного пользователя, и любое множество с таким свойством допустимо.

Лемма 1. Пусть W — допустимое множество, $|W| \geqslant 3$, $P \subseteq W$, $|P| \geqslant 2$. Тогда $\dim(I(P)) \geqslant |P|$, где $\dim(I)$ — размерность интервала I.

Доказательство. Индукция по мощности множества P.

Б а з а и н д у к ц и и. $|P| = 2$. Покажем, что для любых двух различных векторов w_p и w_q из W размерность покрывающего их интервала не может быть равна единице. Предположим противное: w_p отличается от w_q только в одной компоненте. Пусть она имеет номер i. Рассмотрим вектор $w_r \in W$, отличный от w_p и w_q (он существует, так как $|W| \geqslant 3$). Возможны два варианта: либо $w_r[i] = \overline{w_p[i]} = w_q[i]$, либо $w_r[i] = w_p[i] = \overline{w_q[i]}$. В первом случае получим $I(\{w_p, w_q, w_r\}) = I(\{w_p, w_r\})$, во втором — $I(\{w_p, w_q, w_r\}) = I(\{w_q, w_r\})$. Оба варианта противоречат определению допустимого множества.

Ш а г и н д у к ц и и. Пусть для любого P мощности n верно неравенство $\dim(I(P)) \geqslant n$. Рассмотрим произвольное подмножество $P' \subseteq W$ мощности $n+1$. Оно может быть представлено в виде $P' = P \cup \{w\}$, где $|P| = n$, $w \in W \setminus P$. Интервал $I(P \cup \{w\})$ либо равен интервалу $I(P)$, либо имеет бóльшую размерность. Первое невозможно в силу допустимости W, следовательно, $\dim(I(P')) \geqslant \dim(I(P)) + 1 \geqslant n + 1 = |P'|$. ■

Следствием леммы является следующее свойство.

Свойство 2. Мощность допустимого множества W_n не превышает n.

Данное свойство налагает суровое ограничение на количество пользователей системы. Например, для того чтобы обеспечить идентификационными номерами сеть из десяти миллионов пользователей, необходимо использовать ИН длиной не менее мегабайта. Такой объем дополнительного материала является существенным, и его дальнейшее увеличение может создавать проблемы на этапе внедрения метки.

Свойство 3. Множество W_n мощности n допустимо, если и только если оно является сферой радиуса 1 с некоторым вектором a в качестве центра, т. е. имеет вид $O_1(a) = \{x \in \{0,1\}^n : \mathrm{d}(a,x) = 1\}$, где $\mathrm{d}(a,x)$ — расстояние по Хэммингу между векторами a и x.

Доказательство. Необходимость для $n \leqslant 2$ проверяется непосредственно.

Пусть $n > 2$. Построим вектор a для произвольного допустимого множества W_n мощности n. Обозначим различные $(n-1)$-элементные подмножества W_n через P_1, P_2, …, P_n. Согласно лемме 1, $\dim(I(P_i)) \geqslant n-1$ для всех $i = 1, 2, \ldots, n$, а так как $\dim(I(W_n)) = n$ и существует только один интервал размерности n, то $\dim(I(P_i)) = n-1$, $i = 1, 2, \ldots, n$. Следовательно, для каждого i существует единственное j_i, такое, что j_i-я компонента во всех векторах из P_i одинакова. Запишем её значение

[1] Достаточность данного свойства сформулирована и доказана Р. С. Стружковым.

в j_i-ю компоненту вектора a; заметим, что $j_i \neq j_k$ при $i \neq k$ ввиду $\dim(I(W_n)) = n$. По построению, $\mathrm{d}(a, w) = 1$ для любого $w \in W_n$, т. е. $W_n = O_1(a)$.

Достаточность. Покажем, что для любых различных P_1 и P_2 — подмножеств $O_1(a)$ — верно неравенство $I(P_1) \neq I(P_2)$. Очевидно, что если $|P_1| = 1$ или $|P_2| = 1$, то $I(P_1) \neq I(P_2)$. Пусть $|P_1| > 1$, $|P_2| > 1$ и $P_1 \setminus P_2 \neq \varnothing$. Возьмём вектор $w \in P_1 \setminus P_2$; пусть i — номер компоненты, в которой w отличается от a. Так как все векторы в $O_1(a)$, за исключением w, совпадают в i-й компоненте, то i-я компонента в $I(P_2)$ будет внешней, а в $I(P_1)$ — внутренней. Таким образом, $I(P_1) \neq I(P_2)$. ■

Пусть $W = \{w_i : i = 1, 2, \ldots, k\}$; обозначим $W^+(l) = \{w_i \cdot a_i : w_i \in W,\ a_i \in \{0,1\}^l,\ i = 1, 2, \ldots, k\}$, где символом «$\cdot$» обозначена операция конкатенации.

Свойство 4. Если W — допустимое множество, то все множества вида $W^+(l)$ также являются допустимыми для любых l.

Доказательство. От противного: пусть существует l, такое, что для некоторых двух различных $P_1^+(l), P_2^+(l) \subseteq W^+(l)$ верно равенство $I\left(P_1^+(l)\right) = I\left(P_2^+(l)\right)$. Тогда $I(P_1) = I(P_2)$ для соответствующих $P_1, P_2 \subseteq W$, что противоречит допустимости множества W. ■

3. Матричное представление множеств ИН

Рассмотрим множество ИН W_n мощности k. Представим его в виде булевой матрицы $W_{k\times n}$, строками которой будут векторы из множества W_n:

$$W_{k\times n} = \begin{pmatrix} w_1[1] & w_1[2] & \ldots & w_1[n] \\ w_2[1] & w_2[2] & \ldots & w_2[n] \\ \vdots & \vdots & \ddots & \vdots \\ w_k[1] & w_k[2] & \ldots & w_k[n] \end{pmatrix}.$$

Матрицы, соответствующие допустимым множествам, будем также называть допустимыми.

Определим следующие операции над матрицами:

1) перестановка столбцов или строк;
2) инверсия всех элементов заданного столбца;
3) удаление одного из повторяющихся столбцов или добавление столбца, равного одному из уже присутствующих в матрице.

Определение 3. Матрицы M и M' назовем *эквивалентными*, если они могут быть получены друг из друга путем применения операций 1–3. Множества ИН W и W' назовем *эквивалентными*, если соответствующие им матрицы эквивалентны.

Очевидно, что применение к матрице операций 1–3 не влияет на ее допустимость, и матрица, эквивалентная допустимой, также является допустимой.

Утверждение 1. Матрица $W_{k\times n}$ является допустимой, если и только если она эквивалентна блочной матрице вида $(E_k M)$, где E_k — единичная матрица размера $k\times k$; M — произвольная k-строчная матрица.

Доказательство. Необходимость. Пусть допустимой матрице $W_{k\times n}$ соответствует множество $W_n = \{w_1, w_2, \ldots, w_k\}$. Определим два индексных множества $K = \{1, 2, \ldots, n\}$ и $J = \varnothing$.

Рассмотрим интервал $I_1 = I(W_n \setminus \{w_1\})$. Пусть I_1 имеет l внешних компонент с номерами i_1, i_2, …, i_l. Тогда вектор w_1 в этих компонентах отличается от остальных

векторов множества W_n, т. е. матрица $W_{k\times n}$ имеет вид

$$\begin{pmatrix} c_{11} & c_{12} & \dots & \boxed{\overline{c_{i_1}}} & \dots & \boxed{\overline{c_{i_2}}} & \dots & \boxed{\overline{c_{i_l}}} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{i_1} & \dots & c_{i_2} & \dots & c_{i_l} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{k1} & c_{k2} & \dots & c_{i_1} & \dots & c_{i_2} & \dots & c_{i_l} & \dots & c_{kn} \end{pmatrix}.$$

Поскольку матрица булева, выделенные столбцы равны с точностью до инверсии, следовательно, удаление всех этих столбцов, за исключением одного, есть эквивалентное преобразование. Исключим из множества K элементы $i_2, i_3, \dots, i_l$, а элемент i_1 добавим в множество J.

Проведем аналогичные действия для векторов $w_2, w_3, \dots, w_k$; заметим, что множества номеров внешних компонент интервалов $I(W_n \setminus \{w_i\})$, $i = 1, 2, \dots, k$, попарно не пересекаются.

В результате будут построены множества J и K, такие, что

$$J \subseteq K, \quad |J| = k. \tag{1}$$

Составим матрицу $\widetilde{W}_{k\times n'}$ из тех столбцов матрицы $W_{k\times n}$, номера которых входят в множество K. По построению, $\widetilde{W}_{k\times n'}$ эквивалентна $W_{k\times n}$, следовательно, является допустимой. Выделим в $\widetilde{W}_{k\times n'}$ подмножество столбцов, номера которых образуют множество J. Эти столбцы имеют следующий вид:

$$\left(\underbrace{\overbrace{c_{i_j}, c_{i_j}, \dots, c_{i_j}}^{j-1}, \overline{c}_{i_j}, c_{i_j}, \dots, c_{i_j}}_{k}\right)^{\mathrm{T}}, \quad j = 1, 2, \dots, k.$$

Эквивалентными преобразованиями матрица, составленная из таких столбцов, может быть приведена к E_k. Остальные столбцы образуют матрицу M. Если $n' = k$, матрица M пуста.

Достаточность. Матрица E_k допустима по свойству 3, следовательно, по свойству 4 матрица $(E_k M)$ (и любая эквивалентная ей) также является допустимой. ■

Замечание 2. Для любой допустимой матрицы размера $k \times n$ существует эквивалентная ей блочная матрица вида $(E_k M_{k\times m})$, где $0 \leqslant m \leqslant n - k$.

Следствие 1. Для любой допустимой матрицы размера $k \times n$ существует эквивалентная блочная матрица вида $(E_k M)$, в которой подматрица M либо пуста, либо не является допустимой.

Доказательство. По утверждению 1, любая допустимая матрица эквивалентными преобразованиями может быть приведена к виду $(E_k M_{k\times m})$. Рассмотрим подматрицу $M_{k\times m}$.

Если $m < k$, то матрица $M_{k\times m}$ либо пуста, либо не допустима (по свойству 2).

Пусть $m \geqslant k$ и матрица $M_{k\times m}$ допустима. Тогда, с учетом замечания 2, эквивалентными преобразованиями она может быть приведена к виду $(E_k M_{k\times t})$ для $0 \leqslant t \leqslant m-k$, т. е. исходная матрица примет вид $(E_k E_k M_{k\times t})$, и одно из вхождений E_k может быть удалено. Получим $(E_k M_{k\times t})$.

Те же рассуждения применим к матрице $M_{k\times t}$ и так далее, пока за конечное число шагов не будет получена эквивалентная исходной матрица вида $(E_k M_{k\times s})$, где $0 \leqslant s \leqslant m - lk$, $0 \leqslant l \leqslant [m/k]$, и подматрица $M_{k\times s}$ либо пуста, либо не допустима. ■

Утверждение 1 позволяет ввести понятие базы допустимого множества.

Определение 4. *Базой* (*базовой частью*) k-строчной допустимой матрицы называется набор таких её k столбцов, из которых может быть составлена матрица, эквивалентная E_k. Совокупность остальных столбцов матрицы называется *дополнением* (*дополнительной частью*).

Определение 5. Допустимая матрица называется *сильно допустимой*, если удаление из неё любого столбца влечет потерю свойства допустимости. Множество, соответствующее такой матрице, будем также называть *сильно допустимым*.

Утверждение 2. Допустимая матрица является сильно допустимой, если и только если она состоит только из базовой части.

Доказательство. Необходимость. Пусть для некоторой k-строчной сильно допустимой матрицы W существует эквивалентная матрица $(E_k M)$, причем подматрица M не пуста. Удалим из W один из дополнительных столбцов. В результате получим матрицу W', для которой найдется эквивалентная матрица $(E_k M')$. По утверждению 1, W' допустима, что противоречит предположению сильной допустимости W.

Достаточность очевидна. ■

Таким образом, свойство допустимости равносильно наличию подматрицы, эквивалентной единичной матрице заданного размера. Оставшаяся часть матрицы не влияет на допустимость и может быть выбрана произвольно.

4. Построение допустимых множеств

Построение допустимого множества, которому соответствует единичная матрица, не вызывает никаких сложностей, однако на практике этого недостаточно. Необходим алгоритм, с помощью которого можно построить любое допустимое множество. Поскольку в реальных стегосистемах количество идентификаторов и их размеры велики, важным фактором является производительность алгоритма. Для обеспечения возможности распараллеливания алгоритм должен обладать следующим свойством: каждый вектор строится независимо от остальных.

Алгоритм 1 Построение допустимого множества

Вход: $k, n, v, \pi, M_{k\times(n-k)}$
Выход: допустимое множество W из k векторов длины n

1: $i := 1$, w — нулевой вектор длины n, $W := \varnothing$
2: **Пока** $i \leqslant k$
3: $\quad w[i] := 1$
4: $\quad$ **Для** $j := k+1$ до n
5: $\quad\quad w[j] := M[i, j]$
6: $\quad w := w \oplus v$
7: $\quad$ Переставить компоненты вектора w в соответствии с подстановкой π
8: $\quad$ Добавить w в W
9: $\quad i := i + 1$, обнулить w
10: **Вернуть** W

Результатом работы алгоритма является множество из k идентификационных номеров длины n ($n \geqslant k$). Номера столбцов, которые в соответствующей матрице необходимо инвертировать, задаются булевым вектором v длины n. Для изменения порядка следования столбцов используется перестановка $\pi \in S_n$. Пара (v, π) может быть ис-

пользована в качестве ключа. На вход алгоритма подаётся также матрица M размера $k \times (n-k)$, которая вносит в идентификационные номера некоторую избыточность.

Согласно утверждению 1, матрица, соответствующая любому допустимому множеству, может быть построена из матрицы вида $(E_k M)$ при помощи операций перестановки, инвертирования и добавления/удаления повторяющихся столбцов. Таким образом, с помощью алгоритма 1 может быть построено любое допустимое множество.

5. Идентификация злоумышленников

Обсудим проблему идентификации участников сговора по полученному ими ложному ИН. В соответствии со свойством 1, в случае использования допустимого множества ложный ИН никогда не совпадет ни с одним из существующих в системе идентификаторов. Таким образом, для множества ИН W_n задача идентификации может быть сформулирована следующим образом: «Используя ложный ИН w' из $\{0,1\}^n \setminus W_n$, восстановить некоторое непустое подмножество пользователей, участвовавших в атаке сговором».

Пусть, для наглядности, допустимое множество задано матрицей E_k. В этом случае в идентификаторе каждого пользователя есть только одна единица. Она находится в первой компоненте ИН первого пользователя, во второй компоненте ИН второго и так далее. Таким образом, если у злоумышленников нет i-го ИН, то в i-й компоненте построенного ими ложного идентификатора всегда будет ноль. Другими словами, если в i-й компоненте ложного ИН присутствует единица, это однозначно говорит о том, что i-й пользователь участвовал в сговоре. На основе этих рассуждений представляется возможным идентифицировать некоторое непустое подмножество участников сговора по построенному ими ложному ИН во всех случаях, кроме одного — когда последний целиком состоит из нулей.

Рассмотрим пример атаки сговором с участием трёх злоумышленников на множество ИН, заданное матрицей E_5; пусть злоумышленниками являются первые три пользователя:

$$\begin{pmatrix} \boxed{\begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{matrix}} \\ \begin{matrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{matrix} \end{pmatrix}.$$

Ложные идентификаторы в этом случае выбираются из интервала $---00$. Ниже показан результат идентификации злоумышленников в зависимости от построенного ими ложного ИН:

Ложный ИН	Результат идентификации
11000	$\{1,2\}$
10100	$\{1,3\}$
01100	$\{2,3\}$
11100	$\{1,2,3\}$
00000	$\varnothing$

Заметим, что во всех случаях, кроме последнего, найдено некоторое подмножество злоумышленников, при этом ни один не причастный к атаке пользователь не был обвинён. Очевидно, что подобные рассуждения возможны и в том случае, если некоторые столбцы в матрице E_k будут инвертированы. В рассмотренном примере, инвертируя

первый и второй столбцы, получим матрицу

$$\left(\begin{array}{|ccccc|} \hline 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{array} \right).$$

В этом случае результат идентификации будет следующим:

Ложный ИН	Результат идентификации
00000	$\{1, 2\}$
01100	$\{1, 3\}$
10100	$\{2, 3\}$
00100	$\{1, 2, 3\}$
11000	$\varnothing$

В общем случае вывод о том, что i-й пользователь участвовал в атаке сговором, делается, если в i-й компоненте ложного ИН наблюдается значение $\overline{f_{\text{maj}}}(w_1[i], w_2[i], \ldots, w_k[i])$, где f_{maj} — мажоритарная функция:

$$f_{\text{maj}}(x) = \begin{cases} 1, & \text{если } \sum\limits_{i=1}^{n} x_i > n/2 \ , \\ 0 & \text{иначе.} \end{cases}$$

Используя метод из доказательства необходимости в утверждении 1, в матрице, задающей допустимое множество, всегда можно выделить базовую часть. Таким образом, описанный способ идентификации злоумышленников может быть использован в любом допустимом множестве.

6. Мажорирующая атака

В предложенном методе идентификации используются лишь те компоненты ложного ИН, которые соответствуют базовым столбцам, поэтому далее ограничимся рассмотрением только сильно допустимых множеств.

В приведенных примерах идентификации присутствует ложный ИН, который не идентифицирует никого:

$$w[i] = f_{\text{maj}}(w_1[i], w_2[i], \ldots, w_k[i]), \quad i = 1, 2, \ldots, n.$$

Обозначим его w_{maj}. Возникает вопрос, всегда ли злоумышленники могут построить w_{maj} и существует ли стратегия, позволяющая им получать именно этот ИН, а не какой-то случайный вектор из интервала, соответствующего использованным идентификаторам?

Утверждение 3. Если множество W сильно допустимо, $P \subseteq W$ и $|P| \geqslant 2$, то $w_{\text{maj}} \in I(P)$.

Доказательство. Заметим, что в сильно допустимом множестве для любых двух различных его элементов w_s и w_t и для любого $i = 1, 2, \ldots, n$ всегда выполняется следующее: либо $w_s[i] \neq w_t[i]$, либо $w_s[i] = w_t[i] = w_{\text{maj}}[i]$. Другими словами, каждая компонента интервала $I(\{w_s, w_t\})$ либо внутренняя, либо совпадает с соответствующей компонентой w_{maj}, следовательно, $w_{\text{maj}} \in I(\{w_s, w_t\})$. Таким образом, для любого $P \subseteq W$ мощности не меньшей двух $w_{\text{maj}} \in I(P)$. ■

Замечание 3. Идентификационный номер w_{maj} может быть построен следующим образом:

$$w_{\text{maj}}[i] = f_{\text{maj}}(w_p[i], w_q[i], w_r[i]), \quad i = 1, 2, \ldots, n,$$

где w_p, w_q, w_r — любые попарно различные ИН из сильно допустимого множества W.

Таким образом, при наличии трех или более ИН возможно построение метки, не идентифицирующей никого.

Описанный способ построения ложного ИН назовём *мажорирующей атакой*. В соответствии с предложенным методом идентификации проведение данной атаки является единственным способом для злоумышленников избежать ответственности. Поскольку мажорирующая атака состоит в выборе конкретного вектора из $I(P)$, возможна разработка метода эффективного противодействия данной атаке.

Заключение

В работе предлагается модель множеств идентификаторов для стегосистем ИН, позволяющая противостоять атаке сговором. Последнее достигается за счет уникальности наименьших покрывающих интервалов для каждого непустого подмножества идентификаторов. Выделен ряд свойств таких множеств, исследована их структура и предложен алгоритм генерации. Задача идентификации злоумышленников по ИН, полученному в результате сговора, решена для всех коллективных атак, за исключением одного частного случая — мажорирующей атаки. Противостояние последней является одним из приоритетных направлений дальнейшего исследования. Данная задача может быть решена с использованием статистических методов, поскольку, в рамках предложенной модели, возможен произвольный выбор дополнительной части.

ЛИТЕРАТУРА

1. *Грибунин В. Г., Оков И. Н., Туринцев И. В.* Цифровая стеганография. М.: Солон-Пресс, 2002.
2. *Струйжков Р. С., Соловъёв Т. М., Черняк Р. И.* Цифровые водяные знаки, устойчивые к атаке сговором // Прикладная дискретная математика. Приложение. 2009. № 1. С. 56–59.
3. *Соловъёв Т. М., Черняк Р. И.* Стегосистемы идентификационных номеров, устойчивые к атаке сговором // Прикладная дискретная математика. Приложение. 2010. № 3. С. 41–44.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

DOI 10.17223/20710410/15/6

УДК 004.94

РОЛЕВАЯ ДП-МОДЕЛЬ УПРАВЛЕНИЯ ДОСТУПОМ И ИНФОРМАЦИОННЫМИ ПОТОКАМИ В ОПЕРАЦИОННЫХ СИСТЕМАХ СЕМЕЙСТВА *Linux*

П. Н. Девянин

Институт криптографии, связи и информатики, г. Москва, Россия

E-mail: peter_devyanin@hotmail.com

Базовая ролевая ДП-модель управления доступом и информационными потоками в операционных системах (БРОС ДП-модель) достраивается до ролевой ДП-модели управления доступом и информационными потоками в операционных системах (ОС) семейства *Linux* (РОСЛ ДП-модели). Описываются ее новые элементы: имена сущностей, мандатные атрибуты целостности сущностей-контейнеров, функция де-факто владения. Основное отличие модели — четкое разделение де-юре правил преобразования состояний (требующих реализации в ОС) и де-факто правил (используемых только для анализа условий нарушения безопасности системы). Рассматриваются ограничения, инвариантные относительно немонотонных правил преобразования состояний. С учетом внесенных изменений обосновывается утверждение о возможности использования только монотонных правил преобразования состояний для анализа условий передачи прав доступа ролей, получения доступов или реализации информационных потоков.

Ключевые слова: *компьютерная безопасность, ролевая ДП-модель, операционная система Linux.*

1. Введение. Этапы разработки ролевых ДП-моделей

Одним из направлений развития теории компьютерной безопасности является адаптация формальных моделей логического управления доступом и информационными потоками к условиям функционирования реальных компьютерных систем, в особенности ОС. Существующие ОС включают механизмы дискреционного или мандатного управления доступом, сложность реализации которых существенно превосходит возможности аппарата формальных моделей, ориентированных на анализ безопасности этих механизмов. Классические модели, например *Take-Grant* или Белла — ЛаПадулы [1], в основном устарели и не позволяют учитывать свойства современных ОС, существенно влияющие на их безопасность [2]. Таким образом, целесообразна постепенная детализация формальных моделей, уточнение описания их элементов, отработка в рамках моделей техники строгого теоретического обоснования свойств ОС.

Хотя ролевое управление доступом является достаточно сложным механизмом обеспечения безопасности компьютерных систем как с точки зрения научного исследования его свойств, так и с учетом особенностей его практической реализации, оно является одним из самых перспективных, в том числе для применения в ОС. В связи

с этим разработка семейства моделей ролевого управления доступом и информационными потоками (ролевых ДП-моделей) осуществлялась автором поэтапно.

На первом этапе на базе ролевых моделей семейства *RBAC*, дискреционных и мандатных ДП-моделей была построена базовая ролевая ДП-модель (БР ДП-модель) [1], в рамках которой отработана техника обоснования необходимых и достаточных условий передачи прав доступа ролей и реализации информационных потоков по памяти (для случая, когда на траекториях функционирования системы субъект-сессии не получают доступа владения друг к другу с использованием информационных потоков по памяти к функционально ассоциированным с субъект-сессиями сущностям).

На втором этапе с использованием ДП-модели с функционально или параметрически ассоциированными с субъектами сущностями (ФПАС ДП-модели) [3], моделей мандатной политики целостности информации Биба и мандатного ролевого управления доступом [1] была разработана базовая ролевая ДП-модель управления доступом и информационными потоками в ОС (БРОС ДП-модель) [4]. В нее по сравнению с БР ДП-моделью включены некоторые специфичные современным ОС элементы: учетные записи пользователей, сущности, параметрически ассоциированные с субъект-сессиями или ролями («знание» которых — реализация от которых информационных потоков по памяти — позволяет недоверенным субъект-сессиям получить контроль над доверенными субъект-сессиями), де-факто роли, права доступа, возможности и доступы недоверенных субъект-сессий (получаемые ими за счет контроля над доверенными субъект-сессиями), мандатный контроль целостности. Это позволило смоделировать применяемые в ряде современных ОС соответствующие механизмы защиты (например, механизмы *MIC — Mandatory Integrity Control* и *UAC — User Account Control* в ОС семейства *Microsoft Windows*). Основное внимание в БРОС ДП-модели было уделено уточнению условий и результатов применения правил преобразования состояний с учетом включенных в нее новых элементов. Кроме того, так же, как в предыдущих ДП-моделях, была обоснована возможность использования только монотонных правил преобразования состояний при анализе условий передачи прав доступа, реализации информационных потоков по памяти или по времени.

На третьем реализуемом в настоящее время автором этапе исследований формируется ролевая ДП-модель управления доступом и информационными потоками в ОС семейства *Linux* (РОСЛ ДП-модель), основные элементы которой описаны в [5]. Эта модель не сформирована окончательно, в нее (по сравнению с БРОС ДП-моделью) вносятся изменения, направленные в первую очередь на постепенную адаптацию модели к условиям функционирования реальных ОС, на создание предпосылок для разработки механизма управления доступом в защищенных ОС рассматриваемого семейства на основе формальной модели. Рассмотрим эти изменения подробнее.

2. Состояние системы

При описании состояния системы в рамках РОСЛ ДП-модели скорректированы предположения 1 и 5 БРОС ДП-модели (см. [4]), в которых вместо информационных потоков по памяти к сущностям рассматриваются доступы на чтение к ним субъект-сессий. Это связано с тем, что в реальных ОС возникновение информационных потоков является «де-факто следствием» «де-юре получения» субъект-сессиями доступов к сущностям. Кроме того, дополнительно определены следующие функции:

1) $shared\_container : C \to \{\texttt{true}, \texttt{false}\}$ — функция разделяемых контейнеров: сущность-контейнер $c \in C \setminus S$ является разделяемой, когда $shared\_container(c) = \texttt{true}$, в противном случае $shared\_container(c) = \texttt{false}$;

2) $entity\_name : (C \setminus S) \times (E \setminus S) \to NAMES$ — функция имен сущностей (не являющихся субъект-сессиями) в составе сущностей-контейнеров (также не являющихся субъект-сессиями), где $NAMES$ — некоторое множество допустимых имен сущностей, включающее элемент «» — пустая строка. При этом по определению если $e \notin H_E(c)$, то $entity\_names(c, e) =$ «», иначе $entity\_names(c, e) \neq$ «»;

3) $CCRI : E \setminus S \to \{\texttt{true}, \texttt{false}\}$ — функция, задающая способ доступа к сущностям, не являющимся субъектами, внутри контейнеров (с учетом их меток мандатных уровней целостности). Если сущность $e \in C$ является контейнером и доступ к сущностям, содержащимся внутри контейнера e, разрешен без учета уровня целостности контейнера e, то по определению выполняется равенство $CCRI(e) = \texttt{false}$, в противном случае выполняется равенство $CCRI(e) = \texttt{true}$. При этом по определению для каждой сущности $e \in E$, являющейся объектом, выполняется условие $CCRI(e) = \texttt{false}$;

4) $execute\_container : S \times E \to \{\texttt{true}, \texttt{false}\}$ — функция доступа субъект-сессии к сущностям в контейнерах, такая, что по определению для субъект-сессии $s \in S$ и сущности $e \in E$ справедливо равенство $execute\_container(s, e) = \texttt{true}$ тогда и только тогда, когда либо $e \in S$, либо $e \in E \setminus S$ и существует последовательность сущностей $e_1, \ldots, e_n \in E$, где $n \geqslant 1$, $e = e_n$, удовлетворяющих следующим условиям:

— не существует сущности-контейнера $e_0 \in E \setminus S$, такой, что $e_1 \in H_E(e_0)$;
— $e_i \in H_E(e_{i-1})$, где $1 < i \leqslant n$;
— $(e_i, execute_r) \in PA(roles(s))$ и либо $i_e(e_i) \leqslant i_s(s)$, либо $CCRI(e_i) = \texttt{false}$, где $1 \leqslant i < n$.

Таким образом, для сущностей-контейнеров задана функция разделяемых контейнеров (помечаемых в ОС атрибутом «t»), а для всех сущностей, не являющихся субъект-сессиями, определены их имена в составе сущностей-контейнеров. По аналогии с моделью систем военных сообщений [1] добавлен также мандатный атрибут целостности сущностей-контейнеров — $CCRI$, а с использованием функции $execute\_container$ заданы условия доступа субъект-сессии к сущности с учетом ее текущих прав доступа ко всем сущностям-контейнерам, в которые входит данная сущность.

С учетом дополнительных элементов модели в предположении 6 (см. [4]) уточнено, что в начальном состоянии G_0 любой системы $\Sigma(G^*, OP, G_0)$ выполняются следующие условия:

— функции UA_0, PA_0 и $roles_0$ удовлетворяют соответствующим ограничениям $Constraint_U$, $Constraint_P$, $Constraint_S$;
— для любых субъект-сессии $s \in S_0$ и сущности $e \in E_0$ если $(s, e, \alpha_a) \in A_0$ (т. е. доступ уже имеется в начальном состоянии системы), где $\alpha_a \in R_a$, то справедливо равенство $execute\_container(s, e) = \texttt{true}$ (этот доступ мог быть получен субъект-сессией s только с учетом ее текущих прав доступа ко всем сущностям-контейнерам, в которые входит сущность e).

Предположения 7 и 9 дополнены и частично переформулированы.

Предположение 7. Уровень целостности роли не превосходит уровней целостности ролей, которым она подчинена в иерархии ролей. Уровень целостности сущности, входящей в состав сущности-контейнера и не являющейся субъект-сессией, не превосходит уровня целостности сущности-контейнера. Уровни целостности сущностей, параметрически ассоциированных с учетной записью пользователя, совпадают с ее уровнем целостности. Текущий уровень целостности субъект-сессии не превосходит уровня целостности учетной записи пользователя, от имени которой она функционирует, и текущего уровня субъект-сессии, которой она подчинена в иерархии. При этом

в дополнение к предположению 1 недоверенная субъект-сессия всегда может создать недоверенную субъект-сессию с низким уровнем целостности. Уровень целостности роли не может быть выше уровня целостности учетной записи пользователя, которая на нее может быть авторизована, и текущего уровня целостности субъект-сессии, во множество текущих ролей которой она входит. Права доступа владения own_r и на запись $write_r$ к сущности, не являющейся субъект-сессией, могут принадлежать только ролям, имеющим уровень целостности не ниже, чем уровень целостности сущности. Право доступа владения own_r к субъект-сессии может принадлежать только ролям, имеющим уровень целостности не ниже, чем уровень целостности субъект-сессии. Таким образом, выполняются следующие условия:

1. Для ролей $r, r' \in R \cup AR$, если $r \leqslant r'$, то $i_r(r) \leqslant i_r(r')$.
2. Для сущностей $e, e' \in E \setminus S$, если $e \leqslant e'$, то $i_e(e) \leqslant i_e(e')$.
3. Для субъект-сессий $s, s' \in S$, если $s \leqslant s'$, то $i_s(s) \leqslant i_s(s')$.
4. Для каждой сущности $e \in]u[$, где $u \in U$, справедливо равенство $i_e(e) = i_u(u)$.
5. Для субъект-сессии $s \in S$ верно неравенство $i_s(s) \leqslant i_u(user(s))$.
6. Для учетной записи пользователя $u \in U$ и роли $r \in R$, если $r \in UA(u)$, то $i_r(r) \leqslant i_u(u)$.
7. Для субъект-сессии $s \in S$ и роли $r \in R$, если $r \in roles(s)$, то $i_r(r) \leqslant i_s(s)$.
8. Для права доступа к сущности $(e, \alpha) \in P$, где $\alpha \in \{own_r, write_r\}$, и роли $r \in R$, если $(e, \alpha) \in PA(r)$, то или $e \in E \setminus S$ и $i_e(e) \leqslant i_r(r)$, или $e \in S$, $\alpha = own_r$ и $i_s(e) \leqslant i_r(r)$.

Предположение 9. Субъект-сессии могут иметь друг к другу только доступ владения own_a. Роли могут обладать к субъект-сессиям только правом доступа владения own_r. К сущностям, не являющимся субъект-сессиями, субъект-сессии могут иметь любые виды доступа из множества R_a, а роли могут иметь к таким сущностям любые права доступа из множества R_r. Для управления доступом к сущности субъект-сессия должна получить к ней доступ владения. Для создания, переименования или удаления сущности или «жесткой» ссылки на нее в сущности-контейнере субъект-сессии необходимо иметь к сущности-контейнеру доступ на запись. Для переименования, удаления сущности или «жесткой» ссылки на сущность e в сущности-контейнере c ($e \in H_E(c)$), помеченной как разделяемая ($shared\_container(c) = \texttt{true}$), требуется наличие у субъект-сессии текущей роли, обладающей правом доступа владения own_r к сущности e. При этом для осуществления субъект-сессией любых действий над сущностью (управление к ней доступом, получения доступа, создания, удаления и др.), не являющейся субъект-сессией, требуется существование последовательности непосредственно вложенных друг в друга сущностей-контейнеров, начинающейся с некоторой сущности-«корневой контейнер» (например, корневой контейнер «/» в ОС семейства *Linux*) и заканчивающейся сущностью-контейнером, в состав которой непосредственно входит сама сущность, и наличие у субъект-сессии:

— текущих ролей, обладающих в совокупности правами доступа $execute_r$ ко всем сущностям-контейнерам этой последовательности;
— возможности получения доступа (с учетом уровня целостности субъект-сессии) внутрь всех сущностей-контейнеров этой последовательности с учетом их уровней целостности и мандатных атрибутов целостности $CCRI$.

Таким образом, с учетом предположения 5 всегда справедливо:

— для субъект-сессий s, $s' \in S$, если $(s, s', \alpha_a) \in A$, то $\alpha_a = own_a$;
— для роли $r \in R$, если $(s, \alpha_r) \in PA(r)$, где $s \in S$, то $\alpha_r = own_r$;

— для субъект-сессии $s \in S$ и сущности $e \in E$, если $(s, e, \alpha_a) \in A$, то $execute\_container(s, e) = \texttt{true}$.

Ранее при разработке ДП-моделей для отражения ситуации, когда одна субъект-сессия получает контроль над другой субъект-сессий, использовалось право доступа владения own_a. В то же время в реальных ОС такой доступ не всегда может быть явно задан как ее параметр. Например, когда субъект-сессия осуществляет отладку другой субъект-сессии, можно считать, что этот доступ задан явно, а когда первая субъект-сессия через переполнение буфера памяти (реализацию информационного потока по памяти к сущности, функционально ассоциированной с другой субъект-сессий) получила над ней контроль, доступ владения явно не предоставляется. В связи с этим используем обозначение и следующие формулировки предположений 10 и 11: $de\_facto\_own : S \to S$ — функция де-факто владения субъект-сессиями, т. е. по определению будем говорить, что субъект-сессия s де-факто владеет субъект-сессией s', когда выполняется условие $s' \in de\_facto\_own(s)$. При этом по определению всегда $s \in de\_facto\_own(s)$ (субъект-сессия де-факто владеет сама собой) и если $(s, s', own_a) \in A$, то $s' \in de\_facto\_own(s)$ (если есть де-юре владение, то есть и де-факто владение).

Предположение 10. Если субъект-сессия s реализовала информационный поток по памяти от себя к сущности, функционально ассоциированной с другой субъект-сессией s', или субъект-сессия s реализовала информационный поток по памяти к себе от всех сущностей, параметрически ассоциированных с другой субъект-сессией s', то субъект-сессия s получает де-факто владение субъект-сессией s' ($s' \in de\_facto\_own(s)$).

Предположение 11. Если субъект-сессия s де-факто владеет субъект-сессией s', то субъект-сессия s получает следующие возможности:

— использовать роли из множества текущих ролей субъект-сессии s';
— изменять множество текущих ролей субъект-сессии s';
— использовать текущий уровень целостности субъект-сессии s';
— использовать доступы субъект-сессии s';
— получать де-факто владение субъект-сессиями, которыми де-факто владеет субъект-сессия s';
— использовать административные роли субъект-сессии s' для осуществления действий над ролями и сущностями, которые позволяют ей изменять права доступа ролей субъект-сессии s';
— использовать информационные потоки, в реализации которых участвует субъект-сессия s';
— удалить субъект-сессию s'.

При этом используем обозначения:

1) $de\_facto\_roles : S \to 2^{R \cup AR}$ — функция де-факто текущих ролей субъект-сессий, при этом по определению в каждом состоянии системы G для каждой субъект-сессии $s \in S$ верно равенство $de\_facto\_roles(s) = \{r \in R \cup AR$: существует $s' \in S$, такая, что $s' \in de\_facto\_own(s)$ и $r \in roles(s')\}$;

2) $de\_facto\_rights : S \to 2^P$ — функция де-факто текущих прав доступа субъект-сессий, при этом по определению в каждом состоянии системы G для каждой субъект-сессии $s \in S$ верно равенство $de\_facto\_rights(s) = \{p \in P$: существует $r \in de\_facto\_roles(s)$, такая, что $p \in PA(r)\}$;

3) $de\_facto\_accesses : S \to 2^A$ — функция де-факто доступов субъект-сессий, при этом по определению в каждом состоянии системы G для каждой субъект-сессии $s \in S$ верно равенство $de\_facto\_accesses(s) = \{(s', e, \alpha_a) : s' \in de\_facto\_own(s), (s', e, \alpha_a) \in A\}$.

Заметим, что функция де-факто возможностей субъект-сессий ($de\_facto\_actions$) более в рамках РОСЛ ДП-модели не рассматривается.

3. Правила преобразования состояний

В рамках РОСЛ ДП-модели используются следующие правила преобразования состояний из множества OP (табл. 1 и 2), которые либо добавлены впервые, либо существенно отличаются от правил БРОС ДП-модели. По аналогии с моделью *Take-Grant* их можно неформально классифицировать на де-юре правила (правила, которые требуют реализации в ОС, то есть приводящие к «реальным» изменениям ее параметров: изменению множеств текущих ролей, прав доступа ролей, получению доступов субъект-сессий к сущностям и т. д.) и де-факто правила (правила, которые не требуют реализации в ОС, так как используются в модели для отражения факта получения субъект-сессией де-факто владения субъект-сессиями или факта реализации информационного потока по памяти или по времени).

Таким образом, в рамках РОСЛ ДП-модели заданы 18 де-юре и 11 де-факто правил преобразования состояний. Проанализируем их наиболее существенные отличия от аналогичных правил БРОС ДП-модели.

Во всех де-юре правилах, где ранее требовалось выполнение «де-факто условия» — реализация информационного потока по памяти либо от некоторой субъект-сессии к сущности i_entity, либо от сущности e, параметрически ассоциированной с ролью или учетной записью пользователя, к некоторой субъект-сессии, — теперь требуется выполнение «де-юре условия» — наличие у этой субъект-сессии доступа соответственно либо на запись к сущности i_entity, либо на чтение к сущности e. Фактические возможности также заменены на соответствующие доступы — владения, на запись или на чтение.

Правило создания сущности заменено на три правила — создания сущности-объекта, сущности-контейнера и «жесткой» ссылки на сущность. В эти правила, а также в правило переименования сущности добавлен дополнительный параметр — имя сущности ($name$). Кроме того, в правило создания сущности-контейнера $create\_container$ включены параметры t (указывающий, является ли сущность-контейнер разделяемым) и $ccri$ (мандатный атрибут целостности). В связи с этим в модель добавлено новое правило $set\_container\_attr$, позволяющее субъект-сессиям изменять параметры t и $ccri$ сущностей-контейнеров.

В условиях выполнения правил получения доступов владения, на чтение или на запись к сущности добавлена проверка прав доступа к сущностям-контейнерам, содержащим эту сущность, что отражает порядок предоставления аналогичных доступов в реальных ОС семейства *Linux*.

Де-факто правило $de\_facto\_op(x, op(y, y', \ldots))$ не имеет аналогов в других ДП-моделях и позволяет субъект-сессии x, де-факто владеющей субъект-сессиями y и y', выполнить от имени y любое де-юре правило. Данное правило позволило исключить из рассмотрения в рамках РОСЛ ДП-модели де-факто возможности субъект-сессий и более ясно описать условия применения де-юре правил преобразования состояний.

Таблица 1

Де-юре правила преобразования состояний РОСЛ ДП-модели

Правило	Исходное состояние $G = (PA, user, roles, A, F, H_E)$	Результирующее состояние $G' = (PA', user', roles', A', F', H'_E)$
1	2	3
$take\_role(x, x', \{r_j : 1 \leqslant j \leqslant k\})$	$x, x' \in S$, $r_j \in UA(user(x)) \cup \cup AUA(user(x))$, $\{(x, e, read_a) : e \in]r_j[\} \subset A$, $i_r(r_j) \leqslant i_s(x)$, $Constraint_S(roles') =$ `true`, [если $i_r(r_j) = i\_high$, то $(x', i\_entity, write_a) \in A$], где $1 \leqslant j \leqslant k$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $A' = A$, $H'_E = H_E$, $roles'(x) = roles(x) \cup \{r_j : 1 \leqslant j \leqslant k\}$ и для $s \in S \setminus \setminus \{x\}$ выполняется равенство $roles'(s) = roles(s)$, если $x \in (N_S \cup \cup NF_S) \cap S$ и $\{r_j : 1 \leqslant j \leqslant k\} \setminus roles(x) \neq \varnothing$, то $F' = F \cup \{(x, s, write_t) : s \in (N_S \cup NF_S) \cap S$, $x \neq s$ и $x \in de\_facto\_own(s)\}$, иначе $F' = F$
$remove\_role(x, x', \{r_j : 1 \leqslant j \leqslant k\})$	$x, x' \in S$, $r_j \in roles(x)$, $\{(x, e, read_a) : e \in]r_j[\} \subset A$, $Constraint_S(roles') =$ `true`, [если $i_r(r_j) = i\_high$, то $(x', i\_entity, write_a) \in A$], где $1 \leqslant j \leqslant k$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $A' = A$, $H'_E = H_E$, $roles'(x) = roles(x) \setminus \{r_j : 1 \leqslant j \leqslant k\}$ и для $s \in S \setminus \{x\}$ выполняется равенство $roles'(s) = roles(s)$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \cup\{(x, s, write_t)$: $s \in (N_S \cup NF_S)$, $x \neq s$ и $x \in de\_facto\_own(s)\}$, иначе $F' = F$
$grant\_right(x, x', r, \{(y_j, \alpha_{r_j}) : 1 \leqslant j \leqslant k\})$	$x, x' \in S$, $y_j \in E$, $r \in can\_manage\_rights(roles(x) \cap AR)$, $(x, y_j, own_a) \in A$, $i_r(r) \leqslant i_s(x)$, [если $y_j \in S$, то $\alpha_{r_j} = own_r$ и $i_s(y_j) \leqslant i_r(r)$], [если $y_j \in E \setminus S$ и $\alpha_{r_j} \in \{own_r, write_r\}$, то $i_e(y_j) \leqslant i_r(r)$], [$Constraint_P(PA') = true$], [если $i_e(y_j) = i\_high$, то $(x', i\_entity, write_a) \in A$], где $1 \leqslant j \leqslant k$	$S' = S$, $E' = E$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, $PA'(r) = PA(r) \cup \{(y_j, \alpha_{r_j}) : 1 \leqslant j \leqslant k\}$ и для $r' \in R \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \cup\{(x, s, write_t)$: $s \in (N_S \cup NF_S) \cap S$, $x \neq s$, $r \in de\_facto\_roles(s)$, и существует j, такое, что $1 \leqslant j \leqslant k$, $(y_j, \alpha_{r_j}) \notin PA(r)$ и $(y_j, \alpha_a) \in de\_facto\_accesses(s)$, где $\alpha_a \in R_a\}$, иначе $F' = F$
$remove\_right(x, x', r, \{(y_j, \alpha_{r_j}) : 1 \leqslant j \leqslant k\})$	$x, x' \in S$, $y_j \in E$, $(y_j, \alpha_{r_j}) \in PA(r)$, $r \in can\_manage\_rights(roles(x) \cap AR)$, $(x, y_j, own_a) \in A$, $i_r(r) \leqslant i_s(x)$, $Constraint_P(PA') =$ `true`, [если $i_e(y_j) = i\_high$, то $(x', i\_entity, write_a) \in A$], где $1 \leqslant j \leqslant k$	$S' = S$, $E' = E$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, $PA'(r) = PA(r) \setminus \{(y_j, \alpha_{r_j}) : 1 \leqslant j \leqslant k\}$ и для $r' \in R \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \cup\{(x, s, write_t) : s \in (N_S \cup NF_S) \cap S$, $x \neq s$, $r \in de\_facto\_roles(s)$, и существует j, такое, что $1 \leqslant j \leqslant k$ и $(y_j, \alpha_a) \in de\_facto\_accesses(s)$, где $\alpha_a \in R_a\}$, иначе $F' = F$
$create\_object(x, x', r, y, yi, name, z)$	$x, x' \in S$, $y \notin E$, $z \in C \setminus S$, $name \in NAME \setminus \{«»\}$, $r \in \in can\_manage\_rights(roles(x) \cap AR)$, $(x, z, write_a) \in A$, $yi \leqslant \leqslant i_r(r) \leqslant i_s(x)$, $yi \leqslant i_e(z)$, $Constraint_P(PA') =$ `true`, [если $i_e(z) = i\_high$, то $(x', i\_entity, write_a) \in A$]	$S' = S$, $E' = E \cup \{y\}$ ($O' = O \cup \{y\}$, $C' = C$), при этом $y \notin UE \cup RE$, $user' = user$, $roles' = = roles$, $A' = A$, $i'_e(y) = yi$, $CCRI'(y) =$ `false`, $entity\_name(z, y) = name$, $PA'(r) = PA(r) \cup \{(y, ownr)\}$ и для $r' \in R \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$, $H'_E(z) = H_E(z) \cup \{y\}$, $H'_E(y) = \varnothing$, для $e \in \in E \setminus \{z\}$ выполняется равенство $H'_E(e) = = H_E(e)$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t) : e \in E$ и $y \leqslant e\} \cup \cup\{(x, s, write_t) : s \in (N_S \cup NF_S) \cap S, x \neq \neq s, r \in de\_facto\_roles(s)$ и $(z, \alpha_a) \in de\_facto\_accesses(s)$, где $\alpha_a \in R_a\}$, иначе $F' = F$

Продолжение табл. 1

1	2	3
create_container $(x,\ x',\ r,\ y,\ yi,$ $ccri,\ t,\ name,\ z)$	$x, x' \in S$, $y \notin E$, $z \in C \setminus S$, $ccri, t \in \{\texttt{true}, \texttt{false}\}$, $name \in NAME \setminus \{«»\}$, $r \in can\_manage\_rights(roles(x) \cap AR)$, $(x, z, write_a) \in A$, $yi \leqslant i_r(r) \leqslant i_s(x)$, $yi \leqslant i_e(z)$, $Constraint_P(PA') = \texttt{true}$, [если $i_e(z) = i\_high$, то $(x', i\_entity, write_a) \in A$]	$S' = S$, $E' = E \cup \{y\}$ ($O' = O$, $C' = C \cup \{y\}$), при этом $y \notin UE \cup RE$, $user' = user$, $roles' = roles$, $A' = A$, $i'_e(y) = yi$, $CCRI'(y) = ccri$, $entity\_name(z, y) = name$, $PA'(r) = PA(r) \cup \{(y, ownr)\}$ и для $r' \in R \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$, $H'_E(z) = H_E(z) \cup \{y\}$, $H'_E(y) = \varnothing$, для $e \in E \setminus \{z\}$ выполняется равенство $H'_E(e) = H_E(e)$, $shared\_container'(y) = t$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t) : e \in E$ и $y \leqslant e\} \cup \{(x, s, write_t) : s \in (N_S \cup \cup NF_S) \cap S$, $x \neq s$, $r \in de\_facto\_roles(s)$ и $(z, \alpha_a) \in de\_facto\_accesses(s)$, где $\alpha_a \in R_a\}$, иначе $F' = F$
create_hard_link$(x, x', y,$ $name, z)$	$x, x' \in S$, $y \in O \setminus S$, $z \in C \setminus S$, $name \in NAME \setminus \{«»\}$, $y \notin UE \cup RE$, $(x, z, write_a) \in A$, $i_e(y) \leqslant i_e(z)$, [если $i_e(z) = i\_high$, то $(x', i\_entity, write_a) \in A$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $entity\_name(z, y) = name$, $H'_E(z) = H_E(z) \cup \{y\}$, для $e \in E \setminus \{z\}$ выполняется равенство $H'_E(e) = H_E(e)$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t)$: $e \in E$ и $y \leqslant e\}$, иначе $F' = F$
rename_entity $(x, x', y, name, z)$	$x, x' \in S$, $y, z \in E \setminus S$, $y \in H_E(z)$, $name \in NAME \setminus \{«»\}$, $(x, z, write_a) \in A$, [если $i_e(z) = i\_high$, то $(x', i\_entity, write_a) \in A$], [если $shared\_container(z) = \texttt{true}$, то $(y, own_r) \in PA(roles(x))$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, $entity\_name(z, y) = name$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t)$: $e \in E$, $x \neq e$, и или $e \leqslant y$, или $e = z\} \cup \{(x, s, write_t)$: $s \in (N_S \cup NF_S) \cap S$, $x \neq s$, $(e, \alpha_a) \in de\_facto\_accesses(s)$, где $e \in E$, $e \leqslant y$, $\alpha_a \in R_a\}$, иначе $F' = F$
set_container_attr$(x, x', y, ccri, t)$	$x, x' \in S$, $y \in C \setminus S$, $ccri, t \in \{\texttt{true}, \texttt{false}\}$, $(x, y, write_a) \in A$, [если $i_e(y) = i\_high$, то $(x', i\_entity, write_a) \in A$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, $CCRI'(y) = ccri$, $shared\_container'(y) = t$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t)$: $e \in E$, $x \neq e$ и $e \leqslant y\} \cup \{(x, s, write_t)$: $s \in (N_S \cup NF_S) \cap S$, $x \neq s$, $(e, \alpha_a) \in de\_facto\_accesses(s)$, где $e \in E$, $e \leqslant y$, $\alpha_a \in R_a\}$, иначе $F' = F$
delete_entity$(x,$ $x',\ y,\ z)$	$x, x' \in S$, $y \in E \setminus S$, $z \in C \setminus S$, $y \in H_E(z)$, $H_E(y) = \varnothing$, [не существует $z' \in$, такой, что $z' \neq z$ и $y \in H_E(z')$], [$y \notin (UE \cup RE)$, $(x, z, write_a) \in A$, $Constraint_P(PA') = \texttt{true}$], [если $i_e(z) = i\_high$, то $(x',$ $i\_entity,\ write_a) \in A$], [если $shared\_container(z) = \texttt{true}$, то $(y,\ own_r) \in PA(roles(x))$]	$S' = S$, $E' = E \setminus \{y\}$, $user' = user$, $roles' = roles$, $H'_E(z) = H_E(z) \setminus \{y\}$, для всех $e \in E' \setminus \{z\}$ выполняется равенство $H'_E(e) = H_E(e)$, для $r \in R$ выполняется равенство $PA'(r) = PA(r) \setminus \{(y, \alpha_r)$: $\alpha_r \in R_r\}$, $A' = A \setminus \{(s, y, \alpha_a)$: $s \in S$, $\alpha_a \in R_a\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = (F \cup \cup \{(x, e, write_t) : e \in E$, $x \neq e$ и $z \leqslant e\} \cup \cup \{(x, s, write_t) : s \in (N_S \cup NF_S) \cap S'$, $x \neq s$ и $(y, \alpha_a) \in de\_facto\_accesses(s)$, где $\alpha_a \in R_a\}) \setminus (\{(e, y, \alpha_f) : e \in E, \alpha_f \in R_f\} \cup \cup \{(y, e, \alpha_f) : e \in E, \alpha_f \in R_f\})$, иначе $F' = F$

Продолжение табл. 1

1	2	3
$delete\_hard\_link(x, x', y, z)$	$x, x' \in S$, $y \in O \setminus S$, $z \in C \setminus S$, $y \in H_E(z)$, [не существует $z' \in C$, такой, что $z' \neq z$ и $y \in H_E(z')$], [$y \notin (UE \cup RE)$, $(x, z, write_a) \in A$], [если $i_e(z) = i\_high$, то $(x', i\_entity, write_a) \in A$], [если $shared\_container(z) = \texttt{true}$, то $(y, own_r) \in PA(roles(x))$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $H'_E(z) = H_E(z) \setminus \{y\}$, для всех $e \in E' \setminus \{z\}$ выполняется равенство $H'_E(e) = H_E(e)$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = (F \cup \{(x, e, write_t) : e \in E, x \neq e$ и или $y \leqslant e$, или $z \leqslant e\}$, иначе $F' = F$
$create\_first\_session(x, x', u, r, y, z, zi)$	$x, x' \in S$, $u \in U$, $y \in E$, $z \notin E$, $(y, execute_r) \in PA(roles(x))$, $execute\_container(x, y)=\texttt{true}$, $r \in can\_manage\_rights(roles(x) \cap AR)$, $zi \leqslant i_u(u)$, $zi \leqslant i_r(r)$, $\{(x, e, read_a)$: $e \in]u[\} \subset A$, $Constraint_P(PA') = \texttt{true}$, $Constraint_S(roles') = \texttt{true}$, [если $zi = i\_high$, то $(x', i\_entity, write_a) \in A$]	$S' = S \cup \{z\}$, $E' = E \cup \{z\}$, $A' = A \cup \{(x, z, own_a)\}$, $i'_s(z) = zi$, $user'(z) = u$, для $s \in S$ выполняется равенство $user'(s) = user(s)$, $roles'(z) = \varnothing$, для $s \in S$ выполняется равенство $roles'(s) = roles(s)$, $[z] = fa(u, y)$, $]z[= fp(u, y)$, $PA'(r) = PA(r) \cup \{(z, own_r)\}$ и для $r' \in R \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$, $H'_E(z) = \varnothing$, для $e \in E$ выполняется равенство $H'_E(e) = H_E(e)$, если $x \in (NS \cup NF_S) \cap S$, то $F' = F \cup \{(z, x, write_t), (x, z, write_t)\} \cup \{(x, e, write_t) : e \in E$ и $y \leqslant e\} \cup \cup \{(x, s, write_t) : s \in (NS \cup NF_S) \cap S$, $x \neq s$ и $r \in de\_facto\_roles(s)\}$, иначе $F' = F$
$create\_session(x, x', r, y, z, zi)$	$x, x' \in S$, $y \in E$, $z \notin E$, $(y, execute_r) \in PA(roles(x))$, $execute\_container(x, y)=\texttt{true}$, $r \in can\_manage\_rights(roles(x) \cap AR)$, $zi \leqslant i_r(r) \leqslant i_s(x)$, $Constraint_P(PA') = \texttt{true}$, $Constraint_S(roles') = \texttt{true}$, [если $zi = i\_high$, то $(x', i\_entity, write_a) \in A$]	$S' = S \cup \{z\}$, $E' = E \cup \{z\}$, $A' = A \cup \{(x, z, own_a)\}$, $i'_s(z) = zi$, $user'(z) = user(x)$, для $s \in S$ выполняется равенство $user'(s) = user(s)$, $roles'(z) = \varnothing$, для $s \in S$ выполняется равенство $roles'(s) = roles(s)$, $[z] = fa(user(x), y)$, $]z[= fp(user(x), y)$, $H'_E(x) = H_E(x) \cup \{z\}$, $H'_E(z) = \varnothing$, для $e \in E \setminus \{x\}$ выполняется равенство $H'_E(e) = H_E(e)$, $PA'(r) = PA(r) \cup \{(z, own_r)\}$ и для $r' \in R \setminus \{r\}$ выполняется равенство $PA'(r') = PA(r')$, если $x \in (NS \cup NF_S) \cap S$, то $F' = F \cup \{(z, x, write_t), (x, z, write_t)\} \cup \{(x, e, write_t) : e \in E$ и $y \leqslant e\} \cup \{(x, s, write_t) : s \in (NS \cup NF_S) \cap S$, $x \neq s$ и $r \in de\_facto\_roles(s)\}$, иначе $F' = F$
$delete\_session(x, x', z)$	$x, x', z \in S$, $(x, z, own_a) \in A$, $i_s(z) \leqslant i_s(x)$, $Constraint_P(PA') = \texttt{true}$, $Constraint_S(roles') = \texttt{true}$, [если $i_s(z) = i\_high$, то $(x', i\_entity, write_a) \in A$]	$S' = S \setminus \{z\}$, $E' = E \setminus \{z\}$, для $s \in S'$ верно $user'(s) = user(s)$, $roles'(s) = roles(s)$, для $z' \in S$, такой, что $z \in H_E(z')$, верно $H'_E(z') = (H_E(z') \setminus \{z\}) \cup H_E(z)$, при этом выполняется: для $e \in E' \setminus \{z'\}$ верно $H'_E(e) = H_E(e)$, $PA'(r) = PA(r) \setminus \{(z, own_r)\}$, и для $r' \in R \setminus \{r\}$ верно $PA'(r') = PA(r')$, $A' = A \setminus (\{(z, e, \alpha_a) : e \in E, \alpha_a \in R_a\} \cup \{(s, z, own_a) : s \in S\})$, если $x \in (NS \cup NF_S) \cap S$, то $F' = (F \cup \{(x, s, write_t) : e \in E$, $x \neq e$ и $z < e\} \cup \{(x, s, write_t) : s \in (NS \cup NF_S) \cap S$, $x \neq s$ и $z \in de\_facto\_own(s)\}) \setminus (\{(z, e, \alpha_f) : e \in E, \alpha_f \in R_f\} \cup \{(e, z, \alpha_f) : e \in E, \alpha_f \in R_f\})$, иначе $F' = F$

Окончание табл. 1

1	2	3
$access\_own(x, x', y)$	$x, x' \in S$, $y \in E$, $x \neq y$, $(y, own_r) \in PA(roles(x))$, $execute\_container(x, y) =$ `true`, [если $y \in S$, то $i_s(y) \leqslant i_s(x)$], [если $y \in E \setminus S$, то $i_e(y) \leqslant i_s(x)$], [если ($y \in S$ и $i_s(y) = i\_high$) или ($y \in E \setminus S$ и $i_e(y) = i\_high$), то $(x', i\_entity, write_a) \in A$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $H'_E = H_E$, $A' = A \cup \{(x, y, own_a)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t)$: $e \in E$, $x \neq e$ и $y \leqslant e\} \cup \{(x, s, write_t)$: $s \in (N_S \cup NF_S) \cap S$, $x \neq s$ и [если $y \in E \setminus S$, то $(y, own_a) \in de\_facto\_accesses(s)$], [если $y \in S$, то $y \in de\_facto\_own(s)$]$\}$, иначе $F' = F$
$access\_read(x, x', y)$	$x, x' \in S$, $y \in E \setminus S$, $(y, read_r) \in PA(roles(x))$, $execute\_container(x, y) =$ `true`	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $H'_E = H_E$, $A' = A \cup \{(x, y, read_a)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(y, x, write_m)\} \cup \{(x, e, write_t) : e \in E$, $x \neq e$ и $y \leqslant e\}$, иначе $F' = F \cup \{(y, x, write_m)\}$
$access\_write(x, x', y)$	$x, x' \in S$, $y \in E \setminus S$, $(y, write_r) \in PA(roles(x))$, $execute\_container(x, y) =$ `true`, $i_e(y) \leqslant i_s(x)$, [если $i_e(y) = i\_high$, то $(x', i\_entity, write_a) \in A$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $H'_E = H_E$, $A' = A \cup \{(x, y, write_a)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, y, write_m)\} \cup \{(x, e, write_t)$: $e \in E$, $x \neq e$ и $y \leqslant e\}$, иначе $F' = F \cup \{(x, y, write_m)\}$
$delete\_access(x, x', y, \alpha_a)$	$x, x' \in S$, $y \in E \setminus S$, $(x, y, \alpha_a) \in A$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $H'_E = H_E$, $A' = A \setminus \{(x, y, \alpha_a)\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t) : e \in E$, $x \neq e$ и $y \leqslant e\}$, иначе $F' = F$

Таблица 2

Де-факто правила преобразования состояний РОСЛ ДП-модели

Правило	Исходное состояние $G = (PA, user, roles, A, F, H_E)$	Результирующее состояние $G' = (PA', user', roles', A', F', H'_E)$
1	2	3
$de\_facto\_op(x, op(y, y', \ldots))$	$x, y, y' \in S$, $y, y' \in de\_facto\_own(x)$, выполняются условия применения де-юре правила преобразования состояний $op(y, y', \ldots)$, заданные в табл. 1	Соответствуют результатам применения правила $op(y, y', \ldots)$
$control(x, y, z)$	$x, y \in S$, $x \neq y$, $z \in [y]$ и или $x = z$, или $(x, z, write_m) \in F$, или $z \in S$ и $z \in de\_facto\_own(x)$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, $de\_facto\_own'(x) = de\_facto\_own(x) \cup \{y\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t)$: $e \in E$, $x \neq e$ и $y \leqslant e\}$, иначе $F' = F$
$know(x, y)$	$x, y \in S$, $x \neq y$, и для каждой $e \in]y[$ существует $(e, x, write_m) \in F$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, $de\_facto\_own'(x) = de\_facto\_own(x) \cup \{y\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t)$: $e \in E$, $x \neq e$ и $y \leqslant e\}$, иначе $F' = F$
$take\_access\_own(x, y, z)$	$x, y, z \in S$, $y \in de\_facto\_own(x)$, $z \in de\_facto\_own(y)$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, $de\_facto\_own'(x) = de\_facto\_own(x) \cup \{z\}$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t)$: $e \in E$, $x \neq e$ и $z \leqslant e\}$, иначе $F' = F$

Окончание табл. 2

1	2	3
$flow\_memory\_access(x, y, \alpha_a)$	$x \in S$, $y \in E$, $(y, \alpha_a) \in$ $\in de\_facto\_accesses(x)$, где $\alpha_a \in \{read_a, write_a\}$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, если $\alpha_a = read_a$, то $F' = F \cup \{(y, x, write_m)\}$, если $\alpha_a = write_a$, то $F' = F \cup \{(x, y, write_m)\}$
$flow\_time\_access(x, y)$	$x \in S$, $y \in E$, $(y, \alpha_a) \in$ $\in de\_facto\_accesses(x)$ или [$y \in S$ и $y \in de\_facto\_own(x)$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, write_t)$: $e \in E$, $x \neq e$ и $y \leqslant e\} \cup \{(e, x, write_t)$: $x \neq e$ и $e \leqslant y\}$, иначе $F' = F$
$flow(x, y, y', z)$	$x, z \in S$, $y, y' \in E$, $x \neq z$, $y \leqslant y'$, [или $x = y$, или $(y, \alpha_a) \in$ $\in de\_facto\_accesses(x)$, или $y \in S$ и $y \in de\_facto\_own(x)$], [или $z = y'$, или $(y', \beta_a) \in de\_facto\_accesses(z)$, или $y' \in S$ и $y' \in de\_facto\_own(z)$], где $\alpha_a, \beta_a \in R_a$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, если x, $z \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, z, write_t)$, $(z, x, write_t)\}$, иначе $F' = F$
$find(x, y, z)$	$x, y \in S$, $z \in E$, $x \neq z$, [$(x, y, \alpha) \in F$, где $\alpha \in \{write_m, write_t\}$], и [или $(z, \beta) \in de\_facto\_accesses(y)$, где $\beta = write_a$, или $(y, z, \beta) \in F$, где $\beta \in \{write_m, write_t\}$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, если $write_t \notin \{\alpha, \beta\}$, то $F' = F \cup \{(x, z, write_m)\}$, если $write_t \in \{\alpha, \beta\}$ и $x, y \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, z, write_t)\}$, иначе $F' = F$
$post(x, y, z)$	$x, z \in S$, $y \in E$, $x \neq z$, $(y, read_a) \in de\_facto\_accesses(z)$ и [или $(y, \alpha) \in de\_facto\_accesses(x)$, где $\alpha = write_a$, или $(x, y, \alpha) \in F$, где $\alpha \in \{write_m, write_t\}$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, если $\alpha \neq write_t$, то $F' = F \cup \{(x, z, write_m)\}$, если $\alpha = write_t$ и $x, z \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, z, write_t)\}$, иначе $F' = F$
$pass(x, y, z)$	$y \in S$, $x, z \in E$, $x \neq z$, $(x, read_a) \in de\_facto\_accesses(y)$ и [или $(z, \alpha) \in de\_facto\_accesses(y)$, где $\alpha = write_a$, или $(y, z, \alpha) \in F$, где $\alpha \in \{write_m, write_t\}$]	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, если $\alpha \neq write_t$, то $F' = F \cup \{(x, z, write_m)\}$, если $\alpha = write_t$ и $y \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, z, write_t)\}$, иначе $F' = F$
$take\_flow(x, y)$	$x, y \in S$, $x \neq y$, $y \in de\_facto\_own(x)$	$S' = S$, $E' = E$, $PA' = PA$, $user' = user$, $roles' = roles$, $A' = A$, $H'_E = H_E$, если $x \in (N_S \cup NF_S) \cap S$, то $F' = F \cup \{(x, e, \alpha)$: $(y, e, \alpha) \in F$, $e \in E$, $\alpha \in \{write_m, write_t\}\}$, иначе $F' = F \cup \{(x, e, write_m)$: $(y, e, write_m) \in F$, $e \in E\}$

Доступы владения в де-факто правилах $take\_access\_own$, $flow$ и $take\_flow$ заменены на де-факто владение. Новое правило $flow\_memory\_access$ позволяет субъект-сессии, имеющей де-факто доступ на чтение или на запись к сущности, реализовать между ними соответствующий информационный поток по памяти, а правило $flow\_time\_access$ позволяет субъект-сессии, имеющей любой де-факто доступ к сущности или (в случае, когда сущность является субъект-сессией) де-факто владеющей ею, реализовать информационный поток по времени ко всем сущностям и от всех сущностей, которым в иерархии подчинена данная сущность.

Таким образом, зависимости условий и результатов применения де-юре и де-факто правил преобразования состояний РОСЛ ДП-модели существенно изменились. Схе-

ма этих зависимостей показана на рис. 1, на котором сплошными линиями показаны зависимости, возникающие при применении де-юре правил (за исключением информационных потоков), а прерывистыми линиями — зависимости, возникающие при применении де-факто правил или в результате получения информационных потоков при применении де-юре правил.

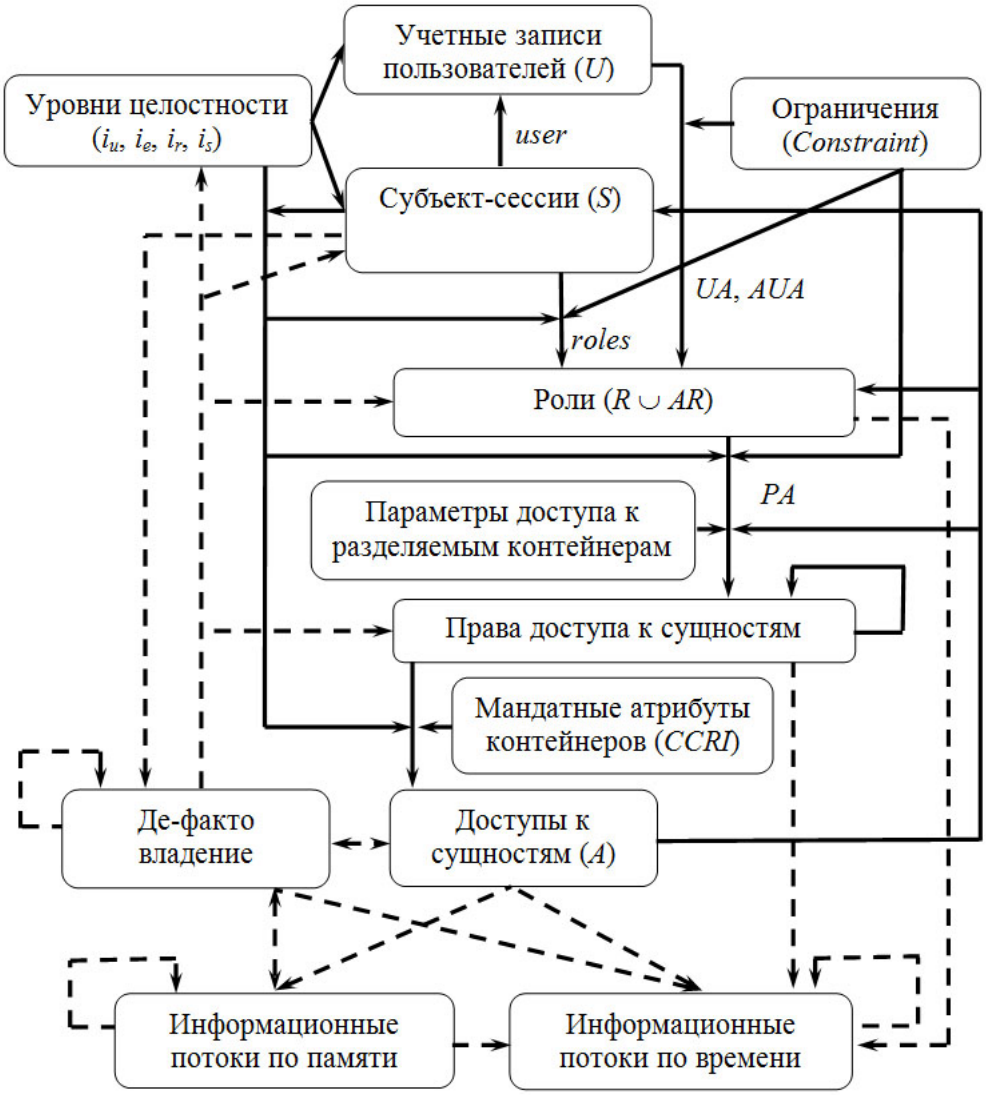


Рис. 1. Зависимости условий и результатов применения правил преобразования состояний РОСЛ ДП-модели

4. Выполнение ограничений и требований мандатного контроля целостности на траекториях системы

Поскольку в рамках РОСЛ ДП-модели, в отличие от других ролевых ДП-моделей, используется механизм ограничений, обоснуем следующее утверждение.

Утверждение 1. Пусть G_0 — начальное состояние системы $\Sigma(G^*, OP, G_0)$, в котором функции $(i_u, i_e, i_r, i_s)_0$ удовлетворяют условиям предположения 7. Тогда в любом состоянии G_N любой траектории $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$, где $op_1, \ldots, op_N$ — правила преобразования состояний и $N \geqslant 0$, функции UA_N, PA_N и $roles_N$ удовлетворяют

соответствующим ограничениям $Constraint_U$, $Constraint_P$ и $Constraint_S$ и функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условиям предположения 7.

Доказательство. Докажем утверждение индукцией по длине N траектории функционирования системы.

Пусть $N = 0$, тогда по предположению 6 в состоянии G_0 функции UA_0, PA_0 и $roles_0$ удовлетворяют соответствующим ограничениям $Constraint_U$, $Constraint_P$ и $Constraint_S$ и по условию функции $(i_u, i_e, i_r, i_s)_0$ удовлетворяют условиям предположения 7.

Пусть $N > 0$ и утверждение верно для всех траекторий длины $0 \leqslant L < N$. Пусть $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ — траектория функционирования системы длины N. По предположению индукции в состоянии G_{N-1} функции UA_{N-1}, PA_{N-1} и $roles_{N-1}$ удовлетворяют соответствующим ограничениям $Constraint_U$, $Constraint_P$ и $Constraint_S$ и функции $(i_u, i_e, i_r, i_s)_{N-1}$ удовлетворяют условиям предположения 7.

Рассмотрим правило преобразования состояний op_N. Если условия его применения не выполняются в состоянии G_{N-1}, то по определению правил преобразования состояний справедливо равенство $G_{N-1} = G_N$ и по предположению индукции в состоянии G_N функции UA_N, PA_N и $roles_N$ удовлетворяют соответствующим ограничениям $Constraint_U$, $Constraint_P$ и $Constraint_S$ и функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условиям предположения 7. Пусть условия применения правила преобразования состояний op_N выполняются в состоянии G_{N-1}.

Де-факто правила (за исключением правила $de\_facto\_op(x, op(y, y', \ldots))$ не изменяют значения функций UA, PA, $roles$, (i_u, i_e, i_r, i_s), множества сущностей, субъект-сессий, текущих ролей субъект-сессий, прав доступа ролей, отношения подчиненности в иерархии на множествах сущностей или субъект-сессий. Таким образом, если op_N является де-факто правилом (за исключением правила $de\_facto\_op(x, op(y, y', \ldots))$, по предположению индукции в состоянии G_{N-1} функции UA_{N-1}, PA_{N-1} и $roles_{N-1}$ удовлетворяют соответствующим ограничениям $Constraint_U$, $Constraint_P$ и $Constraint_S$ и функции $(i_u, i_e, i_r, i_s)_{N-1}$ удовлетворяют условиям предположения 7, то в состоянии G_N функции UA_N, PA_N и $roles_N$ также удовлетворяют соответствующим ограничениям $Constraint_U$, $Constraint_P$ и $Constraint_S$ и функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условиям предположения 7.

Если op_N является де-факто правилом $de\_facto\_op(x, op(y, y', \ldots))$, то результаты его применения совпадают с результатами применения де-юре правила $op(y, y', \ldots)$, поэтому далее при обосновании шага индукции будем считать, что op_N — де-юре правило.

В рамках РОСЛ ДП-модели не заданы де-юре правила преобразования состояний, изменяющие значение функций UA, i_u, i_r и множества UE. Следовательно, поскольку в состоянии G_{N-1} функция UA_{N-1} удовлетворяет ограничениям $Constraint_U$ и функции $(i_u, i_e, i_r, i_s)_{N-1}$ удовлетворяют условиям предположения 7, то в состоянии G_N функция UA_N удовлетворяет ограничениям $Constraint_U$ и функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условиям 1, 4, 6 предположения 7.

Если op_N является де-юре правилом вида $create\_hard\_link(x, x', y, name, z)$, $delete\_hard\_link(x, x', y, z)$, $rename\_entity(x, x', y, name, z)$, $set\_container\_attr(x, x', y, ccri, t)$, $access\_own(x, x', y)$, $access\_read(x, x', y)$, $access\_write(x, x', y)$ или $delete\_access(x, x', y, \alpha_a)$, то оно не изменяет значения функций PA и $roles$. Следовательно, поскольку функции PA_{N-1} и $roles_{N-1}$ соответственно удовлетворяют ограничениям $Constraint_P$ и $Constraint_S$ в состоянии G_{N-1}, то функции PA_N и $roles_N$ соответственно удовлетворяют им в состоянии G_N.

Если op_N является де-юре правилом вида $take\_roles(x, x', \{r_j : 1 \leqslant j \leqslant k\})$ или $remove\_roles(x, x', \{r_j : 1 \leqslant j \leqslant k\})$, то оно не изменяет значение функции PA. Следовательно, поскольку функция PA_{N-1} удовлетворяет ограничениям $Constraint_P$ в состоянии G_{N-1}, то функция PA_N удовлетворяет им в состоянии G_N. При этом соответствие функции $roles_N$ ограничениям $Constraint_S$ в состоянии G_N следует из условий применения правил.

Если op_N является де-юре правилом вида $grant\_rights(x, x', r, \{(y_j, \alpha_{r_j}) : 1 \leqslant j \leqslant k\})$, $remove\_rights(x, x', r, \{(y_j, \alpha_{r_j}) : 1 \leqslant j \leqslant k\})$, $create\_object(x, x', r, y, yi, name, z)$, $create\_container(x, x', r, y, yi, ccri, t, name, z)$ или $delete\_entity(x, x', y, z)$, то оно не изменяет значение функции $roles$. Следовательно, поскольку функция $roles_{N-1}$ удовлетворяет ограничениям $Constraint_S$ в состоянии G_{N-1}, то функция $roles_N$ удовлетворяет им в состоянии G_N. При этом соответствие функции PA_N ограничениям $Constraint_P$ в состоянии G_N следует из условий применения правил.

Если op_N является де-юре правилом вида $create\_first\_session(x, x', u, r, y, z, zi)$, $create\_session(x, x', r, y, z, zi)$ или $delete\_session(x, x', z)$, то из условий применения правил следует, что функции PA_N и $roles_N$ удовлетворяют соответственно ограничениям $Constraint_P$ и $Constraint_S$ в состоянии G_N.

Таким образом, обосновано, что в состоянии G_N функции UA_N, PA_N и $roles_N$ удовлетворяют соответствующим ограничениям $Constraint_U$, $Constraint_P$ и $Constraint_S$.

Если op_N является де-юре правилом вида $rename\_entity(x, x', y, name, z)$, $set\_container\_attr(x, x', y, ccri, t)$, $access\_own(x, x', y)$, $access\_read(x, x', y)$, $access\_write(x, x', y)$ или $delete\_access(x, x', y, \alpha_a)$, то оно не изменяет множества сущностей, субъект-сессий, текущих ролей субъект-сессий, прав доступа ролей, отношения подчиненности в иерархии на множествах сущностей или субъект-сессий. Следовательно, по предположению индукции в состоянии G_N функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условиям 2, 3, 5, 7 и 8 предположения 7.

Если op_N является де-юре правилом вида $remove\_roles(x, x', \{r_j : 1 \leqslant j \leqslant k\})$, $remove\_rights(x, x', \{(y_j, \alpha_{r_j}) : 1 \leqslant j \leqslant k\})$, $delete\_entity(x, x', y, z)$, $delete\_hard\_link(x, x', y, z)$ или $delete\_session(x, x', z)$, то оно не добавляет новых элементов во множества сущностей, субъект-сессий, текущих ролей субъект-сессий, прав доступа ролей, не изменяет отношение подчиненности в иерархии на множествах сущностей или субъект-сессий, остающихся в последующем состоянии системы. Следовательно, по предположению индукции в состоянии G_N функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условиям 2, 3, 5, 7 и 8 предположения 7.

Если op_N является де-юре правилом вида $take\_roles(x, x', \{r_j : 1 \leqslant j \leqslant k\})$, то оно не добавляет новые элементы во множества сущностей, субъект-сессий, прав доступа ролей, не изменяет отношение подчиненности в иерархии на множествах сущностей или субъект-сессий в последующем состоянии системы. Значит, по предположению индукции в состоянии G_N функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условиям 2, 3, 5 и 8 предположения 7. Выполнение условия 7 предположения 7 следует из предположения индукции и из условий и результатов применения правила.

Если op_N является де-юре правилом вида $grant\_rights(x, x', r, \{(y_j, \alpha_{r_j}) : 1 \leqslant j \leqslant k\})$, то оно не добавляет новые элементы во множества сущностей, субъект-сессий, текущих ролей субъект-сессий, не изменяет отношение подчиненности в иерархии на множествах сущностей или субъект-сессий в последующем состоянии системы. Значит, по предположению индукции в состоянии G_N функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условиям 2, 3, 5 и 7 предположения 7. Выполнение условия 8 предположения 7 следует из предположения индукции и из условий и результатов применения правила.

Если op_N является де-юре правилом вида $create\_object(x, x', r, y, yi, name, z)$, $create\_container(x, x', r, y, yi, ccri, t, name, z)$ или $create\_hard\_link(x, x', y, name, z)$, то оно не добавляет новые элементы во множества субъект-сессий, текущих ролей субъект-сессий, не изменяет отношение подчиненности в иерархии на множестве субъект-сессий в последующем состоянии системы. Значит, по предположению индукции в состоянии G_N функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условиям 3, 5 и 7 предположения 7. Выполнение условий 2 и 8 предположения 7 следует из предположения индукции и из условий и результатов применения правил.

Если op_N является де-юре правилом вида $create\_first\_session(x, x', u, r, y, z, zi)$ или $create\_session(x, x', r, y, z, zi)$, то оно не добавляет новые элементы во множества сущностей (не являющихся субъектами) и не изменяет отношение подчиненности в иерархии на множестве сущностей в последующем состоянии системы. Значит, по предположению индукции в состоянии G_N функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условию 2 предположения 7. Выполнение условий 3, 5, 7 и 8 предположения 7 следует из предположения индукции и из условий и результатов применения правила.

Таким образом, обосновано, что в состоянии G_N функции $(i_u, i_e, i_r, i_s)_N$ удовлетворяют условиям 1–8 предположения 7. Шаг индукции доказан. ■

5. Монотонные и немонотонные правила преобразования состояний. Инвариантность ограничений относительно немонотонных правил

По аналогии с существующими ДП-моделями дадим определение.

Определение 1. Монотонное правило преобразования состояний — правило преобразования состояний из множества OP, применение которого не приводит к удалению из состояний:

— ролей из множества текущих ролей субъект-сессий;
— прав доступа ролей к сущностям;
— субъект-сессий, сущностей или «жестких» ссылок на сущности-объекты;
— доступов субъект-сессий к сущностям;
— информационных потоков.

По определению в соответствии с условиями и результатами применения правил преобразования состояний, заданных в табл. 1 и 2, монотонными будут являться следующие правила: $take\_roles(x, x', \{r_j : 1 \leqslant j \leqslant k\})$, $grant\_rights(x, x', r, \{(y_j, r_j) : 1 \leqslant j \leqslant k\})$, $create\_object(x, x', r, y, yi, name, z)$, $create\_container(x, x', r, y, yi, ccri, t, name, z)$, $create\_hard\_link(x, x', y, name, z)$, $rename\_entity(x, x', y, name, z)$, $set\_container\_attr(x, x', y, ccri, t)$, $create\_first\_session(x, x', u, r, y, z, zi)$, $create\_session(x, x', r, y, z, zi)$, $access\_own(x, x', y)$, $access\_read(x, x', y)$, $access\_write(x, x', y)$, $control(x, y, z)$, $know(x, y)$, $take\_access\_own(x, y, z)$, $flow\_memory\_access(x, y, \alpha_a)$, $flow\_time\_access(x, y)$, $flow(x, y, y', z)$, $find(x, y, z)$, $post(x, y, z)$, $pass(x, y, z)$ и $take\_flow(x, y)$. Немонотонными правилами преобразования состояний будут являться де-юре правила $remove\_roles(x, x', \{r_j : 1 \leqslant j \leqslant k\})$, $remove\_rights(x, x', r, \{(y_j, r_j) : 1 \leqslant j \leqslant k\})$, $delete\_entity(x, x', y, z)$, $delete\_hard\_link(x, x', y, z)$, $delete\_session(x, x', z)$, $delete\_access(x, x', y, \alpha_a)$. Де-факто правило $de\_facto\_op(x, op(y, y', \ldots))$ по определению будем считать монотонным или немонотонным в зависимости от монотонности или немонотонности де-юре правила $op(y, y', \ldots)$.

Наличие в РОСЛ ДП-модели механизма ограничений в общем случае может требовать использования монотонных и немонотонных правил при передаче прав доступа ролей или возникновении информационных потоков. Например, пусть в некоторой

системе задано ограничение на значения множеств текущих ролей субъект-сессий, заключающееся в том, что некоторой ролью может одновременно обладать только одна субъект-сессия. Тогда без использования немонотонного правила вида $remove\_roles$ одной субъект-сессией может оказаться невозможным получение данной роли другой субъект-сессией.

Заметим, что в условиях применения правил преобразования состояний, приведенных в табл. 1 и 2, за исключением, возможно, ограничений из множеств $Constraint_P$ и $Constraint_S$, не требуется проверка отсутствия каких-либо элементов (например, сущностей, ролей, прав доступа ролей, информационных потоков), задающих исходное состояние G. По предположению 6 на траекториях системы не изменяются значения функций, задающих уровни целостности сущностей или субъект-сессий, существовавших в предшествующих состояниях системы. Таким образом, в рамках РОСЛ ДП-модели только ограничения обладают свойством, обуславливающим в некоторых случаях необходимость применения немонотонных правил для выполнения в дальнейшем условий применения других правил преобразования состояний. Значит, поиск способов обоснования возможности использования только монотонных правил преобразования состояний при анализе условий передачи прав доступа или возникновения информационных потоков целесообразно осуществлять не в общем случае, а для некоторых заданных для конкретных систем множеств ограничений. Дадим определение и обоснуем утверждение.

Определение 2. Ограничение инвариантно относительно немонотонных правил преобразования состояний в системе $\Sigma(G^*, OP)$, если при условии, что в системе задано только данное ограничение, для любых состояния системы G_0, немонотонного правила преобразования состояний op_1 и правил преобразования состояний op_2, …, op_N, где $N > 1$, справедливо следующее: если $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_{N-1}} G_{N-1}$, $G_0 \vdash_{op_2} G'_2 \vdash_{op_3} \ldots \vdash_{op_{N-1}} G'_{N-1}$ и в состоянии G_{N-1} выполнены ограничения, заданные в условиях применения правила op_N, то эти ограничения выполнены в состоянии G'_{N-1}. Ограничения, заданные в системе $\Sigma(G^*, OP)$, по определению инвариантны относительно немонотонных правил преобразования состояний, когда каждое из ограничений в отдельности инвариантно относительно этих правил.

Так как в правилах преобразования состояний, заданных в рамках РОСЛ ДП-модели, не используются ограничения на значения множеств авторизованных ролей учетных записей пользователей, то по определению 2 в любой системе все ограничения из множества $Constraint_U$ инвариантны относительно немонотонных правил преобразования состояний.

Утверждение 2. Пусть G_0 — начальное состояние системы $\Sigma(G^*, OP, G_0)$, в котором все ограничения инвариантны относительно немонотонных правил преобразования состояний, и функции $(i_u, i_e, i_r, i_s)_0$ удовлетворяют условиям предположения 7. Пусть также существуют состояния системы G_1, …, $G_N = (PA_N, user_N, roles_N, A_N, F_N, H_{E_N})$ и правила преобразования состояний op_1, …, op_N, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$, где $N \geqslant 0$. Тогда существуют состояния G'_1, …, $G'_M = (PA'_M, user'_M, roles'_M, A'_M, F'_M, H'_{E_M})$, где $M \geqslant 0$, и монотонные правила преобразования состояний op'_1, …, op'_M, такие, что $G_0 \vdash_{op'_1} G'_1 \vdash_{op'_2} \ldots \vdash_{op'_M} G'_M$ и выполняются следующие условия:

1. Верно включение $S_N \subset S'_M$ и для каждой субъект-сессии $s \in S_N$ выполняются условия $user_N(s) = user'_M(s)$, $roles_N(s) \subset roles'_M(s)$, $de\_facto\_own_N(s) \subset$ $\subset de\_facto\_own'_M(s)$.

2. Верно включение $E_N \subset E'_M$, для каждой сущности $e \in E_N \setminus S_N$, не являющейся субъектом, выполняется условие $H_{E_N}(e) \subset H'_{E_M}(e)$, для любых сущностей $e, e' \in E_N$ если в состоянии G_N выполняется условие $e < e'$, то данное условие выполняется в состоянии G'_M.

3. Для каждой роли $r \in R$ выполняется условие $PA_N(r) \subset PA'_M(r)$.

4. Верно включение $A_N \subset A'_M$.

5. Верно включение $F_N \subset F'_M$.

6. Функции $(i_u, i_e, i_r, i_s)'_M$ удовлетворяют условиям предположения 7.

Доказательство. Пусть существуют состояния G_0, G_1, …, G_N системы $\Sigma(G^*, OP, G_0)$ и правила преобразования состояний op_1, …, op_N, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$, где $N \geqslant 0$. Докажем утверждение индукцией по длине N последовательности состояний.

Пусть $N = 0$. Тогда положим $M = 0$, $G'_0 = G_0$, и для состояний G_0 и G'_0 условия 1–6 утверждения выполнены.

Пусть $N > 0$ и условия утверждения выполнены для всех последовательностей состояний длины $0 \leqslant L < N$. Докажем, что условия 1–5 утверждения выполнены для последовательностей состояний длины N.

По предположению индукции для последовательности состояний $G_0, G_1, \ldots, G_{N-1}$ существует последовательность состояний $G_0, G'_1, \ldots, G'_K = (PA'_K, user'_K, roles'_K, A'_K, F'_K, H'_{E_K})$, где $K \geqslant 0$, и для состояний G_{N-1} и G'_K выполнены условия 1–5 утверждения.

Рассмотрим правило преобразования состояний op_N. Если условия его применения не выполняются в состоянии G_{N-1}, то справедливо равенство $G_{N-1} = G_N$. Положим $M = K$, и для состояний G_N и G'_M выполнены условия 1–5 утверждения. Пусть условия применения правила op_N выполняются в состоянии G_{N-1}, тогда возможны два случая.

Первый случай: правило преобразования состояний op_N является немонотонным.

Если в $op_N(x, \ldots)$ субъект-сессия x принадлежит $LF_S \cap S_{N-1}$, то положим $M = K$, и условия 1–5 утверждения выполняются для состояний G_N и G'_K.

Пусть $x \in (N_S \cup NF_S) \cap S_{N-1}$.

Пусть $op_N = remove\_roles(x, x', \{r_j : 1 \leqslant j \leqslant l\})$. Тогда в результате применения правила возникают информационные потоки по времени из множества $STime = \{(x, s, write_t) : s \in (N_S \cup NF_S) \cap S_{N-1},\ x \neq s$ и $x \in de\_facto\_own_{N-1}(s)\}$. Так как для состояний G_{N-1} и G'_K выполнены условия 1–5 утверждения, то для $s \in S_{N-1} \subset S'_K$ справедливо $de\_facto\_own_{N-1}(s) \subset de\_facto\_own'_K(s)$. Положим $M = K + |STime|$ и последовательно для каждой субъект-сессии $s_i \in (N_S \cup NF_S) \cap S_{N-1}$, такой, что $x \neq s_i$ и $x \in de\_facto\_own_{N-1}(s_i)$, положим $op'_i = flow\_time\_access(s_i, x)$, где $K < i \leqslant M$. Пусть состояние G'_M получено из состояния G'_K в результате применения последовательности правил op'_{K+1}, …, op'_M: $G'_K \vdash_{op'_{K+1}} \ldots \vdash_{op'_M} G'_M$. Таким образом, для состояний G_N и G'_M выполнены условия 1–5 утверждения.

Пусть $op_N = remove\_rights(x, x', r, \{(y_j, r_j) : 1 \leqslant j \leqslant k\})$. Тогда в результате применения правила возникают информационные потоки по времени из множества $STime = \{(x, s, write_t) : s \in (N_S \cup NF_S) \cap S_{N-1},\ x \neq s,\ r \in de\_facto\_roles_{N-1}(s)$ и существует j, такое, что $1 \leqslant j \leqslant k$ и $(y_j, \alpha_a) \in de\_facto\_accesses_{N-1}(s)$, где $\alpha_a \in R_a\}$. Положим $Z = |STime|$. Так как для состояний G_{N-1} и G'_K выполнены условия 1–5 утверждения, то выполняются условия $(x, y_j, own_a) \in A'_K$ и для каждого s_i, к которому создается информационный поток из множества $STime$, выполняются условия $(y_j, \alpha_a) \in de\_facto\_accesses_{N-1}(s_i) \subset de\_facto\_accesses'_K(s_i)$, где $1 \leqslant i \leqslant Z$. Положим $op'_{K+i} = flow(x, y_j, y_j, s_i)$, где $1 \leqslant i \leqslant Z$, и $M = K + Z$. Пусть состояние G'_M

получено из состояния G'_K в результате применения последовательности правил op'_{K+1}, $\ldots$, op'_M: $G'_K \vdash_{op'_{K+1}} \ldots \vdash_{op'_M} G'_M$. Таким образом, для состояний G_N и G'_M выполнены условия 1–5 утверждения.

Пусть $op_N = delete\_entity(x, x', y, z)$. Тогда в результате применения правила возникают информационные потоки по времени, в том числе из множества $STime = \{(x, s, write_t) : s \in (N_S \cup NF_S) \cap S_{N-1}, x \neq s$ и $(y, \alpha_a) \in de\_facto\_accesses_{N-1}(s)$, где $\alpha_a \in R_a\}$. Положим $Z = |STime|$. Так как для состояний G_{N-1} и G'_K выполнены условия 1–5 утверждения, то выполняются условия $(x, z, write_a) \in A'_K$, и для каждого s_i, к которому создается информационный поток из множества $STime$, выполняются условия $(y, \alpha_a) \in de\_facto\_accesses_{N-1}(s_i) \subset de\_facto\_accesses'_K(s_i)$, где $1 \leqslant i \leqslant Z$. Положим $op'_{K+i} = flow(x, y, y, s_i)$, где $1 \leqslant i \leqslant Z$, $op'_{K+Z+1} = flow\_time\_access(x, y)$, и $M = K + Z + 1$. Пусть состояние G'_M получено из состояния G'_K в результате применения последовательности правил op'_{K+1}, $\ldots$, op'_M: $G'_K \vdash_{op'_{K+1}} \ldots \vdash_{op'_M} G'_M$. Таким образом, для состояний G_N и G'_M выполнены условия 1–5 утверждения.

Пусть $op_N = delete\_hard\_link(x, x', y, z)$. Так как для состояний G_{N-1} и G'_K выполнены условия 1–5 утверждения, то выполняются условия $(x, z, write_a) \in A'_K$, $i_e(y) \leqslant i_r(r) \leqslant i_s(w)$, $i_e(y) \leqslant i_e(z) \leqslant i_s(w)$, и если $i_e(z) = i\_high$, то $(x', i\_entity, write_a) \in A'_K$. Следовательно, в состоянии G'_K выполнены условия применения правила $create\_hard\_link(x, x', y, name, z)$, где $name \in NAME \setminus \{«»\}$ — некоторое допустимое имя сущности. Тогда положим $M = K + 1$, $op'_M = create\_hard\_link(x, x', y, name, z)$, и пусть состояние G'_M получено из состояния G'_K применением к нему правила op'_M: $G'_K \vdash_{op'_M} G'_M$. Таким образом, для состояний G_N и G'_M выполнены условия 1–5 утверждения.

Пусть $op_N = delete\_session(x, x', z)$. Тогда в результате применения правила возникают информационные потоки по времени, в том числе из множества $STime = \{(x, s, write_t) : s \in (N_S \cup NF_S) \cap S_{N-1}, x \neq s$ и $z \in de\_facto\_own_{N-1}(s)\}$. Положим $Z = |STime|$. Так как для состояний G_{N-1} и G'_K выполнены условия 1–5 утверждения, то выполняются условия $(x, z, write_a) \in A'_K$, и для каждого s_i, к которому создается информационный поток из множества $STime$, выполняются условия $z \in de\_facto\_own_{N-1}(s_i) \subset de\_facto\_own'_K(s_i)$, где $1 \leqslant i \leqslant Z$. Положим $op'_{K+i} = flow(x, z, z, s_i)$, где $1 \leqslant i \leqslant Z$, $op'_{K+Z+1} = flow\_time\_access(x, z)$ и $M = K + Z + 1$. Пусть состояние G'_M получено из состояния G'_K в результате применения последовательности правил op'_{K+1}, $\ldots$, op'_M: $G'_K \vdash_{op'_{K+1}} \ldots \vdash_{op'_M} G'_M$. Таким образом, для состояний G_N и G'_M выполнены условия 1–5 утверждения.

Пусть $op_N = delete\_access(x, x', y, \alpha_a)$. Так как для состояний G_{N-1} и G'_K выполнены условия 1–5 утверждения, то выполняется условие $(x, y, \alpha_a) \in A'_K$. Положим $M = K + 1$, $op'_M = flow\_time\_access(x, y)$, и пусть состояние G'_M получено из состояния G'_K применением к нему правила op'_M: $G'_K \vdash_{op'_M} G'_M$. Таким образом, для состояний G_N и G'_M выполнены условия 1–5 утверждения.

Пусть $op_N = de\_facto\_op(x, op(y, y', \ldots))$, где $op(y, y', \ldots)$ — немонотонное правило преобразования состояний. Так как для состояний G_{N-1} и G'_K выполнены условия 1–5 утверждения, то $de\_facto\_own_{N-1}(x) \subset de\_facto\_own'_K(x)$ и в состоянии G'_K выполняются условия применения правила $op(y, y', \ldots)$, являющегося одним из шести рассмотренных немонотонных правил. Следовательно, повторяя рассуждения для этих правил, получаем состояние G'_M, такое, что для состояний G_N и G'_M выполнены условия 1–5 утверждения.

Второй случай: правило преобразования состояний op_N является монотонным. Положим $M = K + 1$, $op'_M = op_N$, таким образом, монотонные правила исходной после-

довательности $op_1, \ldots, op_N$ без изменений добавляются в последовательность $op'_1, \ldots, op'_M$. Так как для состояний G_{N-1} и G'_K выполнены условия 1–5 утверждения и по предположению 6 на траекториях системы не изменяются значения функций, задающих уровни целостности сущностей или субъект-сессий, существовавших в предшествующих состояниях системы, то в состоянии G'_K выполняются все условия применения правила op_N, за исключением, возможно, ограничений, заданных в op_N. Данные ограничения выполнены в состоянии G_{N-1}, предположим, что они не выполнены в состоянии G'_K.

Если $N = 1$, то $G'_K = G_{N-1} = G_0$, противоречие. Следовательно, выполняется неравенство $N > 1$. Будем считать, что из последовательности $op_1, \ldots, op_{N-1}$ удалены все правила, условия применения которых не выполняются. Если последовательность $op_1, \ldots, op_{N-1}$ состоит только из монотонных правил, то справедливо $K = N-1$, $op_i = op'_i$, где $1 \leqslant i \leqslant N-1$, и $G'_K = G_{N-1}$. Противоречие. Следовательно, в последовательности $op_1, \ldots, op_{N-1}$ есть немонотонные правила. По определению 2, данные немонотонные правила, начиная с конца последовательности $op_1, \ldots, op_{N-1}$, могут быть удалены из нее, и на их место могут быть вставлены соответствующие монотонные правила вида $create\_hard\_link(x, x', y, name, z)$, $flow\_time\_access(x, y)$, $flow(x, y, y', z)$, использованные в доказательстве для первого случая. Эти правила добавляют все информационные потоки, которые создавались соответствующими немонотонными правилами, и не влияют на выполнение ограничений (так как не изменяют значения функций PA и $roles$). В результате будет получена последовательность монотонных правил $op'_1, \ldots, op'_K$, и по определению 2 в состоянии G'_K должно выполняться ограничение, заданное в правиле op_N. Противоречие. Значит, данное ограничение выполнено в состоянии G'_K.

Таким образом, пусть состояние G'_M получено из состояния G'_K с использованием монотонного правила op_N: $G'_K \vdash_{op_N} G'_M$. В соответствии с заданными в табл. 1 и 2 результатами применения правил преобразования состояний для состояний G_N и G'_M выполнены условия 1–5 утверждения.

Выполнение условия 6 утверждения во всех рассмотренных случаях следует из утверждения 1.

Следовательно, условия 1–6 утверждения выполнены для последовательностей состояний длины N, и шаг индукции доказан. ■

Таким образом, в рамках РОСЛ ДП-модели возможно задание ограничений, позволяющих при дальнейшем анализе условий передачи прав доступа ролей, реализации информационных потоков по памяти или по времени использовать только монотонные правила преобразования состояний.

Утверждение 3. Пусть G_0 — начальное состояние системы $\Sigma(G^*, OP, G_0)$, в которой все ограничения инвариантны относительно немонотонных правил преобразования состояний, и $op \in OP$ — правило преобразования состояний. Если в состоянии G_0 выполнены ограничения, заданные в условиях применения правила op, то эти ограничения выполнены в состоянии G_N любой траектории системы $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$, где $op_1, op_2, \ldots, op_N$ — монотонные правила преобразования состояний и $N \geqslant 1$.

Доказательство. Предположим противное. Пусть существуют монотонные правила преобразования состояний $op_1, op_2, \ldots, op_N$, где $N \geqslant 1$, такие, что $G_0 \vdash_{op_1} G_1 \vdash_{op_2} \ldots \vdash_{op_N} G_N$ — траектория системы, и в состоянии G_N не выполнены ограничения, заданные в условиях применения правила op. Выберем N минимальной длины, тогда в состоянии G_{N-1} выполнены ограничения, заданные в условиях применения правила op. Значит, без ограничения общности можно считать, что существует монотонное

правило преобразования состояний op_1, такое, что $G_0 \vdash_{op_1} G_1$, в состоянии G_1 не выполнены ограничения, заданные в условиях применения правила op, при этом условия применения правила op_1 выполнены в состоянии G_0. Кроме того, по предположению 6 в состоянии G_0 функции UA_0, PA_0 и $roles_0$ удовлетворяют соответствующим ограничениям $Constraint_U$, $Constraint_P$ и $Constraint_S$.

Правило op не может являться правилом вида $create\_hard\_link(x, x', y, name, z)$, $rename\_entity(x, x', y, name, z)$, $set\_container\_attr(x, x', y, ccri, t)$, $access\_own(x, x', y)$, $access\_read(x, x', y)$, $access\_write(x, x', y)$, $control(x, y, z)$, $know(x, y)$, $take\_access\_own(x, y, z)$, $flow\_memory\_access(x, y, \alpha_a)$, $flow\_time\_access(x, y)$, $flow(x, y, y', z)$, $find(x, y, z)$, $post(x, y, z)$, $pass(x, y, z)$ и $take\_flow(x, y)$, а также правилом вида $de\_facto\_op(x, op'_1(y, y', \ldots))$, где $op'_1(y, y', \ldots)$ — де-юре правило одного из перечисленных видов. Эти правила не изменяют значения функций UA, PA и $roles$, поэтому после их применения выполнявшиеся в состоянии G_0 ограничения, заданные в условиях применения правила op, должны выполняться в состоянии G_1. Следовательно, возможны семь случаев.

Первый случай: $op_1 = take\_roles(x, x', \{r_j : 1 \leqslant j \leqslant k\})$. Положим $op_2 = remove\_roles(x, x', \{r_j : 1 \leqslant j \leqslant k\})$.

Второй случай: $op_1 = grant\_rights(x, x', r, \{(y_j, r_j) : 1 \leqslant j \leqslant k\})$. Положим $op_2 = remove\_rights(x, x', r, \{(y_j, r_j) : 1 \leqslant j \leqslant k\})$.

Третий и четвертый случаи: $op_1 = create\_object(x, x', r, y, yi, name, z)$ или $op_1 = create\_container(x, x', r, y, yi, ccri, t, name, z)$. Положим $op_2 = delete\_entity(x, x', y, z)$.

Пятый и шестой случаи: $op_1 = create\_first\_session(x, x', u, r, y, z, zi)$ или $op_1 = create\_session(x, x', r, y, z, zi)$. Положим $op_2 = delete\_session(x, x', z)$.

Седьмой случай: $op_1 = de\_facto\_op(x, op'_1(y, y', \ldots))$, где $op'_1(y, y', \ldots)$ — де-юре правило одного из рассмотренных шести видов. Положим $op_2 = de\_facto\_op(x, op'_2(y, y', \ldots))$, где $op'_2(y, y', \ldots)$— соответствующее для каждого из рассмотренных шести случаев немонотонное де-юре правило.

Рассмотрим траекторию $G_0 \vdash_{op_1} G_1 \vdash_{op_2} G_2$. Во всех семи случаях выполнены условия применения немонотонного правила op_2, в том числе: так как $UA_0 = UA_2$, $PA_0 = PA_2$, $roles_0 = roles_2$, то выполнены ограничения, заданные в условиях применения правила op_2. Следовательно, в состоянии G_2 выполнены ограничения, заданные в условиях применения правила op. Следовательно, по определению 2 эти ограничения выполнены в состоянии G_1. Противоречие.

Утверждение доказано. ■

Таким образом, если в системе заданы ограничения, инвариантные относительно немонотонных правил преобразования состояний, и в ее начальном состоянии выполнены ограничения, заданные в условиях применения некоторого монотонного правила, то эти ограничения будут выполнены в любом состоянии на любой траектории системы, полученной из начального состояния в результате применения монотонных правил преобразования состояний.

6. Способ задания индивидуальных ролей учетных записей пользователей

При реализации в существующих ОС (использующих, как правило, дискреционное управление доступом) ролевого управления доступом, возможно, будет необходимо задание для каждой учетной записи пользователя индивидуальных прав доступа к сущностям. Например, эта необходимость возникает в следующих случаях:

— когда имеются сущности-файлы, принадлежащие только субъект-сессиям, функционирующим от имени учетной записи некоторого пользователя;
— когда субъект-сессия, создавая новую субъект-сессию, должна получить к ней право доступа владения.

Для обеспечения такой возможности в РОСЛ ДП-модели целесообразно для каждой учетной записи пользователя включить во множество ее авторизованных ролей принадлежащие только ей пары (для каждого уровня целостности, не превосходящего уровень целостности учетной записи пользователя) ролей: индивидуальную роль и индивидуальную административную роль, задав соответствующие параметры ограничений. Таким образом, сделаем предположение, дополняющее предположение 1.

Предположение 12. Для каждой учетной записи пользователя $u \in U$, каждого уровня целостности $i \leqslant i_u(u)$ существуют роль $u\_role\_i \in R$ и административная роль $u\_admin\_role\_i \in AR$, которые имеют следующие свойства:

— обладают уровнем целостности i;
— входят во множества авторизованных ролей и авторизованных административных ролей соответственно только учетной записи пользователя u;
— множества параметрически ассоциированных с ними сущностей совпадают с множеством сущностей, параметрически ассоциированных с учетной записью пользователя u.

С использованием административной роли $u\_admin\_role\_i$ разрешено включать или удалять права доступа во множество прав доступа роли u_role_i. При этом не накладываются следующие ограничения:

— из множества $Constraint_U$, требующие наличия во множестве авторизованных ролей учетной записи пользователя u ролей u_role_i и $u\_admin\_role\_i$;
— из множества $Constraint_P$, разрешающие получение любой ролью u_role_i права доступа владения к сущности при ее создании субъект-сессией, функционирующей от имени учетной записи пользователя u, или получение любой ролью u_role_i любого права доступа к сущности при наличии у роли u_role_i права доступа владения к этой сущности;
— из множества $Constraint_S$, разрешающие обладание субъект-сессиями, функционирующими от имени учетной записи пользователя u, текущими ролями u_role_i и $u\_admin\_role\_i$.

Таким образом, для каждой учетной записи пользователя $u \in U$ выполняются условия:

— $u\_role\_i \in UA(u)$, $u\_admin\_role\_i \in AUA(u)$;
— $i_r(u\_role\_i) = i_r(u\_admin\_role\_i) = i$;
— $]u\_role\_i[=]u\_admin\_role\_i[=]u[$;
— $u\_role\_i \in can\_manage\_rights(u\_admin\_role\_i)$.

В результате предположение 12 позволяет задать функции авторизованных ролей учетных записей пользователей, текущих ролей субъект-сессий и прав доступа ролей с учетом одной из особенностей реализации управления доступом в ОС.

Заключение

Основным направлением развития РОСЛ ДП-модели стала ее поэтапная адаптация к условиям функционирования реальных ОС семейства *AltLinux*. При этом в первую очередь элементы модели, правила преобразования состояний были переопределены с целью обеспечения четкого разделения де-юре и де-факто условий функционирования системы. Кроме того, включение в модель механизма ограничений потребова-

ло описания особого их вида — ограничений, инвариантных относительно немонотонных правил преобразования состояний, позволяющих при анализе условий передачи прав доступа ролей, получения доступов и возникновения информационных потоков по памяти или по времени использовать только монотонные правила. Таким образом, были созданы предпосылки для формулирования и обоснования в рамках РОСЛ ДП-модели условий нарушения безопасности системы и практической проверки их корректности на макете реальной ОС.

ЛИТЕРАТУРА

1. *Девянин П. Н.* Модели безопасности компьютерных систем. Управление доступом и информационными потоками. Учеб. пособие для вузов. М.: Горячая линия-Телеком, 2011. 320 с.
2. *Девянин П. Н., Захаренков П. С.* Способ реализации информационного потока по времени в операционных системах с мандатным управлением доступом через clipboard // Методы и технические средства обеспечения безопасности информации: материалы Юбилейной 20-й науч.-техн. конф. 27 июня – 01 июля 2011 г. СПб.: Изд-во Политехн. ун-та, 2011. С. 76–77.
3. *Колегов Д. Н.* ДП-модель компьютерной системы с функционально и параметрически ассоциированными с субъектами сущностями // Вестник Сибирского государственного аэрокосмического университета им. акад. М. Ф. Решетнева. 2009. Вып. 1(22). Ч. 1. С. 49–54.
4. *Девянин П. Н.* Правила преобразования состояний базовой ролевой ДП-модели управления доступом и информационными потоками в операционных системах // Прикладная дискретная математика. 2011. № 1(11). С. 78–95.
5. *Девянин П. Н.* Моделирование ролевого управления доступом в операционных системах семейства Linux // Проблемы информационной безопасности. Компьютерные системы. 2011. № 1. С. 24–43.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

DOI 10.17223/20710410/15/7

УДК 004.432:004.435

ЯЗЫК ПРОГРАММИРОВАНИЯ AspectTalk

Д. А. Стефанцов

Национальный исследовательский Томский государственный университет, г. Томск, Россия

E-mail: d.a.stefantsov@isc.tsu.ru

Описывается язык объектно-ориентированного (ООП) и аспектно-ориентированного (АОП) программирования AspectTalk, состоящий из базового языка, метаязыка, библиотек ООП и АОП. Он реализуется с помощью метапрограммирования, протоколов метаобъектов и механизма примесей. Приводится сравнение AspectTalk с языками программирования, имеющими похожие возможности.

Ключевые слова: *язык программирования, метапрограммирование, Smalltalk, ООП, АОП, метаобъектный протокол.*

Введение

Язык AspectTalk разработан для проведения экспериментальных исследований в области защиты программных систем обработки информации путём их интеграции с политиками безопасности средствами аспектно-ориентированного программирования [1 – 3].

Будучи языком метапрограммирования, язык АОП может быть построен на базе любого языка программирования. В качестве базового языка в AspectTalk используется собственный диалект языка объектно-ориентированного программирования Smalltalk, поскольку, во-первых, в настоящий момент ООП является наиболее распространённой парадигмой программирования, а во-вторых, язык Smalltalk имеет самый маленький набор правил записи среди существующих языков ООП.

В ООП вычисления производятся посылкой сообщений от одних объектов другим с просьбой произвести некоторые вычисления. При этом каждый объект обладает набором сообщений, на которые он может ответить. Этот набор называется *протоколом* объекта. Посылка сообщений — это способ вызова подпрограммы в объектной системе. Подробнее о системах ООП можно прочитать, например, в [4].

Свойства ООП и АОП в языке AspectTalk реализуются с помощью метаязыка, модули на котором изменяют семантику конструкций базового языка. Библиотека АОП реализована в два этапа: на первом — с помощью метаязыка в объектную систему вводятся метаобъекты, на втором — с помощью протоколов метаобъектов описывается механизм примесей.

Любые вычисления можно построить только на посылке сообщений, по примеру лямбда-исчисления [5], но в реальной вычислительной системе есть потребности взаимодействия с внешними устройствами — дисками, клавиатурой, монитором, мышью и т. д., а также в эффективных вычислениях и в управлении памятью. Для того чтобы

удовлетворить эти потребности, в объектную систему вводится дополнительная сущность — *виртуальная машина* (ВМ), выполняющая требуемый набор операций, называемых, по традиции, *примитивами*.

Для упрощения процесса трансляции и выполнения программы ВМ представляют, как правило, не набором подпрограмм, а в виде самостоятельной программы, принимающей на вход последовательности команд. Допустимые для ВМ команды образуют *язык ВМ*.

В данной работе описывается язык AspectTalk: подробно — его базовый язык и метаязык и кратко — библиотеки ООП и АОП. Последняя реализована с помощью протоколов метаобъектов. Сообщается также об используемых ВМ и о трансляторе с AspectTalk в промежуточный язык. Показано, как с помощью основных конструкций языка AspectTalk можно реализовать свойства распространённых объектно-ориентированных и аспектно-ориентированных систем.

1. Базовый язык

Объекты представляют собой наборы данных и допустимых операций над этими данными. Данные и операции, как правило, подбираются способом, подходящим для той или иной предметной области. Например, есть объекты-строки, представляющие собой последовательности символов. Операции над объектами-строками — это конкатенация, взятие подстроки, поиск подстроки в строке и т. д. Есть также объекты-числа с допустимыми операциями — сложением, умножением, делением, вычитанием и т. д. Данные объекта и операции над объектом называются соответственно *член-функциями* и *член-данными* объекта.

Объектная система строится рекурсивно: данные объекта являются ссылками на другие объекты. Для упрощения вводятся *элементарные данные* — объекты, реализация которых скрыта от пользователя объектной системы (программиста) с помощью ВМ. Элементарные данные могут не содержать ссылок на другие объекты.

Существует несколько способов задания множеств операций над объектами. Алгоритм поиска операции по её имени называется *диспетчеризацией*.

Член-данные и член-функции объекта хранятся в *словарях* — объектах, являющихся важными компонентами любой объектной системы. Словарь представляет собой функцию, отображающую множество одних объектов в множество других. Например, словарь может отображать числа на строки, содержащие записи этих чисел в восьмеричной системе счисления, и быть определённым только для чётных чисел. Он может отображать строки на числа и другие словари и быть определённым только для строк «мой дядя», «самых», «честных правил». Аргументы функции, представляемой словарём, называются *ключами*. Пара (ключ, значение) в словаре называется *элементом* этого словаря. Операции над словарями позволяют получить значение, соответствующее некоторому ключу, изменить значение, соответствующее некоторому ключу (т. е. расширить функцию), добавить новый ключ в словарь (т. е. расширить функцию, представляемую словарём) и т. д.

По цели использования все словари объектной системы можно разделить на пользовательские и системные. Первые пользователь объектной системы (т. е. программист) явно задаёт в программе, вторые определяются в объектной системе неявно и используются для её работы. Например, множество переменных, используемых в подпрограмме, задаётся словарём, отображающим имена переменных в их значения.

Здесь описываются базовая часть языка AspectTalk, а также конструкции языка в порядке их усложнения. Для каждой конструкции приводится соответствующая

часть грамматики в бэкусовой форме. Лексемы языка вводятся по мере необходимости с помощью регулярных выражений.

1.1. Литералы

Базовый язык AspectTalk включает стандартные типы объектов, необходимые для реализации большинства алгоритмов, — числа, строки, массивы и т. п. Для описания объектов стандартных типов в языке AspecTalk существуют специальные конструкции, называемые *литералами* и обладающие следующим свойством: два объекта, описанные одной конструкцией, равны. Как правило, строковое представление объекта стандартного типа, используемое, например, при выводе этого объекта на экран, совпадает с литералом, описывающим данный объект (листинг 1).

```
1 integer   ::= [+\-]?([0-9]+r)?[0-9a-zA-Z]+
2 string    ::= '([^']|'')+'
3 char      ::= \$.
4 symbol    ::= #[^\t\n\r [\]{}()"'#;.]+
5 <literal> ::= integer | string | char | symbol | <array>
6 <array>   ::= "#(" <literal> ")"
```

Листинг 1. Литералы языка AspectTalk

Числа

На данный момент язык AspecTalk поддерживает только литералы целых чисел. Соответствующее регулярное выражение представлено в строке 1 листинга 1. Целое число начинается с необязательного знака, после чего следует необязательное указание на систему счисления (по умолчанию используется десятичная), а затем записывается модуль числа.

Основание системы счисления может принимать значения от 2 до 36 включительно. Для записи цифр больше 9 используются символы английского алфавита в верхнем или нижнем регистре.

Примеры литералов, задающих целые числа: `+0`, `2r0`, `16r0`, `-3r0`. Все перечисленные литералы задают число 0.

Строки

Строковые литералы представляют собой последовательности знаков, заключенные в одинарные кавычки (см. строку 2 листинга 1). Для того чтобы включить знак одинарной кавычки в строковый литерал, его необходимо продублировать.

Примеры строковых литералов: `'мой дядя'`, `'самых'`, `'честных правил'`, `'Dot your i''s and cross your t''s'`.

Знаки

Знаки являются составными частями строк и задаются литералами, удовлетворяющими регулярному выражению в строке 3 листинга 1: литерал, задающий знак, начинается с символа «$», после которого следует требуемый знак.

Например, строка `'мой дядя'` содержит знаки `$м`, `$о`, `$й`, `$ `, `$д`, `$я`, `$д`, `$я`.

Символы

Символы используются в качестве ключей в системных словарях, т. е. в словарях член-функций, член-данных, аргументов, локальных переменных. Ключи в словарях член-функций называются *селекторами*, в прочих системных словарях — *идентификаторами*. Регулярное выражение для литералов, описывающих символы, представ-

лено в строке 4 листинга 1: описание символа начинается со знака «#», после которого записывается последовательность знаков. Примеры символов: `#size`, `#at:put:`.

Массивы

В языке AspectTalk, как и в языке Smalltalk, существуют литералы для массивов, элементы которых могут быть заданы литералами. Описание массива начинается со знаков «`#(`», после чего следуют 0 или более литералов, разделённых пробельными знаками, затем следует знак «`)`». В строках 5 и 6 листинга 1 приведена бэкусова форма, определяющая возможные литералы для описания массивов; внутри массивов знаки «#» могут быть опущены при описании символов и вложенных массивов.

Примеры массивов: `#()`, `#(1 #two (three) 'four')`.

Во многих языках программирования литералы отождествляются с константами. В языке AspectTalk этого отождествления следует избегать, поскольку массив, заданный литералом, не является константой и может быть изменён.

1.2. Переменные

В объектной системе объекту может быть сопоставлено имя. Соответствие имени объекту хранится в системных словарях. Элементы системных словарей, используемые для хранения данных, называются *переменными*. Будем говорить, что в переменной x *хранится объект* y или *значение* y, если объект y имеет имя x в объектной системе.

Имя переменной является идентификатором и представляет собой последовательность букв, цифр и знаков подчёркивания, начинающуюся не с цифры. Регулярное выражение, описывающее возможные идентификаторы, есть `[a-zA-Z_][0-9a-zA-Z_]*`. Примеры идентификаторов: `x`, `y`, `index`.

Переменные в языке AspectTalk хранят только ссылки на объекты, как, например, переменные в языках Smalltalk, Python и Java. Объекты хранятся в динамической памяти ВМ и удаляются механизмом сборки мусора в случае, если объект становится недоступным через текущие переменные программы. В отличие от языка Java, переменной не сопоставлена метка типа хранящегося объекта, поэтому в одной и той же переменной можно хранить объекты разных типов, как в языках Smalltalk и Python.

Присвоение значения переменной производится с помощью специальной операции, имеющей, по историческим причинам, три обозначения: «`:=`», «`<-`» и «`_`». В листинге 2 приведены примеры присвоений: в строке 1 переменной `x` присваивается значение `'Hello, world!'`, в строке 2 переменной `y` присваивается значение переменной `x`, в строке 3 производится множественное присвоение — в переменных `x` и `y` сохраняется ссылка на целое число `31337`.

```
1 x := 'Hello, world!'
2 y := x
3 y := x := 31337
```

Листинг 2. Примеры присвоений в языке AspectTalk

В языке AspectTalk есть псевдопеременные — переменные, значение которых всегда определено тем или иным образом и не может быть изменено командой присвоения. Примерами псевдопеременных являются `true`, `false` и `nil`, указывающие всегда на булевы константы 0, 1 и «пустой объект» соответственно.

1.3. Посылка сообщений

Объекты взаимодействуют при помощи посылки сообщений друг другу. Сообщение включает в себя имя и набор параметров. После получения сообщения объект

выполняет процедуру диспетчеризации — находит необходимый алгоритм обработки принятого сообщения. В процессе обработки сообщения объект может посылать сообщения другим объектам или самому себе. Заканчивается процесс обработки возвратом результата объекту, пославшему сообщение.

Результат процедуры диспетчеризации зависит от параметров трёх типов — объекта, принимающего сообщение, имени сообщения и параметров сообщения.

Процесс посылки сообщения можно рассматривать как способ вызова подпрограммы. Главные отличия: 1) позднее связывание сообщения с алгоритмом, выполняющим его обработку; 2) алгоритму обработки сообщения всегда передаётся дополнительный параметр — ссылка на объект, принявший сообщение, которая хранится в псевдопеременной `self`.

Унарные сообщения

Унарные сообщения не имеют параметров. Для посылки унарного сообщения необходимо после объекта-получателя записать селектор сообщения. В листинге 3 приведены примеры посылки унарных сообщений: в строке 1 массиву посылается сообщение с селектором `#size`, после чего результат — размер массива, число 3 — записывается в переменную `n`; в строке 2 числу `n` посылается сообщение с селектором `#factorial`, после чего результат — число 6 — записывается в переменную `x`.

```
1  n := #(one two three) size
2  x := n factorial
```

Листинг 3. Примеры посылки унарных сообщений

Далее для краткости сообщение с селектором `#x` будем называть сообщением `x`.

Бинарные сообщения

Бинарные сообщения — это сообщения с одним параметром, селектор которых записывается с помощью специальных знаков: «`+`», «`-`», «`*`», «`/`», «`%`», «`<`», «`<=`», «`=`», «`>=`», «`>`», «`~=`», «`==`», «`~~`», «`@`», «`,`». Получатель сообщения указывается перед селектором, а параметр — после. Бинарные сообщения введены в язык для записи математических операций в естественном виде. Примеры посылки бинарных сообщений приведены в листинге 4: в строке 1 объекту `1` посылается сообщение `+` с параметром `2`, а результату — объекту `3` — посылается сообщение `*` с параметром `3`, после чего результат — объект `9` — записывается в переменную `x`; в строке 2 объекту `9`, хранящемуся в переменной `x`, посылается сообщение `=` с параметром `9`, после чего результат — объект `true` — записывается в переменную `y`.

```
1  x := (1 + 2) * 3
2  y := x = 9
```

Листинг 4. Примеры посылки бинарных сообщений

Сообщения с ключевыми словами

Для посылки сообщений с ненулевым количеством параметров в языке AspectTalk реализованы сообщения с ключевыми словами. Ключевое слово записывается в виде идентификатора со знаком двоеточия на конце: `at:`, `and:`, `or:`. Селектор сообщения с ключевыми словами содержит столько ключевых слов, сколько сообщение содержит параметров: `#at:put:`, `#to:do:`, `#ifTrue:ifFalse:`. В записи посылки сообщений с ключевыми словами сначала указывается объект-получатель, а затем записывается последовательность ключевых слов, после каждого из которых следует параметр.

В листинге 5 приведены примеры посылки сообщений с ключевыми словами: в строке 1 массиву посылается сообщение `at:` с параметром `2`, результатом является второй элемент массива — символ `#two`; в строке 3 массиву из пяти элементов, хранящемуся в переменной `y`, посылается сообщение `at:put:` с параметрами `5` и `0`, в результате чего в пятый элемент массива записывается число `0`.

```
1 x := #(one two three) at: 2
2 y := #(0 0 0 0 1)
3 y at: 5 put: 0
```

Листинг 5. Примеры посылки сообщений с ключевыми словами

Данный синтаксис записи сообщений с ключевыми словами отличается от наиболее распространённого варианта записи вызова метода с несколькими параметрами в таких языках, как C++, Java, Python, где имя сообщения записывается слитно, после чего следуют параметры в скобках через запятую. Главной целью введения такой формы записи в язык AspectTalk, как и в язык Smalltalk и, например, в Objective-C, является повышение читаемости программы.

Порядок вычисления выражений

В записях посылок сообщений вместо любого объекта может, в свою очередь, стоять выражение, результатом обработки которого является объект. Записи такого вида будем называть *выражениями*. Вложенные выражения анализируются с учётом приоритетов посылки сообщений. Порядок посылки сообщений в выражении в языке AspectTalk позаимствован из языка Smalltalk и определяется следующими правилами:

1) приоритеты типов сообщений в порядке убывания:
   а) унарные;
   б) бинарные;
   в) с ключевыми словами;
2) выражения для объектов получателя сообщения и параметров сообщения вычисляются в произвольном порядке;
3) сообщения с одинаковым приоритетом посылаются в порядке слева направо;
4) порядок вычисления выражения можно изменить с помощью скобок `(` и `)`.

В листинге 6 иллюстрируется порядок вычисления выражений:

1) в строке 2 приведено выражение, иллюстрирующее приоритет операций: наименее приоритетной операцией является посылка сообщения `at:put:` — она будет выполнена последней, при этом получатель (объект `x`) и параметры (`1` и результат выражения `x size + 1`) вычисляются в произвольном порядке, возможно, зависящем от транслятора; вычисление выражений `x` и `1` тривиально; в выражении `x size + 1` посылка бинарного сообщения `+` — наименее приоритетная операция, она будет выполняться последней, при этом получатель и параметр (`x size` и `1`) вычисляются в произвольном порядке;
2) в строке 3 приведено вычисление математического выражения с помощью бинарных селекторов; поскольку у бинарных селекторов одинаковый приоритет, посылка соответствующих сообщений начинается слева направо, т. е. сначала будет послано сообщение `+` объекту `1` с параметром `2`, затем результату (объекту `3`) будет послано сообщение `*` с параметром `3`; в данном случае приоритет операций языка расходится с приоритетом операций в математических записях; эта особенность взята из языка Smalltalk, у которого язык AspectTalk заимствует большинство свойств;

3) в строке 4 иллюстрируется использование скобок для изменения порядка вычисления выражения: второй элемент массива `x` увеличивается на единицу.

```
1 x := #(1 2 3 4 5)
2 x at: 1 put: x size + 1
3 y := 1 + 2 * 3
4 x at: 2 put: (x at: 2) + 1
```

Листинг 6. Примеры, иллюстрирующие порядок вычисления выражений

Каскады сообщений

В программах на AspectTalk, так же как и в программах на Smalltalk, часто возникает необходимость послать несколько сообщений одному и тому же объекту. Для этого вводится специальная синтаксическая конструкция — *каскады сообщений*. Для описания каскада необходимо после описания объекта (литерала, переменной или сообщения с посылками сообщений) перечислить посылаемые сообщения с параметрами, каждое из которых необходимо предварить знаком «`;`». Результатом вычисления выражения с каскадом всегда является объект, принимающий сообщения каскада.

В листинге 7 представлен пример использования каскадов сообщений. В этом примере словарь используется для хранения сведений о человеке: в строке 1 результатом выражения `Dictionary new` является новый словарь, которому в строках 2–4 посылаются три сообщения `at:put:`, сохраняющие в нём данные о человеке. Результат вычисления каскада — новый словарь, в который добавлено три элемента, — помещается в переменную `person`.

```
1 person := Dictionary new;
2     at: 'name' put: 'Иван Иванов';
3     at: 'age' put: 42;
4     at: 'height' put: 178
```

Листинг 7. Пример использования каскадов сообщений

Следует отметить, что каскадные выражения отличаются в диалектах языка Smalltalk — Smalltalk-80 и Little Smalltalk. В языке AspectTalk используются каскадные выражения из последнего.

1.4. Блоки

В языке AspectTalk выражения, состоящие из литералов, переменных, операций присвоения значения, посылок сообщений и каскадов, могут выполняться последовательно. При этом каждое из предложений выполняется ради побочного эффекта — изменения значения переменной, вывода строки на экран и т. п. Предложения отделяются друг от друга знаком точки «`.`». Точка после последнего предложения не обязательна. Пример последовательности предложений показан в листинге 8. Всякая программа на AspectTalk является последовательностью предложений.

```
1 s := 'Мой дядя самых честных правил'.
2 w := $я.
3 r := 0.
```

Листинг 8. Последовательность предложений на языке AspectTalk

В языке AspectTalk, как и в языке Smalltalk, можно описать объекты, представляющие собой последовательности предложений и называемые *блоками*. Как и всякие

объекты, блоки обладают состоянием и могут принимать и обрабатывать сообщения. Одним из таких сообщений является сообщение, запускающее выполнение предложений, содержащихся в блоке. Таким образом, блоки представляют собой разновидность подпрограмм. Параметры и результат сообщения, запускающего выполнение предложений блока, являются соответственно входными и выходным параметрами подпрограммы. С другой стороны, блоки являются объектами и могут быть сохранены в переменной, переданы в качестве параметров в другой блок, возвращены в качестве результата вычисления другого блока, а также могут принимать сообщения.

Описание блока начинается со знака «[», после которого следует необязательное описание входных параметров блока: имя каждого параметра предваряется знаком «:», параметры отделяются пробельными символами, описание параметров заканчивается знаком «|». После необязательного описания параметров записываются предложения блока. Описание блока заканчивается знаком «]». Блоки могут быть вложенными, т. е. содержать описания других блоков.

Если блок не имеет параметров, сообщение, запускающее выполнение предложений блока, имеет селектор `#value`. В противном случае селектор состоит из ключевых слов `value:`, количество которых равно числу параметров блока. Результат последнего предложения считается результатом выполнения всего блока.

В листинге 9, строка 1, показано описание блока с одним параметром и одним предложением, возвращающим значение `true`, если строка-параметр имеет ненулевую длину, и `false` в противном случае. В строке 2 запускается выполнение блока и его результат записывается в переменную `x`.

```
1 notEmpty := [ :str | str size > 0 ].
2 x := notEmpty value: s.
3 lcm := [ :n :m | d := n gcd: m. (n * m) / d ]
```

Листинг 9. Блок с одним параметром

Предложения блока могут содержать локальные переменные. В строке 3 листинга 9 определён блок, вычисляющий наименьшее общее кратное двух чисел. В процессе вычисления используется локальная переменная `d`, хранящая наибольший общий делитель чисел. Всякая запись значения в переменную, не содержащуюся в текущих системных словарях, добавляет эту переменную в словарь локальных переменных. Переменные в AspectTalk могут вводиться по необходимости во время своего первого использования. Словарь локальных переменных очищается при каждом запуске блока.

Поиск переменных всегда начинается со словаря локальных переменных текущего блока, продолжается в словаре параметров блока, после чего поиск рекурсивно продолжается среди переменных объемлющего блока. Таким образом, поиск переменных производится в цепочке словарей, начинающейся со словаря локальных переменных и заканчивающейся словарём глобальных переменных. В случае, когда переменная была найдена в одном из объемлющих блоков, указатель на эту переменную сохраняется в текущем блоке-объекте и хранится в нём до тех пор, пока он не будет удалён сборщиком мусора. Благодаря этой особенности, блоки в AspectTalk являются *замыканиями* в терминах таких языков, как Scheme и Python. Замыкания также реализованы в одной из популярных библиотек языка C++ [6].

В листинге 10 приведён пример программы, демонстрирующей возможности замыканий. В блоке `gen` объявляется локальная переменная `n` и инициализируется нулём, после чего блок возвращает вложенный блок, увеличивающий `n` на 1 и возвращающий значение `n`. Поскольку словарь локальных переменных блока очищается при каждом

запуске блока, указатели на переменные `n` в блоках `c1` и `c2` будут указывать на разные области памяти. Следовательно, `c1` и `c2` будут изменять разные области памяти и вести независимый счёт. Комментарии в строках 4–6 описывают значение соответствующих переменных.

```
1 gen := [ n := 0. [ n := n + 1. n ] ].
2 c1 := gen value.
3 c2 := gen value.
4 x := c1 value. " x = 1 "
5 y := c1 value. " y = 2 "
6 z := c2 value. " z = 1 "
```

Листинг 10. Замыкания в языке AspectTalk

Замыкания являются мощным механизмом, позволяющим кратко описывать сложные алгоритмы и структуры данных. В частности, библиотека ООП, описываемая далее, реализована с помощью замыканий.

Условные переходы

В языке AspectTalk, как и в языке Smalltalk, нет специальных синтаксических конструкций для определения условных переходов. Для реализации ветвлений в программе используются блоки и булевы значения `true` и `false`. Последним могут быть посланы сообщения с селекторами `#ifTrue:`, `#ifFalse:`, `#ifTrue:ifFalse:`, `#ifFalse:ifTrue:`, принимающие блоки в качестве параметров и запускающие их выполнение в зависимости от значения, при этом результат сообщения равен результату запущенного блока или `nil`, если не был запущен ни один блок (см. таблицу).

Ветвления в языке AspectTalk

Селектор сообщения	`true`	`false`
`#ifTrue:`	Запускается блок-параметр	`nil`
`#ifFalse:`	`nil`	Запускается блок-параметр
`#ifTrue:ifFalse:`	Запускается 1-й блок-параметр	Запускается 2-й блок-параметр
`#ifFalse:ifTrue:`	Запускается 2-й блок-параметр	Запускается 1-й блок-параметр

В листинге 11 приведены примеры условного перехода. В строке 1 первый знак строки `s` сравнивается со знаком, хранящимся в переменной `w`; в случае совпадения переменная `r` увеличивается на 1. В строках 2–4 устанавливается взаимная простота чисел `n` и `m`. Если значение выражения `(n gcd: m) = 1` равно `true`, то переменной `x` присваивается значение `1`, а если `false`, то `0`.

```
1 (s at: 1) = w ifTrue: [ r := r + 1].
2 (n gcd: m) = 1
3     ifTrue: [ x := 1 ]
4     ifFalse: [ x := 0 ].
```

Листинг 11. Пример условного перехода в языке AspectTalk

Циклы

Как и в случае условных переходов, в языке AspectTalk нет специальных синтаксических конструкций для организации циклов — они реализуются с помощью блоков.

Самый простой в реализации вид циклов — циклы с условием. Цикл с условием реализуется посылкой сообщения `#whileTrue` блоку. Для обработки этого сообщения блок вычисляет сам себя (посылая сообщение `#value` объекту в псевдопеременной `self`), и

если результат — это объект `true`, то рекурсивно посылает себе сообщение `#whileTrue`. Рекурсия продолжается до тех пор, пока значение, возвращаемое в ответ на сообщение `#value`, равно `true`. Следовательно, цикл с условием можно описать следующим образом: тело цикла записывается в виде предложений некоторого блока без параметров; последнее предложение представляет собой условие продолжения цикла; описанному блоку посылается сообщение `#whileTrue`. Очевидно, реализуемый подобным образом цикл является циклом с постусловием.

Цикл с предусловием реализуется с помощью сообщения `#whileTrue:` — блок-приёмник сообщения вычисляет себя (`self value`), и если результат равен `true`, вычисляет блок-параметр, после чего рекурсивно посылает себе сообщение `#whileTrue:` с тем же параметром, который был получен из предыдущего сообщения. Условие продолжения цикла записывается в блоке-приёмнике, а тело цикла — в блоке-параметре.

Аналогичным образом обрабатываются сообщения `#whileFalse` и `#whileFalse:`; единственное отличие состоит в том, что проверяемое условие — это условие выхода из цикла, а не условие продолжения цикла.

В листинге 12 приведён пример программы на AspectTalk, подсчитывающей количество вхождений знака `$я` в строку `'Мой дядя самых честных правил`. Результат накапливается в переменной `r`. В данной программе используется сообщение `#ifTrue:`.

```
1 s := 'Мой дядя самых честных правил'.
2 w := $я.
3 r := 0.
4
5 [ s size > 0 ] whileTrue:
6     [ (s at: 1) = w
7           ifTrue: [ r := r + 1 ].
8       s := s allButFirst. ].
```

Листинг 12. Пример использования цикла с условием

Несмотря на то, что циклы в AspectTalk реализуются с помощью рекурсии, их выполнение не приводит к росту стека вызовов подпрограмм, так как в их реализации используется механизм хвостовой рекурсии, описанный в [7].

Очевидно, цикл со счётчиком можно реализовать с помощью цикла с условием, заведя переменную цикла, увеличивая её на каждой итерации и сравнивая с конечной границей. Данная дисциплина организации цикла для удобства реализована в виде сообщений `#to:do:` и `#to:by:do:`, посылаемых целым числам: получатель является начальной границей счётчика, параметр ключевого слова `to:` — конечной границей счётчика, параметр ключевого слова `do:` — телом цикла. В сообщении `#to:by:do:` есть также возможность указать шаг счётчика с помощью параметра `by:`. Необходимо отметить, что счётчик передаётся в блок — тело цикла при вычислении последнего.

В листинге 13 приведён тот же алгоритм, что и на листинге 12, но с использованием цикла со счётчиком вместо цикла с условием.

```
1 s := 'Мой дядя самых честных правил'.
2 w := $я.
3 r := 0.
4 1 to: s size do:
5     [ :i | (s at: i) = w ifTrue: [ r := r + 1 ] ].
```

Листинг 13. Пример использования цикла со счётчиком

В языке AspectTalk есть возможность организации циклов по элементам объектов-агрегатов, содержащих в себе множество равнозначных ссылок на другие объекты. Примерами объектов-агрегатов являются массивы, строки и словари. Для описания подобного цикла объекту-агрегату посылается сообщение `#do:`, параметром которого является блок — тело цикла. В процессе обработки сообщения объект-агрегат перебирает содержащиеся в нём ссылки на объекты и для каждого из них посылает сообщение `#value:` телу цикла, передавая в качестве параметра текущий элемент.

В листинге 14 представлен тот же алгоритм, что и в листингах 12 и 13, но реализованный с помощью цикла по строке, являющейся агрегатом объектов-знаков.

```
1 s := 'Мой дядя самых честных правил'.
2 w := $я.
3 r := 0.
4
5 s do: [ :c | c = w ifTrue: [ r := r + 1 ] ].
```

Листинг 14. Пример использования цикла по объектам-агрегатам

1.5. Примитивы

Примитивом называется операция обращения к виртуальной машине. Примитивы используются для выполнения простых действий, таких, как арифметические операции над целыми числами, вывод строки на экран и т. п. Некоторые операции, которые могут быть выполнены средствами языка AspectTalk, реализуются виртуальной машиной в целях оптимизации. К таким операциям относится, например, преобразование числа в строку.

В отличие от Smalltalk, примитивы в AspectTalk имеют явные параметры и возвращаемое значение. В Smalltalk параметры и возвращаемое значение примитивов берутся из текущего стека, что затрудняет понимание текста программы, а также делает язык зависимым от реализации ВМ.

Примитивы можно разбить на три группы:

1) примитивы общего назначения;
2) примитивы, используемые для ввода и вывода;
3) примитивы, относящиеся к тому или иному типу данных.

В листинге 15 проиллюстрированы примитивы этих трёх групп: в строке 1 указан примитив `newObject`, результатом выполнения которого является новый объект; примитив `write` принимает два параметра — имя потока вывода (`stdout`) и строку, выводит переданную строку на поток вывода и возвращает `nil` (см. строку 2); в строке 3 используется примитив `integerAddition`, принимающий на вход два числа и возвращающий их сумму.

```
1 <newObject>
2 <write stdout 'Hello, world!'>
3 <integerAddition x 1>
```

Листинг 15. Примеры примитивов в языке AspectTalk

1.6. Грамматика языка AspectTalk

В листинге 16 описаны все возможные лексемы языка в виде регулярных выражений (строки 1–11) и правила синтаксиса в бэкусовой форме (строки 13–29). Лексемы записаны в виде идентификаторов, нетерминалы грамматики — в виде идентификато-

ров, окружённых угловыми скобками «<» и «>». Простые (односимвольные и двухсимвольные) лексемы представлены в синтаксических правилах буквально и окружены знаками «"».

Выделение группы терминалов и нетерминалов знаками

1) «(» и «)?» означает, что эта группа может присутствовать и отсутствовать в выражении, задаваемом правилом;
2) «(» и «)*» означает повторение группы ноль или более раз;
3) «(» и «)+» означает повторение группы один или более раз.

Выше описана семантика всех приведённых в листинге 16 конструкций, за исключением команды возврата результата в строке 14, которая будет описана в п. 3.

```
 1 identifier   ::= [a-zA-Z_][0-9a-zA-Z_]*
 2 keyword      ::= [a-zA-Z_][0-9a-zA-Z_]*:
 3 binarysel    ::= +|-|*|/|%|<|<=|=|>=|>|~=|==|~~|@|,
 4 unarysel     ::= [a-zA-Z_][0-9a-zA-Z_]*
 5 integer      ::= [+\-]?([0-9]+r)?[0-9a-zA-Z]+
 6 symbol       ::= #[^\t\n\r [\]{}()"'#;.]+
 7 string       ::= '([^']|'')+'
 8 comment      ::= "([^"]|"")+"
 9 char         ::= \$.
10 delimiter    ::= [\t\r\n ]+
11 assign       ::= _|<-|:=
12
13 <statements> ::= <statement> ( "." <statement> )*
14 <statement>  ::= ( "^" )? <expression>
15 <expression> ::= identifier assign <expression>
16 <expression> ::= <cascaded>
17 <cascade>    ::= <simple> ( ";" <selectors> )*
18 <simple>     ::= <binary> ( keyword <binary> )*
19 <binary>     ::= <unary> ( binarysel <unary> )*
20 <unary>      ::= <primary> ( unarysel )*
21 <primary>    ::= <variable> | <literal> |
22                  <block> | <primitive> | "(" <cascade> ")"
23 <selectors>  ::= ( unarysel | binarysel <unary> |
24                    ( keyword <binary> )+ )+
25 <block> ::= "[" ( (":" identifier)+ "|")?  <statements> "]"
26 <literal>    ::= integer | symbol | string | char | <array>
27 <array>      ::= "#(" ( <literal> )* ")"
28 <primitive>  ::= "<" identifier ( <primary> )* ">"
29 <variable>   ::= identifier
```

Листинг 16. Грамматика языка AspectTalk

2. Метаязык

Метаязык используется для изменения семантики синтаксических конструкций языка. Инструкции метаязыка записываются в отдельных модулях и имеют собственные грамматические правила.

Содержание метаязыкового модуля в AspectTalk описывает синтаксические правила, с помощью которых транслятор определяет конструкции, семантику которых необходимо изменить. Правила определяются в виде, схожем с бэкусовой формой (ли-

стинг 17). В качестве элементов описания допускается использование нетерминалов и лексем, определённых в грамматике базового языка AspectTalk. Последним правилом в метаязыковом модуле должно быть правило вида «`<handler> ::= identifier`», в котором задаётся имя блока, вызываемого вместо всех конструкций, описанных с помощью правил модуля. Таким образом семантика этих конструкций заменяется на определённую в указанном блоке. Такой блок будем называть *обработчиком*.

```
 1 <rules> ::= <rule> <newline> <rules>
 2 <rule> ::= <item> "::=" <items> |
 3            "<handler>" "::=" identifier
 4 <items> ::= <ritem> <items> |
 5             <ritem> "|" <items> |
 6             "(" <items> ")?" |
 7             "(" <items> ")*" |
 8             "(" <items> ")+" |
 9             "(" <items> ")"
10 <ritem> ::= <item> | <item> "->" identifier
11 <item> ::= identifier | "<" identifier ">"
```

Листинг 17. Грамматика метаязыковых модулей на AspectTalk

В правилах метаязыковых модулей можно использовать конструкцию «`->`», позволяющую получить объект из контекста выполнения синтаксической конструкции (см. строку 10). После служебного слова «`->`» указывается идентификатор, который будет использован в словаре всех объектов из контекста выполнения синтаксической конструкции. Этот словарь будет передан в блок-обработчик при его вызове.

Если конструкции, описанные правилами метаязыкового модуля, имеют возвращаемое значение (т. е. если описывается выражение), то значение, возвращаемое блоком-обработчиком, возвращается вместо значения описанных конструкций.

Заметим, что метаязыковые конструкции языка Groovy [8] позволяют решать аналогичные задачи, однако вариант AspectTalk удобнее в использовании, так как позволяет декларативно объявить интересующие конструкции, не заставляя программиста работать с конкретными деревьями разбора, которые могут различаться в разных трансляторах.

2.1. Порядок выполнения метаязыковых модулей

Порядок выполнения языковых модулей задаётся либо в отдельном файле, либо в командной строке. Сначала загружаются метаязыковые модули в указанном порядке, потом — модули на базовом языке в указанном порядке.

3. Реализация ООП в AspectTalk

С помощью метаязыка в языке AspectTalk реализованы свойства, традиционные для объектно-ориентированных языков: инкапсуляция, наследование и полиморфизм. В отличие от своего предшественника, AspectTalk лишён декларативных конструкций, описывающих как член-данные и член-функции объектов, так и традиционную для объектно-ориентированных языков иерархию наследования. Все необходимые действия задаются конструкциями языка, описанными выше.

3.1. Инкапсуляция

С каждым объектом в языке AspectTalk связаны два словаря — член-данных и член-функций объекта. Для каждого объекта эти словари могут формироваться индивидуально. При этом говорят, что объект инкапсулирует член-данные.

В листинге 18 приведён пример объявления объекта добавлением к его словарю член-данных переменной с именем `word` и к словарю его член-функций элемента с именем `say`, который выводит на экран строку `'Hello, world!'` при выполнении.

```
1 a := Object new;
2     addVariable: #word;
3     addMethod: #say usingBlock:
4         [ word := 'Hello, world!'.
5           word writeln ].
```

Листинг 18. Объявление объекта

Теперь если послать объекту `a` сообщение `say`, то на экран выведется строка `'Hello, world!'`. В листинге 18 модификация словарей объекта производится с помощью заранее определённых для всех объектов сообщений `addVariable:` и `addMethod:usingBlock:`. Эти методы обращаются к соответствующим примитивам виртуальной машины, которая и производит модификацию словарей.

В описании грамматики языка в листинге 16 указана конструкция, не описанная до сих пор (см. строку 14) — команда возврата результата, начинающаяся со знака «`^`», за которым следует выражение. Эта команда возвращает значение указанного выражения в качестве результата обработки текущего сообщения. Если команда встретилась в блоке, не являющемся член-функцией некоторого объекта, то работа блока прерывается, управление передаётся блоку, запустившему данный, и производится проверка, является ли он член-функцией некоторого объекта, и т. д. до тех пор, пока очередной блок из стека вызова блоков не окажется член-функцией некоторого объекта. После этого значение указанного в команде выражения возвращается в качестве результата работы найденной член-функции.

Следует упомянуть об изменении порядка поиска переменных в случае ООП: словарь член-данных текущего объекта встраивается в цепочку словарей переменных сразу после словарей локальных переменных и параметров текущего блока.

3.2. Наследование

Для того чтобы избежать необходимости задания словарей для каждого объекта, используются специальные объекты, называемые *классами*. Класс — это объект, способный конструировать другие объекты и задавать для них часть словарей член-данных и член-функций. В листинге 18 использован класс `Object`, посылкой сообщения `new` которому конструируется новый объект. Именно в классе `Object` определены член-функции `addVariable:` и `addMethod:usingBlock:`, позволяющие модифицировать словари объекта.

В листинге 19 приведено определение класса на языке AspectTalk. Как и в случае с определением объекта, удобным средством описания оказывается механизм каскадов сообщений.

Определяется класс `Talkative`; все объекты, сконструированные с его помощью (далее такие объекты называются *экземплярами* соответствующего класса), будут понимать сообщение `say` и выводить строку `'Hello, world!'` в процессе их обработки. В строке 8 представлено объявление экземпляра класса `Talkative` — объекта `a`. Это делается посылкой сообщения `new` объекту-классу `Talkative` (так же, как в листинге 18 сообщение `new` посылается объекту-классу `Object`). При получении данного сообщения класс не только конструирует свой экземпляр, но и посылает ему сообщение

`init` перед возвратом его в вызвавший алгоритм. Поэтому метод с именем `init` может считаться аналогом конструкторов в языках C++ и Java.

```
1 Talkative := Class new;
2     addInstanceVariable: #word;
3     addInstanceMethod: #init usingBlock:
4         [ word := 'Hello, world!' ];
5     addInstanceMethod: #say usingBlock:
6         [ word writeln ].
7
8 a := Talkative new
```

Листинг 19. Определение класса на языке AspectTalk

Модификация словарей экземпляров класса производится с помощью сообщений `addInstanceVariable:` и `addInstanceMethod:usingBlock:`, которые создают соответственно член-данные и член-функции класса как объекта. В языках C++ и Java такие член-данные и член-функции называются статическими.

Для сокращения избыточности программ в языке AspectTalk используется механизм наследования, привычный для всех распространённых объектно-ориентированных языков программирования: член-данные и член-функции, объявленные в классе-родителе, наследуются дочерним классом и не требуют повторного объявления. В листинге 20 приведён пример описания унаследованного класса. Наследование происходит с помощью передачи параметра сообщению `new:`.

```
1 Memorizer := Class new: Talkative;
2     addInstanceMethod: #remember: usingBlock:
3         [ :what | word := what ].
4
5 a := Memorizer new
```

Листинг 20. Объявление класса-потомка

Экземпляры класса `Memorizer`, подобно экземплярам класса `Talkative`, выводят на экран содержимое член-данного `word`, однако позволяют модификацию этого член-данного с помощью посылки сообщения `remember:`.

Как и в языках Java и Smalltalk, все классы объединены в единую иерархию наследования с корнем — классом `Object`. Следует отметить, что, как и в языках-предшественниках Smalltalk-80 и Little Smalltalk, в языке AspectTalk нет множественного наследования, характерного для языков C++ и Python. Ещё одной особенностью AspectTalk по сравнению с языком C++ является тот факт, что все член-функции на этом языке являются «виртуальными» в терминологии языка C++. Это происходит потому, что все переменные в языке AspectTalk являются указателями, как и в языках Smalltalk-80, Little Smalltalk, Python, Java и многих других, в которых также все член-функции являются «виртуальными» в терминологии языка C++.

3.3. Полиморфизм

Тот факт, что все член-данные в языке AspectTalk являются «виртуальными», обусловливает свойство полиморфизма языка. Следует отметить также, что два объекта

совершенно разных классов могут быть переданы в один блок при условии, что оба они обрабатывают сообщения, посылаемые параметрам блока в его предложениях. Это свойство обусловливается отсутствием типизации переменных языка AspectTalk.

При необходимости установления принадлежности объекта классу используются сообщения `isKindOf:` и `isMemberOf:`, определённые для всякого объекта и принимающие класс в качестве параметра. В первом случае результат будет истинен, если переданный в качестве параметра класс встречается на пути от класса объекта-получателя до корня иерархии наследования. Во втором случае результат будет истинен только тогда, когда получатель является экземпляром класса, переданного в качестве параметра.

3.4. Стандартная библиотека классов

Стандартная библиотека классов языка AspectTalk содержит описание объекта `Metaclass`, метакласса `Class` и следующих стандартных классов: `Object`, `Block`, `Execution`, `Comparable`, `Integer`, `Symbol`, `Char`, `String`, `Array`, `UndefinedObject`, `Boolean`, `True`, `False`.

На рис. 1 изображена иерархия наследования классов стандартной библиотеки языка AspectTalk. Все прямые подклассы класса `Object` сгруппированы для краткости в один узел.

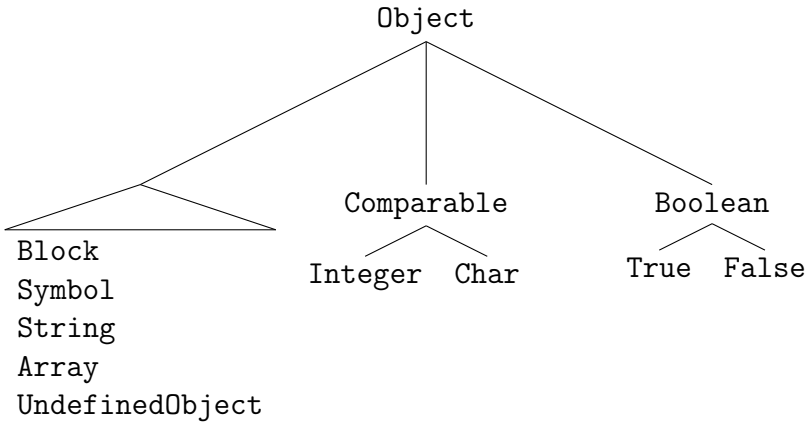


Рис. 1. Иерархия наследования классов стандартной библиотеки языка AspectTalk

4. Реализация АОП в AspectTalk

Аспектно-ориентированные свойства языка AspectTalk, как и его объектно-ориентированные свойства, описаны в библиотеке, реализованной с помощью метаязыка. Технология реализации АОП — протоколы метаобъектов [9]. Важным метаобъектом для реализации АОП является *метакласс*, который, во-первых, является классом класса, а во-вторых, модифицирует механизм передачи сообщений между объектами.

В листингах 19 и 20 уже использовался метакласс `Class`, посылкой сообщения `new` которому создаются объекты-классы. Это отражает первую особенность метаклассов: всякий класс в языке AspectTalk является экземпляром некоторого метакласса и принимает сообщения `addInstanceVariable:` и `addInstanceMethod:usingBlock:`, с помощью которых модифицируются словари экземпляров классов.

В листинге 21 отражена вторая особенность метаклассов — изменение механизма диспетчеризации сообщений. Если экземпляру класса, который, в свою очередь, является экземпляром метакласса `DialogueClass` либо подклассом экземпляра метакласса

`DialogueClass`, будет послано сообщение «`say`», то сначала будет исполнен блок, объявленный с помощью сообщения «`addEnvelope:usingBLock`», посланного метаклассу `DialogueClass`.

```
1 DialogueClass := Metaclass new;
2   addEnvelope: 'say' usingBlock: [ :sender :message :args |
3     '-- ' write.
4     ^ receiver call: message with: args.
5   ].
6 Talkative class: DialogueClass.
```

Листинг 21. Описание метакласса на языке AspectTalk

Этот блок имеет доступ ко всем член-данным объекта-получателя сообщения `say`; в качестве аргументов ему передаются объект-отправитель, селектор сообщения и массив, содержащий параметры сообщения. Псевдопеременная `receiver` ссылается на объект-получатель. Имея эти данные, можно повторно переслать сообщение (выполнить оригинальную член-функцию). В данном случае блок-конверт выводит на экран два знака «-» и пробел, после чего запускает выполнение оригинальной член-функции. Последняя строка листинга 21 меняет метакласс класса `Talkative` с `Class` на `DialogueClass`. В результате все сообщения, выводимые на экран экземплярами классов `Talkative` и `Memorizer`, будут предварены префиксом «`-- `».

Метаклассы, так же как и классы, организованы в иерархию наследования. При этом наследуются не только словари член-функций и член-данных, определённых для классов, но и словари конвертов, определённые для экземпляров этих классов.

Для защиты программных систем путём их интеграции с политикой безопасности (ПБ) посредством АОП предлагается использовать механизм примесей как альтернативу механизму множественного доступа. Этот метод защиты подробно изложен в [3]. Его суть следующая:

1) незащищённая программная система описывается в терминах классов;
2) реализация ПБ описывается в терминах примесей;
3) декларативным способом устанавливается соответствие классов и примесей, а также шаблоны для вызовов алгоритмов-конвертов.

Примеси могут быть описаны в библиотеке АОП с помощью метаклассов. На этапе трансляции происходит объединение классов и примесей в один общий, составной класс с изменённой диспетчеризацией сообщений с помощью алгоритмов-конвертов.

Похожий подход предложен в [10], но аспектно-ориентированная система у нас проще системы, описанной в [10], поскольку она не разделяет статические и динамические свойства системы и описывает оба типа соответствующих примесей единообразно.

5. Транслятор с AspectTalk в промежуточный язык

Технология ВМ широко используется в современном программировании. Существующие ВМ в достаточной мере покрывают потребности языка AspectTalk: 1) обладают возможностями управления памятью, в том числе имеют функцию «сборки мусора»; 2) обладают необходимым набором примитивных операций, в том числе над требуемыми элементарными данными; 3) выполнены в виде отдельной программы. В большинстве случаев языком ВМ является польская инверсная запись, задаваемая последовательностью команд ВМ.

Изначально для реализации AspectTalk была использована стековая ВМ. В настоящее время AspectTalk транслируется в язык Scheme [13], а в качестве ВМ используется интерпретатор Scheme. Использование Scheme в качестве промежуточного языка упрощает трансляцию модулей на метаязыке.

6. Сравнение с другими языками

По ходу изложения уже отмечались отдельные сходства и различия между языком AspectTalk и существующими языками программирования, имеющими метаязыковые конструкции. Перечислим основные из них.

По сравнению с AspectJ [11]:

1) упрощённый синтаксис;
2) явный порядок применения нескольких аспектов;
3) невозможность описания логических парадоксов [12];
4) аспекты, независимые от программы (в т. ч. в аспектной библиотеке).

По сравнению со Scheme [13]:

1) больше возможностей для императивного программирования, привычного для большинства программистов;
2) макросы не имеют системного имени; возможность переопределить базовые конструкции языка с помощью макросов.

В настоящее время AspectTalk транслируется в Scheme, поэтому программы, эквивалентные AspectTalk, можно писать сразу на Scheme. (Эта ситуация аналогична ситуации с первыми трансляторами с C++, которые переводили программы на C++ в программы на C.) При этом использование нового языка для обозначения часто используемых конструкций на другом языке имеет следующие преимущества:

1) AspectTalk поощряет АОП и метапрограммирование в целом;
2) он запрещает небезопасные использования конструкций.

По сравнению со Smalltalk [14]:

1) объектная модель не является частью языка (наследование, словари времени трансляции);
2) объектная библиотека AspectTalk не ограничивает использование метаклассов: есть возможность описывать метаклассы явно, порождать несколько классов от одного метакласса;
3) метапрограммирование не ограничивается работой с метаклассами, разрешены описания точек выполнения с помощью метаязыка;
4) некоторые конструкции базового языка AspectTalk имеют отличную от своих аналогов в Smalltalk семантику.

Стоит отметить также, что язык Python [15], как и AspectTalk, не ограничивает использование метаклассов, но, как и Smalltalk, имеет жёсткую объектную модель и ограниченные возможности метапрограммирования.

По сравнению с MetaclassTalk [10]:

1) аспектная библиотека AspectTalk похожа на АОП в MetaclassTalk;
2) в MetaclassTalk метаобъектный протокол — единственное средство метапрограммирования; метаязык AspectTalk предоставляет возможность реализации более мощной аспектной библиотеки.

По сравнению с Groovy [8]:

1) язык Groovy, как и язык AspectTalk, состоит из двух уровней: языка и метаязыка;
2) метаязык Groovy так же, как и метаязык AspectTalk, позволяет изменять семантику некоторых синтаксических конструкций, однако для этого необходима манипуляция деревом синтаксического разбора как структурой в языке Groovy; это осложняет реализацию метаязыковых модулей.

Заключение

Язык AspectTalk разработан для проведения экспериментальных исследований в области АОП. В основе языка лежат свойства языков Smalltalk, Scheme, MetaclassTalk, Python, AspectJ и Groovy, которые комбинировались и модифицировались для достижения простоты синтаксиса и семантики, отсутствия противоречий (невозможности описания логических парадоксов), возможности реализации АОП, удобного для описания политик безопасности и интеграции их с программами.

Свойства АОП и метапрограммирования, реализованные в AspectTalk, могут использоваться для реализаций аспектно-ориентированных версий других языков программирования. В частности, перспективным направлением представляется реализация JIT-транслятора, переводящего последовательность инструкций для архитектуры x86 в другую последовательность инструкций для этой же архитектуры, но содержащую в себе, помимо оригинальных команд, команды политики безопасности. Возможными средствами для реализации подобного транслятора являются библиотеки [16, 17].

За рамками данной работы остались формальное описание семантики языка AspectTalk и детальное описание его примитивов, стандартной библиотеки и реализации библиотек ООП и АОП в нём и транслятора с него в язык Scheme. Предполагается, что это будет сделано в последующей публикации автора.

ЛИТЕРАТУРА

1. *Стефанцов Д. А.* Реализация политик безопасности в компьютерных системах с помощью аспектно-ориентированного программирования // Прикладная дискретная математика. 2008. № 1(1). С. 94–100.
2. *Стефанцов Д. А.* Технология и инструментальная среда создания защищённых систем обработки информации // Прикладная дискретная математика. Приложение. 2009. № 1. С. 55–56.
3. *Стефанцов Д. А.* Внедрение политик безопасности в программные системы обработки информации // Прикладная дискретная математика. 2011. № 3(13). С. 55–64.
4. *Budd T. A.* An Introduction to Object-Oriented Programming. 3rd edition. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001. 611 p.
5. *Revesz G. E.* Lambda-calculus, Combinators and Functional Programming. New York, NY, USA: Cambridge University Press, 2009. 192 p.
6. `www.boost.org/libs/lambda` — Boost C++ Libraries. Chapter 14. Boost. Lambda. 2009.
7. *Абельсон Х., Сассман Дж. Дж.* Структура и интерпретация компьютерных программ. М.: Добросвет, 2010. 608 с.
8. `http://groovy.codehaus.org/` — Groovy. A dynamic language for the Java platform. 2012.
9. *Kiczales G., des Rivières J., and Bobrow D. G.* The art of metaobject protocol. Cambridge, MA, USA: MIT Press, 1991. 345 p.

10. *Bouraqadi N., Seriai A., and Leblanc G.* Towards unified aspect-oriented programming // ESUG 2005 Research Conf. Brussels, Belgium, 2005. 22 p.
11. `http://www.eclipse.org/aspectj/` — AspectJ. Crosscutting objects for better modularity. 2012.
12. *Forster F. and Steimann F.* AOP and the antinomy of the liar // Workshop on the Foundations of Aspect-Oriented Languages. 2006. P. 47–56.
13. *Sperber M., Dybvig R. K., Flatt M., et al.* Revised[6] Report on the Algorithmic Language Scheme. New York, NY, USA: Cambridge University Press, 2010. 302 p.
14. *Goldberg A., Robson D., and Harrison M. A.* Smalltalk-80: The Language and its Implementation. Boston, MA, USA: Addison-Wesley, 1983. 714 p.
15. `http://python.org/` — Python Programming Language. 2012.
16. `http://www.pintool.org/` — PIN Toool. 2011.
17. `http://dynamorio.org/` — DynamicRIO. Dynamic Instrumentation Tool Platform. 2012.

ПРИКЛАДНАЯ ТЕОРИЯ ГРАФОВ

DOI 10.17223/20710410/15/8

УДК 519.17

ХАРАКТЕРИЗАЦИЯ ГРАФОВ С ЗАДАННЫМ ЧИСЛОМ ДОПОЛНИТЕЛЬНЫХ РЕБЕР МИНИМАЛЬНОГО ВЕРШИННОГО 1-РАСШИРЕНИЯ

М. Б. Абросимов

Саратовский государственный университет им. Н. Г. Чернышевского, г. Саратов, Россия

E-mail: mic@rambler.ru

Рассматривается задача описания графов, минимальное вершинное 1-расширение которых имеет заданное число дополнительных ребер. Дается решение, когда число дополнительных ребер равно одному, двум и трем.

Ключевые слова: *граф, минимальное вершинное 1-расширение, точное вершинное 1-расширение, оптимальная отказоустойчивая реализация.*

Введение

Понятие вершинного (реберного) расширения происходит из работ [1 – 3], в которых удалось описать минимальные вершинные расширения для некоторых видов графов. В работе [4] было доказано, что задача проверки вершинного расширения графа является NP-полной, и поэтому в общем виде задача описания минимальных вершинных расширений произвольного графа, скорее всего, решена быть не может. Основное направление исследований в этой области продолжает следовать подходу Дж. П. Хейза, при котором описывается частное решение для графов определенного вида. Можно выделить и другие пути. В данной работе рассматривается задача описания графов, которые имеют минимальные вершинные 1-расширения с заданным числом дополнительных ребер.

Граф $G^* = (V^*, \alpha^*)$ называется *минимальным вершинным k-расширением* (МВ-kР, k — натуральное) n-вершинного графа $G = (V, \alpha)$, если выполняются следующие условия:

1) граф G^* является вершинным k-расширением графа G, то есть граф G вкладывается в каждый подграф графа G^*, получающийся удалением любых его k вершин;
2) граф G^* содержит $n + k$ вершин, то есть $|V^*| = |V| + k$;
3) α^* имеет минимальную мощность при выполнении условий 1 и 2.

Граф H называется *точным вершинным k-расширением* графа G, если любой граф, получающийся удалением произвольных k вершин графа H, изоморфен графу G.

Будем использовать основные понятия теории графов, опираясь преимущественно на работу [5]. Очевидно, что минимальным вершинным 1-расширением вполне несвязного графа O_n (то есть графа без ребер) будет вполне несвязный граф O_{n+1}. Заметим далее, что если к произвольному n-вершинному графу G добавить одну вершину и соединить со всеми остальными вершинами, то получившийся граф будет являться

вершинным 1-расширением графа G (*тривиальным*). Очевидно, что количество добавленных ребер есть n. Для некоторых графов тривиальное вершинное 1-расширение является и минимальным, например для полного графа K_n. Таким образом, число дополнительных ребер минимального вершинного 1-расширения произвольного n-вершинного графа заключено между 0 и n.

1. Вспомогательные результаты

Докажем несколько вспомогательных утверждений о свойствах минимальных вершинных k-расширений.

Лемма 1. Минимальное вершинное k-расширение графа без изолированных вершин не содержит вершин со степенью ниже $k+1$.

Лемма 2. Пусть наибольшая из степеней вершин графа G есть s и в точности m вершин имеют такую степень, тогда минимальное вершинное k-расширение графа G содержит, по крайней мере, $k+m$ вершин степени не ниже s.

Лемма 3. Если максимальная степень вершины графа G есть $d > 0$, то его минимальное вершинное k-расширение G^* содержит не менее kd дополнительных ребер.

Лемма 4. Если минимальная степень вершины графа G есть $d > 0$, то его минимальное вершинное k-расширение G^* не содержит вершин степени ниже $d+k$.

Лемма 5. Если максимальная степень вершины минимального вершинного 1-расширения графа есть d, то число дополнительных ребер в расширении не меньше d.

Доказательство. Пусть G — n-вершинный граф из условия соответствующей леммы, G^* — минимальное вершинное k-расширение графа G с числом вершин $(n+k)$.

1. Лемма 1 является частным случаем леммы 4.

2. Если число вершин степени не ниже s в G^* меньше $k+m$, то, удалив k таких вершин (если число вершин меньше k, то можно удалить все вершины степени не ниже s и подходящее количество любых других вершин), получим граф, в котором менее m вершин степени не ниже s. В такой граф нельзя вложить граф G.

3. После удаления любых k вершин из графа G^* в получившемся графе должна оставаться по крайней мере одна вершина степени d. Будем выбирать для удаления из графа G^* последовательно k вершин наибольшей степени. Так как степень каждой вершины не меньше d, то и количество удаленных ребер — не менее kd.

4. Пусть граф G^* имеет вершину v степени ниже $d+k$. Рассмотрим подграф, получающийся из G^* удалением k смежных с v вершин (если степень вершины v меньше k, то можно удалить все смежные с ней вершины и подходящее количество любых других вершин, кроме v). Он содержит вершину степени ниже d, и в него нельзя вложить граф G.

5. Удаление вершины v наибольшей степени d из графа G^* приводит к удалению и d ребер, а по условию граф $G^* - v$ допускает вложение графа G. ■

Замечание 1. Из леммы 1 следует, что минимальное вершинное 1-расширение любого связного графа содержит как минимум 2 дополнительных ребра.

2. Основные результаты

Теорема 1. Минимальное вершинное k-расширение вполне несвязного n-вершинного графа O_n единственно и есть вполне несвязный $(n+k)$-вершинный граф O_{n+k}. Никакие другие графы не могут иметь минимальные вершинные k-расширения с нулевым числом дополнительных ребер.

Доказательство. Если у графа есть хотя бы одно ребро, то по лемме 5 число дополнительных ребер минимального вершинного k-расширения больше нуля. ■

Степенным множеством графа называется множество, составленное из чисел, являющихся степенями вершин графа. Под *объединением* двух графов $G_1 = (V_1, \alpha_1)$ и $G_2 = (V_2, \alpha_2)$ понимается граф $G_1 \cup G_2 = (V_1 \cup V_2, \alpha_1 \cup \alpha_2)$.

Теорема 2. Графы со степенным множеством $\{1, 0\}$ и только они имеют минимальные вершинные 1-расширения с одним дополнительным ребром; для каждого графа со степенным множеством $\{1, 0\}$ такое расширение единственно с точностью до изоморфизма.

Доказательство. Рассмотрим произвольный граф G. Обозначим через d максимальную из степеней его вершин. Пусть далее G^* — минимальное вершинное 1-расширение графа G, которое отличается от него на одно дополнительное ребро. Граф G отличен от вполне несвязного, так как минимальное вершинное 1-расширение вполне несвязного графа по теореме 1 не имеет дополнительных ребер. Следовательно, $d > 0$.

По лемме 3 $d < 2$, так как в противном случае число дополнительных ребер минимального вершинного 1-расширения больше единицы. Итак, $d = 1$.

Если граф не содержит изолированных вершин, то его степенное множество имеет вид $\{1\}$ — это графы, являющиеся объединением некоторого количества полных двухвершинных графов K_2. По лемме 1 минимальное вершинное 1-расширение любого такого графа не содержит вершин со степенью ниже 2, а по лемме 5 количество дополнительных ребер также не менее 2. Таким образом, только графы со степенным множеством $\{1, 0\}$ могут иметь минимальное вершинное 1-расширение, отличающееся на одно дополнительное ребро. Покажем, что они действительно имеют такое расширение, причем единственное.

Граф G со степенным множеством $\{1, 0\}$ можно представить в виде $K_2 \cup ... \cup K_2 \cup O_p$, где $p > 0$ (рис. 1,*а*). На рис. 1,*б* представлено минимальное вершинное 1-расширение графа G, которое отличается как раз на одно дополнительное ребро. Легко видеть, что других минимальных вершинных 1-расширений граф G иметь не может: при $p > 1$ минимальное вершинное 1-расширение будет иметь степенное множество $\{1, 0\}$, а при $p = 1$ — степенное множество $\{1\}$. ■

а *б*

Рис. 1. Граф со степенным множество $\{1, 0\}$ (*а*) и его минимальное вершинное 1-расширение (*б*)

Теорема 3 (Дж. П. Хейз [1]). Единственное с точностью до изоморфизма минимальное вершинное 1-расширение n-вершинной цепи P_n есть $(n + 1)$-вершинный цикл C_{n+1}.

Теорема 4. Среди связных графов только цепи имеют минимальные вершинные 1-расширения с двумя дополнительными ребрами; для каждой цепи такое расширение единственно с точностью до изоморфизма.

Доказательство. Рассмотрим произвольный связный граф G. Обозначим через d максимальную из степеней его вершин. Пусть далее G^* — минимальное вершинное 1-расширение графа G, которое отличается от него на два дополнительных ребра. По лемме 1 граф G^* не содержит вершин степени меньше 2, а по лемме 5 он не может содержать вершин степени больше 2, таким образом, граф G^* является регулярным графом степени 2.

Так как граф G связный, то $d > 0$. По лемме 3 $d < 3$, так как в противном случае число дополнительных ребер минимального вершинного 1-расширения было бы больше двух. Итак, установлено, что $d = 1$ или $d = 2$. Рассмотрим оба случая.

Если $d = 1$, то граф G — это 2-вершинная цепь P_2, которая имеет единственное минимальное вершинное 1-расширение — цикл C_3, причем количество дополнительных ребер в точности равно двум.

Если $d = 2$, то граф G может иметь степенное множество либо $\{2\}$, либо $\{2, 1\}$. В первом случае по лемме 4 его минимальное вершинное 1-расширение не может содержать вершин степени ниже 3, а по лемме 5 получаем, что и количество дополнительных ребер должно быть не менее трех.

Заметим, что все проведенные до этого момента рассуждения справедливы не только для связных графов, но и для любых графов без изолированных вершин.

Таким образом, среди связных графов с числом вершин $n > 2$ только граф со степенным множеством $\{2, 1\}$ может иметь минимальное вершинное 1-расширение с двумя дополнительными ребрами. Однако связные графы со степенным множеством $\{2, 1\}$ — это цепи. По теореме 3 цепи удовлетворяют утверждению теоремы. ■

Теорема 5. Среди несвязных графов без изолированных вершин только графы вида $P_n \cup C_{n+1} \cup ... \cup C_{n+1}$ при $n > 1$ имеют минимальные вершинные 1-расширения с двумя дополнительными ребрами, причем это расширение с точностью до изоморфизма совпадает с $C_{n+1} \cup C_{n+1} \cup ... \cup C_{n+1}$.

Доказательство. Рассмотрим произвольный несвязный граф G без изолированных вершин. Пусть граф G^* — минимальное вершинное 1-расширение графа G, которое отличается на два дополнительных ребра.

Как было установлено в доказательстве предыдущей теоремы, граф G^* является регулярным графом степени 2, а граф G имеет степенное множество вида $\{2, 1\}$. По условию граф G^* отличается на одну дополнительную вершину и два дополнительных ребра, следовательно, в графе G может быть только две вершины степени 1, которые соединяются в графе G^* ребрами с дополнительной вершиной. Из этих рассуждений можно сделать несколько выводов:

1) граф G имеет единственную компоненту, которой является цепь, а все остальные компоненты являются циклами;

2) все компоненты графа G^* являются циклами;

3) граф G^* — единственное минимальное вершинное 1-расширение графа G.

Докажем, что все компоненты связности графа G^* изоморфны.

Обозначим через P_n компоненту связности графа G, являющуюся цепью. Рассмотрим удаление произвольной вершины v в графе G^*. Граф $G^* - v$ имеет столько же ребер, сколько и граф G, причем G вкладывается в граф $G^* - v$, следовательно, $G^* - v$ изоморфен графу G. Но тогда компонента графа G^* с удаленной вершиной v изоморфна цепи P_n, то есть исходная компонента в графе G^* была циклом C_{n+1}. В силу произвольности выбора получаем, что все компоненты связности в графе G^* изо-

морфны циклу C_{n+1}, то есть граф G^* имеет вид $C_{n+1} \cup C_{n+1} \cup ... \cup C_{n+1}$, а граф G — $P_n \cup C_{n+1} \cup ... \cup C_{n+1}$. ■

Замечание 2. В теоремах 1–5 минимальные вершинные 1-расширения являются и точными вершинными 1-расширениями.

Вектором степеней графа называется вектор, составленный из степеней вершин графа в порядке невозрастания.

Теорема 6. Связные графы, имеющие минимальные вершинные 1-расширения с тремя дополнительными ребрами, могут иметь только следующий вид:

1) полный граф K_3;
2) графы с вектором степеней вида $(3, \ldots, 3, 2, 2, 2)$, имеющие точное вершинное 1-расширение;
3) графы с вектором степеней $(3, 3, 3, \ldots, 3, 2, \ldots, 2, 1)$ особого вида.

Доказательство. Рассмотрим произвольный связный n-вершинный граф G. Пусть G^* — минимальное вершинное 1-расширение графа G, которое отличается от него на три дополнительных ребра. По лемме 3 в графе G не может быть вершин со степенью больше 3. Перечислим все возможные степенные множества для графа G и затем последовательно их рассмотрим: $\{1\}$, $\{2\}$, $\{2, 1\}$, $\{3\}$, $\{3, 1\}$, $\{3, 2\}$, $\{3, 2, 1\}$. Заметим, что по лемме 5 в графе G^* не может быть вершины со степенью больше 3. Кроме того, если в графе G^* есть вершина v степени 3, то граф $G^* - v$ будет изоморфен графу G: действительно, граф G должен вкладываться в любой максимальный подграф графа G^*, но граф $G^* - v$ содержит столько же ребер, сколько и граф G, откуда и следует их изоморфизм.

$\{1\}$: из связных графов такое степенное множество может иметь только цепь P_2, единственное минимальное вершинное 1-расширение которой по теореме 3 есть цикл C_3.

$\{2\}$: из связных графов такое степенное множество могут иметь только циклы C_n. По леммам 4 и 5 граф G^* должен быть кубическим графом. Количество ребер в графах G и G^* равно n и $3(n+1)/2$ соответственно. Тогда количество дополнительных ребер графа G^* составит $(n+3)/2$, и только при $n = 3$ оно равно трем. Цикл C_3 изоморфен полному графу K_3, и его минимальное вершинное 1-расширение — полный граф K_4 — отличается от него на три дополнительных ребра. Получаем случай 1 утверждения теоремы.

$\{2, 1\}$: из связных графов такое степенное множество может иметь только цепь P_n. По теореме 3 единственное минимальное вершинное 1-расширение цепи P_n — цикл C_{n+1}, и оно отличается на 2 дополнительных ребра.

$\{3\}$: по лемме 4 в графе G^* степень вершин не ниже 4, а по лемме 5 число дополнительных ребер тогда также не менее 4, следовательно, этот случай исключается.

$\{3, 1\}$: с учетом леммы 4 граф G^* может иметь степенное множество вида $\{3\}$ или $\{3, 2\}$. Первое степенное множество соответствует кубическому графу и в данном случае не подходит, так как при удалении одной вершины кубического графа не может получиться граф с вершиной степени 1. Итак, граф G^* может иметь только степенное множество вида $\{3, 2\}$. Рассмотрим удаление из графа G^* произвольной вершины v степени 3. Как было отмечено ранее, граф $G^* - v$ изоморфен графу G. Если вершина v смежна хотя бы с одной другой вершиной степени 3, то в графе $G^* - v$, а значит, и в графе G есть вершина степени 2, что невозможно. Но тогда в графе G^* вершина v должна быть смежна только с вершинами степени 2. В силу произвольности выбора вершины степени 3 получаем, что граф G^* в данном случае может иметь только вид

$K_{3,2} = O_3 + O_2$ (рис. 2); тогда граф G имеет вид $K_{3,1} = O_1 + O_3$. Легко видеть, что граф $K_{3,2}$ не является минимальным вершинным 1-расширением графа $K_{3,1}$. В самом деле, удаление любой вершины степени 2 из графа $K_{3,2}$ приводит к графу C_4, в котором нет вершин степени 3.

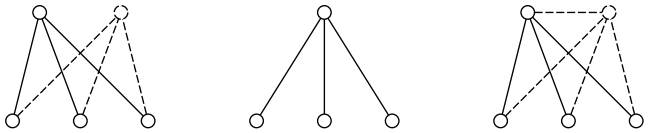
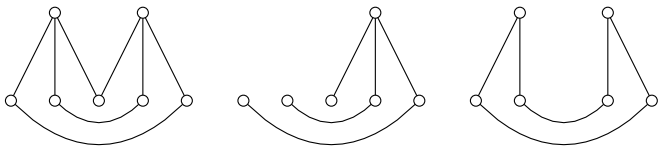


Рис. 2. Графы $K_{3,2}$, $K_{3,1}$ и MB-1P графа $K_{3,1}$

$\{3, 2\}$: с учетом леммы 4 граф G^* является кубическим графом. Любой максимальный подграф графа G^* изоморфен графу G, то есть граф G^* является точным вершинным 1-расширением графа G. Получаем случай 2 утверждения теоремы.

$\{3, 2, 1\}$: аналогично случаю степенного множества $\{3, 1\}$ приходим к выводу, что граф G^* должен иметь степенное множество $\{3, 2\}$. Рассмотрим удаление произвольной вершины v степени 3 из графа G^*. Получившийся граф имеет столько же ребер, сколько и граф G, а следовательно, граф $G^* - v$ изоморфен G. Это означает, что каждая вершина степени 3 в графе G^* соединена с одинаковым количеством вершин степени 2: с 0, 1, 2 или 3 вершинами. Первый случай можно исключить из рассмотрения, так как если вершины степени 3 не смежны ни с одной вершиной степени 2, то граф G^* несвязный. Рассмотрим оставшиеся случаи. Обозначим через m_1 количество вершин степени 3 в графе G^*. По теореме о четности числа вершин нечетной степени число m_1 четно.

I с л у ч а й. Пусть каждая вершина степени 3 графа G^* смежна с 3 вершинами степени 2. Предположим, что в графе G^* есть вершина u степени 2, смежная с двумя вершинами степени 3 (рис. 3,*а*). Но тогда в графе $G^* - v$ (а значит, и в графе G) будет $m_1 - 1$ вершин степени 3 (рис. 3,*б*), а в графе $G^* - u$ будет $m_1 - 2$ вершин степени 3 (рис. 3,*в*), то есть граф G не вкладывается в граф $G^* - u$, что противоречит предположению.



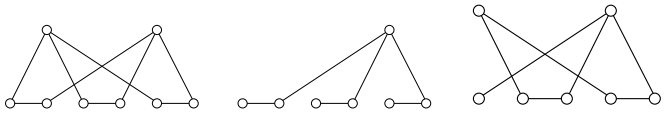
а *б* *в*

Рис. 3. Случай I

Итак, в графе G^* каждая вершина степени 2 смежна не более чем с одной вершиной степени 3 (рис. 4,*а*). Это означает, что вершины степени 3 соединены цепями, состоящими из вершин степени 2. Обозначим кратчайшее расстояние между вершинами степени 3 через d_3. Так как вершины степени 3 несмежны между собой, то $d_3 > 1$. Более того, так как в графе G^* нет вершин степени 2, смежных с двумя вершинами степени 3, то $d_3 > 2$. Не ограничивая общности, будем считать, что расстояние между вершинами v_1 и v_2 степени 3 равно d_3.

Рассмотрим граф $G^* - v_1$. В этом графе $m_1 - 1$ вершин степени 3, из вершины v_2 выходит цепь длины $d_3 - 1$, и это самая короткая цепь (рис. 4,*б*).

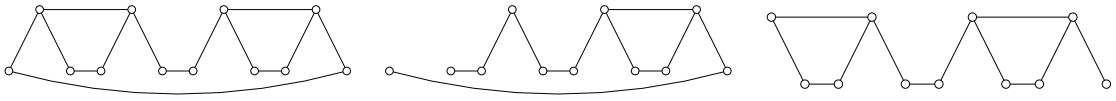
Рассмотрим граф, получающийся удалением из графа G^* первой вершины степени 2 в цепи, соединяющей вершину v_1 с v_2. В этом графе также $m_1 - 1$ вершин степени 3, а из вершины v_2 выходит цепь длины $d_3 - 2$ (рис. 4,*в*). Очевидно, что этот граф не может содержать в себе граф $G^* - v_1$.



а *б* *в*

Рис. 4. Случай I. Продолжение

II с л у ч а й. Пусть каждая вершина степени 3 смежна с двумя вершинами степени 2. Рассмотрим пару вершин u_1, u_2 степени 2, смежных с вершиной v степени 3. Если вершины u_1 и u_2 смежны, то граф $G^* - v$ несвязный. Следовательно, граф G^* представляет собой $m_1/2$ пар смежных вершин степени 3. С каждой вершиной степени 3 смежна пара несмежных вершин степени 2 (рис. 5,*а*). Рассмотрим граф $G^* - v$. В этом графе $m_1/2 - 1$ пар смежных вершин степени 3 и две вершины степени 1, смежные с вершинами степени 2. Таким образом, расстояние от вершины степени 1 до ближайшей вершины степени 3 не менее 2 (рис. 5,*б*).



а *б* *в*

Рис. 5. Случай II

Рассмотрим граф $G^* - u$, где u — произвольная вершина степени 2 (рис. 5,*в*). Этот граф должен допускать вложение графа G (или изоморфного ему графа $G^* - v$) и отличается от него на одно дополнительное ребро. В графе G^* вершина u смежна с одной вершиной степени 3 и с одной вершиной степени 2; обозначим эти вершины u_1 и w соответственно. В графе $G^* - u$, следовательно, будет $m_1/2 - 1$ пар смежных вершин степени 3 и еще одна вершина u_2 степени 3, смежная только с вершинами степени 2. Вершина w в графе $G^* - u_1$ будет единственной вершиной степени 1, причем она смежна с вершиной степени 3. Предположим, что вложение графа G в граф $G^* - u$ построено. Тогда вершине u_2 может соответствовать только вершина степени 2 и лишнее ребро — ребро, инцидентное вершине u_2. Обозначим через H граф, получающий из графа $G^* - u$ после удаления этого ребра. Тогда H должен быть изоморфен $G^* - v$, но в графе H вершина степени 1 смежна с вершиной степени 3, а в графе $G^* - v$ вершины степени 1 смежны вершинам степени 2. Получаем противоречие.

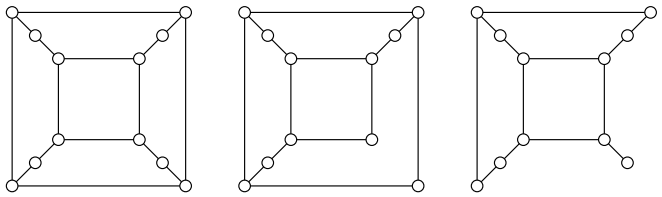
III с л у ч а й. Итак, остается последняя возможность. Пусть каждая вершина степени 3 смежна с одной вершиной степени 2 и с двумя вершинами степени 3. Но тогда в графе G^* должно быть не менее четырёх вершин степени 3, и их количество должно быть четным. Граф $G^* - v$ должен быть изоморфен графу G, одна его вершина имеет степень 1, а остальные — степень 2 или 3.

Пусть u — произвольная вершина графа G^* степени 2. Вершина u не может быть смежна с двумя вершинами степени 2, так как в этом случае в графе $G^* - u$ окажется две вершины степени 1 и вложение графа G будет невозможно. Таким образом, каждая вершина степени 2 смежна либо с двумя вершинами степени 3, либо с одной.

Каждая вершина степени 3 графа G^* смежна с двумя другими вершинами степени 3, то есть граф, индуцированный всеми вершинами степени 3, представляет собой цикл или объединение нескольких циклов. Покажем, что последнее невозможно. Предположим противное и обозначим через k количество циклов, составленных из вершин степени 3, $k > 1$. Будем рассматривать далее только эти циклы. Граф G изоморфен графу, получающемуся из G^* удалением любой вершины степени 3. Удалив одну вершину степени 3, разомкнем один цикл, составленный из вершин степени 3, и в получившемся графе окажется на один цикл меньше: $k - 1$.

Относительно вершин степени 2 рассмотрим две возможности.

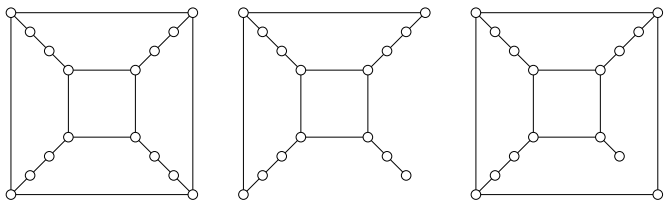
1. В графе G^* есть вершина u степени 2, смежная с двумя вершинами степени 3, которые принадлежат различным циклам (рис. 6,*а*). Удаление вершины u из графа G^* приведет к тому, что циклов останется $k - 2$ (рис. 6,*б*). Тогда в граф $G^* - u$ нельзя будет вложить граф G (рис. 6,*в*).



а *б* *в*

Рис. 6. Случай III

2. В графе G^* нет вершин степени 2, смежных с двумя вершинами степени 3, которые принадлежат различным циклам. Это означает, что все вершины степени 2 смежны с одной вершиной степени 3 и с одной вершиной степени 2 (рис. 7,*а*). Удаление вершины u степени 2 из графа G^* приведет к тому, что циклов останется $k - 1$, как и в графе G (рис. 7,*б*), однако в графе G единственная вершина степени 1 смежна с вершиной степени 3, а в графе $G^* - u$ расстояние от единственной вершины степени 1 до вершины степени 3 равно двум. Очевидно, что в граф $G^* - u$ нельзя вложить граф G (рис. 7,*в*).



а *б* *в*

Рис. 7. Случай III. Продолжение

Таким образом, получается, что граф G^*, если он является минимальным вершинным 1-расширением графа G и отличается от него на 3 дополнительных ребра, должен иметь следующий вид:

1) степенное множество $\{3, 2\}$;

2) количество вершин степени 3 четно и больше 3;

3) вершины степени 2 смежны либо с двумя вершинами степени 3, либо с одной вершиной степени 3 и одной вершиной степени 2, то есть вершины степени 3 соединяются цепью, состоящей не более чем из трех ребер;

4) вершины степени 3 смежны с двумя другими вершинами степени 3, причем граф, индуцированный всеми вершинами степени 3, представляет собой цикл.

Получаем случай 3 утверждения теоремы. ■

Следствие 1. Семейство из п. 3 теоремы содержит графы с числом вершин $3k-1$ и $4k-1$, $k > 1$.

Доказательство. Из доказанных свойств минимальных вершинных 1-расширений графов рассматриваемого семейства следует, что в них каждая вершина степени 3 соединяется цепью из одной или двух вершин степени 2 с другой вершиной степени 3, причём длины цепей одинаковы для всех пар. Если $2k$ — количество вершин степени 3, то на каждую из k пар вершин степени 3 приходится либо одна, либо две вершины степени 2 минимального вершинного 1-расширения. В первом случае получим $3k$ вершин, во втором — $4k$.

Заметим, что для заданного числа вершин всегда можно построить соответствующего представителя семейства. Для этого достаточно построить цикл с числом вершин $2k$, разбить вершины цикла на k пар смежных вершин и каждую из этих пар соединить цепью из одной (двух) вершин. Удалив любую вершину степени 3 из получившегося графа, получим искомого представителя семейства. ■

На рис. 8 представлены все графы и их минимальные вершинные 1-расширения с числом вершин до 8, соответствующие п. 3 доказанной теоремы.

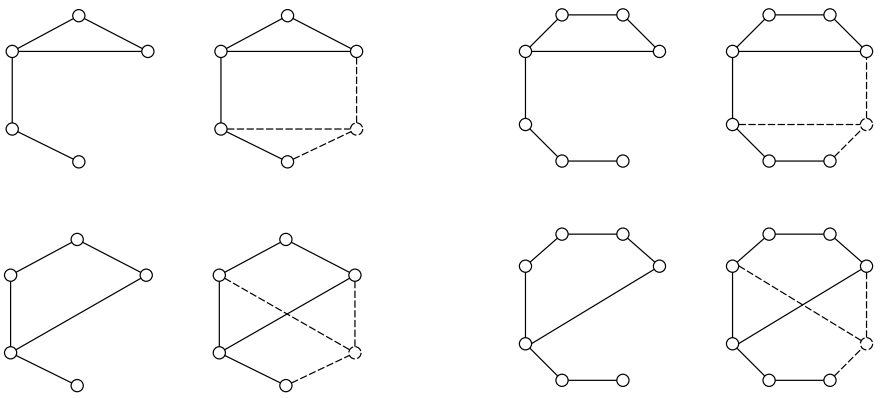


Рис. 8. Малые графы и их МВ-1Р с тремя дополнительными ребрами

Заметим, что во всех рассмотренных случаях графы имеют единственное с точностью до изоморфизма минимальное вершинное 1-расширение.

ЛИТЕРАТУРА

1. *Hayes J. P.* A graph model for fault-tolerant computing system // IEEE Trans. Comput. 1976. V. C25. No. 9. P. 875–884.
2. *Harary F. and Hayes J. P.* Edge fault tolerance in graphs // Networks. 1993. V. 23. P. 135–142.
3. *Harary F. and Hayes J. P.* Node fault tolerance in graphs // Networks. 1996. V. 27. P. 19–23.

4. *Абросимов М. Б.* О сложности некоторых задач, связанных с расширениями графов // Матем. заметки. 2010. № 5(88). С. 643–650.
5. *Богомолов А. М., Салий В. Н.* Алгебраические основы теории дискретных систем. М.: Наука, 1997.

ДИСКРЕТНЫЕ МОДЕЛИ РЕАЛЬНЫХ ПРОЦЕССОВ

DOI 10.17223/20710410/15/9

УДК 519.17

КЛЕТОЧНО-АВТОМАТНАЯ МОДЕЛЬ ДИНАМИКИ ЧИСЛЕННОСТИ ОРГАНИЗМОВ ОЗЕРА БАЙКАЛ[1]

И. В. Афанасьев

Институт вычислительной математики и математической геофизики СО РАН, г. Новосибирск, Россия

E-mail: ivafanas@gmail.com

Предложена и исследована клеточно-автоматная модель динамики численности организмов озера Байкал. Проведены вычислительные эксперименты по моделированию распределения плотности организмов по пространству при начальном скоплении хищников, а также в случае загрязнения области.

Ключевые слова: *самоорганизация, дискретное моделирование, клеточный автомат, композиционные модели, модель численности, хищник — жертва.*

Введение

Под самоорганизацией понимается процесс упорядочения (пространственного, временного или пространственно-временного) в системе за счёт согласованного взаимодействия множества элементов её составляющих. Такое определение было дано Г. Хакеном в 1980 г. в рамках науки синергетики [1].

Известно большое число самоорганизующихся процессов в биологии, химии, физике, социологии, экономике. Примерами служат процессы смены агрегатных состояний веществ, кристаллизации, лазер, спонтанная свертка протеинов, гомеостаз, окраска животных и т. д.

С развитием исследований в области самоорганизации возникает потребность в построении моделей для процессов пространственно-временной самоорганизации. В основном модели базируются на нелинейных дифференциальных уравнениях в частных производных, которые сложно, а иногда и невозможно решать и эффективно распараллелить. Альтернативным решением являются клеточно-автоматные (КА) модели, позволяющие, используя сравнительно простые правила, моделировать сложные нелинейные процессы, включая и процессы самоорганизации. Исследования КА-моделей самоорганизации проводили S. Wolfram [2], L. O. Chua [3], В. К. Ванаг [4]. Имеется КА-модель для некоторых реакций Белоусова — Жаботинского [5], исследована КА-модель для процесса «разделения фаз» [6, 7].

КА представляет собой структурированный набор конечных (элементарных) автоматов (клеток). Для каждой клетки определено множество соседних клеток и функция перехода, зависящая от их состояний. Структура пространства, в котором расположены клетки, для двумерного случая представляется прямоугольной, треугольной или шестиугольной сеткой. Существуют методы композиции КА, позволяющие объединять несколько КА в более сложные системы — композиционные КА [8]. Параллель-

[1]Работа поддержана грантом РФФИ № 11-01-00567а и Программой Президиума РАН № 14-6 (2001).

ная композиция двух и более КА заключается в том, что все КА функционируют одновременно, а оператор перехода каждого КА дополнительно зависит от состояний клеток остальных КА. Была предложена и исследована параллельно-композиционная КА-модель для задачи «хищник — жертва», описывающая взаимодействие двух групп организмов [8].

Известны работы по изучению динамики популяции с помощью систем дифференциальных уравнений [9, 10]. В них принимаются следующие ограничения:

1) гипотеза о сильном перемешивании (параметры популяции осреднены по пространству);
2) более трёх групп организмов не исследуется.

В работе [11], посвященной исследованию динамики популяции восьми групп организмов, с помощью численного моделирования было снято второе ограничение.

В данной работе представлена параллельно-композиционная КА-модель динамики численности организмов озера Байкал для восьми групп, снимающая также и первое ограничение. Модель позволяет учитывать влияния локальных изменений (загрязнений, случайных скоплений). При моделировании использованы данные о популяции рыб Байкала из работы [11].

В п. 1 дана постановка задачи моделирования динамики численности популяции рыб Байкала. Описаны межгрупповые демографические взаимодействия и взаимодействия вида «хищник — жертва». В п. 2 приводится описание КА-модели популяций. В п. 3 приведены результаты вычислительных экспериментов для двух случаев:

1) наличия косяка рыб в начальном состоянии;
2) перманентного загрязнения области.

Экспериментально показано, что после серии затухающих колебаний модель приходит к устойчивому состоянию.

1. Постановка задачи

Учёт всех видов организмов фауны Байкала — трудоемкая задача. По биомассе озера лидерство занимает голомянка (60 % биомассы). Её основной корм — рачок макрогектопуса и собственная молодь. Аналогично [11] в КА-модели рассмотрены три вида организмов: рачок макрогектопуса, малая голомянка и большая голомянка. Каждый из видов разделен на возрастные группы.

Рачок макрогектопуса (*Macrohectopus branickii*):

1) m_1 неполовозрелые особи;
2) m_2 половозрелые особи.

Малая голомянка (*Comephorus dybowski*):

1) d_1 однолетки;
2) d_2 неполовозрелые;
3) d_3 половозрелые.

Большая голомянка (*Comephorus baikalensis*):

1) b_1 однолетки;
2) b_2 неполовозрелые;
3) b_3 половозрелые.

Далее везде верхний индекс обозначает название вида (m — рачок макрогектопуса, d — малая голомянка, b — большая голомянка), а нижний индекс — номер возрастной группы $i \in \{1, 2, 3\}$. Соотношения пищевых цепей и переход особей между возрастными группами в результате старения и рождения изображены на рис. 1.

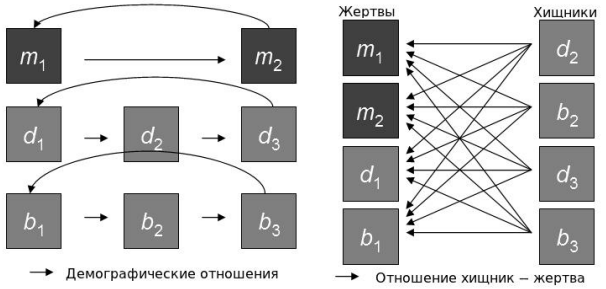


Рис. 1. Отношения между различными группами

Неполовозрелые особи макрогектопуса (возрастная группа 1) порождаются половозрелыми особями, в результате старения становятся половозрелыми особями макрогектопуса. Служат кормом для неполовозрелых и половозрелых особей малой и большой голомянок. Предполагается, что корма неполовозрелым рачкам макрогектопуса всегда хватает.

Половозрелые особи макрогектопуса (возрастная группа 2) появляются в результате старения неполовозрелых особей макрогектопуса, порождают неполовозрелых особей макрогектопуса. Служат кормом для неполовозрелых и половозрелых особей малой и большой голомянок. Предполагается, что корма половозрелым рачкам макрогектопуса всегда хватает.

Однолетки малой голомянки (возрастная группа 1) порождаются половозрелыми особями малой голомянки, в результате старения становятся неполовозрелыми особями малой голомянки. Служат кормом для неполовозрелых и половозрелых особей малой и большой голомянок. Предполагается, что корма однолеткам малой голомянки всегда хватает.

Неполовозрелые особи малой голомянки (возрастная группа 2) появляются в результате старения однолеток малой голомянки, сами в результате старения переходят в половозрелых особей малых голомянок. Питаются неполовозрелыми и половозрелыми особями макрогектопуса, а также однолетками малой и большой голомянок.

Половозрелые особи малой голомянки (возрастная группа 3) порождают однолеток малой голомянки, появляются за счет старения неполовозрелых особей малой голомянки. Питаются неполовозрелыми и половозрелыми особями макрогектопуса, а также однолетками малой и большой голомянок.

Однолетки большой голомянки (возрастная группа 1) порождаются половозрелыми особями большой голомянки, в результате старения становятся неполовозрелыми особями большой голомянки. Служат кормом для неполовозрелых и половозрелых особей малой и большой голомянок. Предполагается, что корма однолеткам большой голомянки всегда хватает.

Неполовозрелые особи большой голомянки (возрастная группа 2) появляются в результате старения однолеток большой голомянки, сами в результате старения переходят в половозрелых особей большой голомянки. Питаются неполовозрелыми и половозрелыми особями макрогектопуса и однолетками малой и большой голомянок.

Половозрелые особи большой голомянки (возрастная группа 3) порождают однолеток большой голомянки, появляются за счет старения неполовозрелых особей большой голомянки. Питаются неполовозрелыми и половозрелыми особями макрогектопуса, а также однолетками малой и большой голомянок.

Для исследования процессов взаимодействия между этими видами организмов в изменяющихся условиях озера Байкал построена КА-модель, учитывающая пространственное распределение организмов по озеру.

2. КА-модель

При построении КА-модели были приняты следующие допущения:

1) Влияние других видов организмов, являющихся пищей или хищниками по отношению к исследуемым восьми группам организмов, считается постоянным. Предполагается, например, что рачкам макрогектопуса всегда хватает фитофага эпишуры для питания, а нерпа всегда ест одинаковое число голомянок.
2) Влияние окружающей среды считается постоянным. В модели не учтены сезонные особенности поведения организмов и окружающей среды.
3) Особи каждой группы организмов осреднены по всем параметрам (скорость, размеры, рацион питания, половое соотношение и т. д.).
4) Размножаются только особи последних возрастных групп.
5) Насыщение не учитывается.

Определим КА-модель динамики численности организмов Байкала:

$$\aleph = \langle \Sigma, M, f, \rho \rangle, \tag{1}$$

где Σ — алфавит состояний; M — множество имен клеток; f — глобальный оператор перехода; ρ — режим функционирования.

Модель представляет собой параллельную композицию восьми КА, каждый из которых предназначен для моделирования динамики численности отдельной группы организмов. Множество имен клеток M разбито на 8 попарно непересекающихся подмножеств, соответствующих этим группам; каждое из подмножеств биективно отображается на квадратную сетку $N \times N$:

$$M = M_1^m \cup M_2^m \cup M_1^d \cup M_2^d \cup M_3^d \cup M_1^b \cup M_2^b \cup M_3^b;$$
$$M_i^\alpha \cap M_j^\beta \neq \varnothing \Leftrightarrow i = j \ \wedge \ \alpha = \beta;$$
$$M_i^\alpha = \{(p,q)_i^\alpha : 1 \leqslant p \leqslant N,\ 1 \leqslant q \leqslant N\}.$$

Клеткой называется элемент множества $M \times \Sigma$. Множество $\Omega \subseteq M \times \Sigma$, такое, что $|\Omega| = |M|$ и имена всех клеток из Ω уникальны, называется *клеточным массивом*. Состояниями клеток являются целые числа $n \in \Sigma$, обозначающие численность особей данного вида и возрастной группы, находящихся в клетке. Клетки вида $(p,q)_i^\alpha$ и $(p,q)_j^\beta$ будем называть *близнецами*. Клетки с именами $(p_1,q_1)_i^\alpha$ и $(p_2,q_2)_i^\alpha$ называют *ближайшими соседями*, если $\mid p_1 - p_2 \mid + \mid q_1 - q_2 \mid = 1$.

Конечный набор $S(p,q) = \langle (\varphi_1(p,q), n_1), \ldots, (\varphi_k(p,q), n_k) \rangle$, где $n_i \in \Sigma$, $\varphi_i : M \to M$, называется *локальной конфигурацией*. В данной работе рассматриваются функции φ вида $\varphi(p,q) = ((p+a) \bmod N, (q+b) \bmod N)$, где a,b — некоторые целые константы. Значения индексов $p+a$ и $q+b$ берутся по модулю N для задания периодических граничных условий. Клетки с именами $\varphi_1(p,q), \ldots, \varphi_k(p,q)$ называют *соседями*.

В общем случае *локальный оператор перехода* имеет вид

$$f_{\text{loc}} : \{S(p,q)\} \to \{S(p,q)\}.$$

После применения оператора перехода f_{loc} локальная конфигурация клетки с именем (p,q) изменится на $f_{\text{loc}}(S(p,q))$.

Применение локального оператора перехода ко всем клеткам КА принято называть *итерацией* или *применением соответствующего глобального оператора перехода*. Последовательный процесс применения итерации называется *эволюцией*.

В данной работе используются два локальных оператора перехода:

— f^1_{loc} — локальный оператор целочисленной диффузии моделирует перемещение особей по озеру; соответствующий глобальный оператор — f_1;
— f^2_{loc} — локальный оператор изменения численности моделирует процессы рождаемости, смертности, поедания и старения; соответствующий глобальный оператор — f_2.

Существуют два основных режима применения глобального оператора перехода — *синхронный* и *асинхронный*. Синхронный режим предполагает, что аргументы локального оператора перехода — состояния клеток на текущей итерации t. На каждой итерации вычисляются значения новых состояний всех клеток, затем все клетки одновременно меняют старые состояния на новые. При асинхронном режиме новое состояние клетки вычисляется от тех значений состояний клеток-соседей, которые на данный момент имеют место, и старое состояние клетки сразу меняется на новое. При этом клетка, обновляющая свое состояние, выбирается случайно [12].

В композиционной модели (1) использован более сложный, составной режим функционирования, комбинирующий синхронный и асинхронный режимы.

Режим функционирования ρ композиционной КА-модели (1) следующий: сначала глобальный оператор целочисленной диффузии f_1 несколько раз применяется асинхронно к каждому из подмножеств M^β_j независимо, затем асинхронно применяется глобальный оператор изменения численности f_2: случайно выбирается пара индексов (p, q) и оператор f_2 применяется одновременно ко всем клеткам-близнецам с именами $(p, q)^\beta_j$.

Далее подробно рассмотрим операторы перехода состояний.

2.1. Оператор целочисленной диффузии

Оператор f^1_{loc} целочисленной диффузии [13] применяется асинхронно к каждому из множеств M^β_j, представляющих различные группы организмов:

$$f^1_{\text{loc}} : \{S_1(p,q)\} \to \{S_1(p,q)\};$$
$$S_1(p,q) = \langle((p,q)^\beta_j, n), ((p-1,q)^\beta_j, n_1), ((p,q-1)^\beta_j, n_2), ((p,q+1)^\beta_j, n_3), ((p+1,q)^\beta_j, n_4)\rangle.$$

Применение целочисленной диффузии с коэффициентом $\sigma \in (0, 1)$ к клетке с именем (p, q) клеточного массива описывается следующим образом:

1) Пусть $(p_1, q_1), (p_2, q_2), (p_3, q_3), (p_4, q_4)$ — имена клеток — ближайших соседей клетки с именем (p, q).
2) Равновероятно выбирается клетка-сосед с именем (p_i, q_i), $i \in \{1, 2, 3, 4\}$.
3) Пусть n и n_i — состояния клеток с именами (p, q) и (p_i, q_i) соответственно.
4) Новые состояния n' и n'_i клеток с именами (p, q) и (p_i, q_i) вычисляются по формулам $n' = n - [\sigma \cdot n] + [\sigma \cdot n_i]$, $n'_i = n_i + [\sigma \cdot n] - [\sigma \cdot n_i]$.

Оператор целочисленной диффузии за одно применение оператора изменения численности применяется к каждому из множеств M^β_j разное количество раз. Пусть особь вида β возраста j за физическое время, соответствующее одному применению оператора изменения численности, преодолевает физическое расстояние L^β_j. Пусть c — физическое расстояние между центрами ближайших клеток клеточного массива. Тогда количество применений оператора целочисленной диффузии к множеству M^β_j за одно

применение оператора изменения численности вычисляется по формуле

$$K_{1j}^{\beta} = \frac{L_j^{\beta}}{c}. \tag{2}$$

2.2. Оператор изменения численности

Оператор изменения численности f_{loc}^2 применяется одновременно для всех клеток-близнецов:

$$f_{\text{loc}}^2 : \{S_2(p,q)\} \to \{S_2(p,q)\},$$
$$S_2(p,q) = \langle((p,q)_1^m, n_1^m), ((p,q)_2^m, n_2^m), ((p,q)_1^d, n_1^d), ((p,q)_2^d, n_2^d), ((p,q)_3^d, n_3^d),$$
$$((p,q)_1^b, n_1^b), ((p,q)_2^b, n_2^b), ((p,q)_3^b, n_3^b)\rangle.$$

Режим выполнения глобального оператора f_2: пара индексов (p,q), определяющая множество клеток-близнецов, выбирается случайно, и оператор f_{loc}^2 применяется одновременно ко всем клеткам-близнецам.

Новое значение клетки после применения оператора f_{loc}^2 будет равно (индексы, обозначающие вид и возрастную группу, опущены)

$$n' = \begin{cases} n + \lfloor \Delta n \rfloor, & \text{если random} < \text{frac}(\Delta n), \\ n + \lceil \Delta n \rceil & \text{иначе,} \end{cases}$$

где Δn — прирост численности особей; $\text{frac}(\Delta n)$ — дробная часть прироста численности особей. То есть с вероятностью $\text{frac}(\Delta n)$ численность особей изменится на $\lceil \Delta n \rceil$, а с вероятностью $(1 - \text{frac}(\Delta n))$ — на $\lfloor \Delta n \rfloor$.

Прирост численностей особей i-й возрастной группы вычисляется по формуле

$$\Delta n_i = (\rho_i n_j - \lambda_i n_i - \theta_i n_i)\Delta\tau,$$

где j — номер возрастной группы особей организмов, «порождающих» особей i-й возрастной группы; $\Delta\tau$ — физическое время, соответствующее одной итерации КА; $\rho_i n_j$ — прирост численности особей i-й возрастной группы за счет рождаемости в случае $i = 1$ или старения особей $(i-1)$-й возрастной группы в случае $i > 1$;

$$\rho_i = \begin{cases} \rho, & \text{если } i = 1, \\ \theta_{i-1} & \text{иначе;} \end{cases}$$

$\lambda_i n_i$ — число погибших особей от хищников или естественной смерти; $\theta_i n_i$ — число особей, перешедших в $(i+1)$-ю возрастную группу за счет старения.

Коэффициенты рождаемости ρ_1^m, ρ_1^d, ρ_1^b, старения θ_1^m, θ_2^m, θ_1^d, θ_2^d, θ_3^d, θ_1^b, θ_2^b, θ_3^b и смертности λ_1^m, λ_2^m, λ_1^d, λ_1^b взяты из работы [11].

В отличие от [11], в КА-модели учтена зависимость коэффициентов смертности хищников λ_2^d, λ_3^d, λ_2^b, λ_3^b от количества пищи путем введения коэффициентов относительного недостатка пищи $\xi_{ji}^{\beta\alpha}$. Относительное перенаселение особей вида α возраста i вычисляется по формуле

$$\chi_i^{\alpha} = \frac{n_i^{\alpha}}{n_i^{\alpha *}},$$

где $n_i^{\alpha *}$ — средняя численность по Байкалу [11].

Относительный недостаток пищи β возраста j для хищников α возраста i равен

$$\xi_{ji}^{\beta\alpha} = 1 - \frac{\chi_j^\beta}{\chi_i^\alpha}.$$

Если относительный недостаток $\xi_{ji}^{\beta\alpha}$ пищи больше нуля, то он с множителем c, зависящим от предпочтений в рационе хищника, добавляется к постоянному коэффициенту смертности λ^*.

Пусть

$$\gamma(\xi) = \begin{cases} \xi, & \xi > 0, \\ 0, & \xi \leqslant 0. \end{cases}$$

Таким образом, коэффициенты смертности с учетом их зависимости от количества пищи приняты равными

$$\lambda_2^b = \lambda_2^{b*} + \left(1 - \lambda_2^{b*}\right)\left(c_b \frac{\gamma(\xi_{12}^{mb}) + \gamma(\xi_{22}^{mb})}{2} + (1 - c_b)\frac{\gamma(\xi_{12}^{bb}) + \gamma(\xi_{12}^{db})}{2}\right);$$

$$\lambda_3^b = \lambda_3^{b*} + \left(1 - \lambda_3^{b*}\right)\left(c_b \frac{\gamma(\xi_{13}^{mb}) + \gamma(\xi_{23}^{mb})}{2} + (1 - c_b)\frac{\gamma(\xi_{13}^{bb}) + \gamma(\xi_{13}^{db})}{2}\right);$$

$$\lambda_2^d = \lambda_2^{d*} + \left(1 - \lambda_2^{d*}\right)\left(c_d \frac{\gamma(\xi_{12}^{md}) + \gamma(\xi_{22}^{md})}{2} + (1 - c_b)\frac{\gamma(\xi_{12}^{bd}) + \gamma(\xi_{12}^{dd})}{2}\right);$$

$$\lambda_3^d = \lambda_3^{d*} + \left(1 - \lambda_3^{d*}\right)\left(c_d \frac{\gamma(\xi_{13}^{md}) + \gamma(\xi_{23}^{md})}{2} + (1 - c_b)\frac{\gamma(\xi_{13}^{bd}) + \gamma(\xi_{13}^{dd})}{2}\right),$$

где λ_2^{b*}, λ_3^{b*}, λ_2^{d*}, λ_3^{d*} — коэффициенты смертности, а $c_b = 0{,}37$, $c_d = 0{,}07$ — пищевые предпочтения особей, принятые в [11]. То есть 37 % пищевого рациона больших голомянок составляют однолетки голомянок, 63 % — рачки макрогектопуса; 7 % пищевого рациона малых голомянок составляют однолетки голомянок, 93 % — рачки макрогектопуса. В коэффициенте смертности неполовозрелых больших голомянок λ_2^b термы имеют следующие значения:

1) $\left(1 - \lambda_2^{b*}\right)$ — доля остающихся в живых рыб от действия факторов, учтенных в постоянном коэффициенте смертности λ_2^{b*};
2) $\left(\gamma(\xi_{12}^{mb}) + \gamma(\xi_{22}^{mb})\right)/2$ — относительный недостаток рачков макрогектопуса;
3) $\left(\gamma(\xi_{12}^{bb}) + \gamma(\xi_{12}^{db})\right)/2$ — относительный недостаток однолеток голомянок.

3. Вычислительные эксперименты

Для моделирования была написана программа на языке C++ с использованием библиотек Qt и OpenGL.

Для проведения вычислительных экспериментов принята следующая оценка параметров моделирования. Можно считать размеры физической области равными 40×40 км. Модельная область представима квадратной сеткой размера 200×200. Физическое расстояние между центрами соседних клеток равно 200 м. Физическое время $\Delta\tau$, соответствующее одной итерации КА, принято равным двум дням. Принято, что за время $\Delta\tau$ с учетом вертикальных миграций и случайности перемещения особи пересекают следующие расстояния (в километрах):

$$L_1^m = 0{,}2, \quad L_2^m = 0{,}4, \quad L_1^d = 1, \quad L_2^d = 2, \quad L_3^d = 4, \quad L_1^b = 2, \quad L_2^b = 3, \quad L_3^b = 5.$$

По соотношению (2) определяется число применений оператора диффузии для видов на одно применение оператора изменения численности:

$$K_{11}^m = 1, \quad K_{12}^m = 2, \quad K_{11}^d = 5, \quad K_{12}^d = 10, \quad K_{13}^d = 20, \quad K_{11}^b = 10, \quad K_{12}^b = 15, \quad K_{13}^b = 25.$$

Из соображений, что на одну клетку КА приходится ровно 4 соседних клетки, значения коэффициентов целочисленной диффузии выбраны равными

$$\sigma_1^m = \sigma_2^m = \sigma_1^d = \sigma_2^d = \sigma_3^d = \sigma_1^b = \sigma_2^b = \sigma_3^b = 0{,}25.$$

3.1. Первый эксперимент. Косяк рыб

Цель эксперимента — исследовать эволюцию КА-модели в случае начального неравномерного распределения особей по области моделирования.

В начальном состоянии число особей половозрелых и неполовозрелых рачков макрогектопуса выбирается случайным в пределах 30 % отклонения от среднего по Байкалу [11]. Число особей малой и большой голомянок выбирается аналогичным образом, за исключением центральной вертикальной полосы шириной 1/3 от ширины области, где число особей голомянок увеличено в 2 раза (рис. 2).

Эксперимент описывает поведение экосистемы в случае, если в озеро попадает косяк хищников.

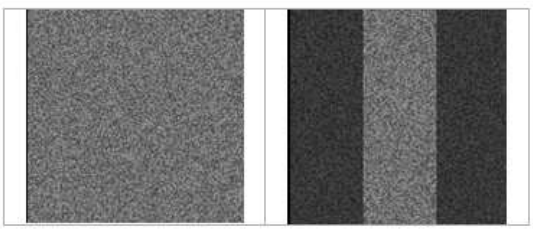


Рис. 2. Начальное глобальное состояние для всех возрастных групп рачков макрогектопуса (слева) и всех возрастных групп малой и большой голомянки (справа). Более светлый цвет означает бóльшую численность особей

В результате колебаний численности система приходит к устойчивому состоянию — равномерному распределению особей по области. На рис. 3 приведена эволюция неполовозрелых особей макрогектопуса и малой голомянки.

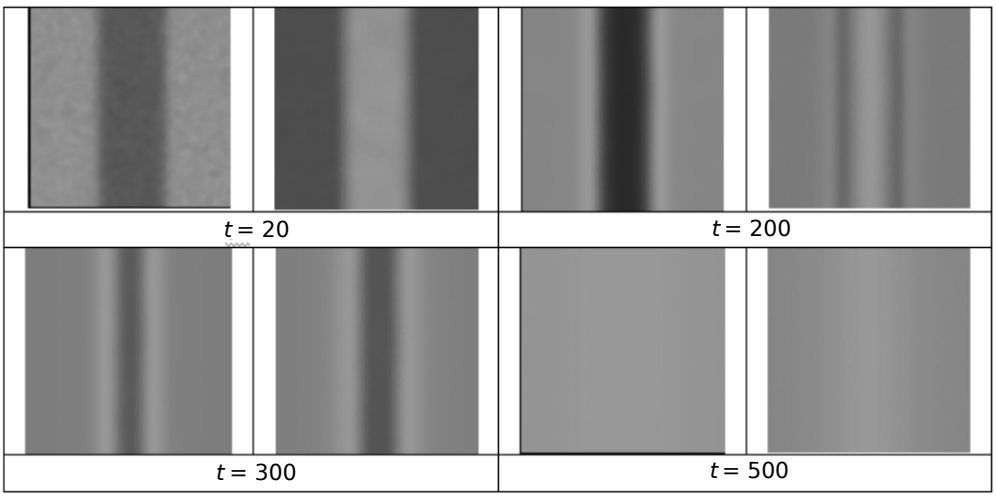


Рис. 3. Некоторые глобальные состояния модели на итерациях $t = 20, 200, 300, 500$: слева — неполовозрелые особи макрогектопуса; справа — неполовозрелые особи малой голомянки

Динамика численности неполовозрелых особей макрогектопуса и голомянок в центре области (где первоначально был косяк рыб) и на её левой границе (вдалеке от косяка) приведена на рис. 4. В точке на границе области изменения численности особей незначительны и влияние косяка практически незаметно. Например, численность неполовозрелых особей рачка макрогектопуса изменяется в пределах 7 %. В центре области изменения численности особей гораздо сильнее. При избытке хищников в 2 раза уменьшается численность рачков, и вследствие недостатка пищи уменьшается численность рыб. После смерти большей части рыб численность рачков вырастает в 5 раз. Вслед за ростом численности рачков растет и численность рыб. Далее система приходит к устойчивому состоянию.

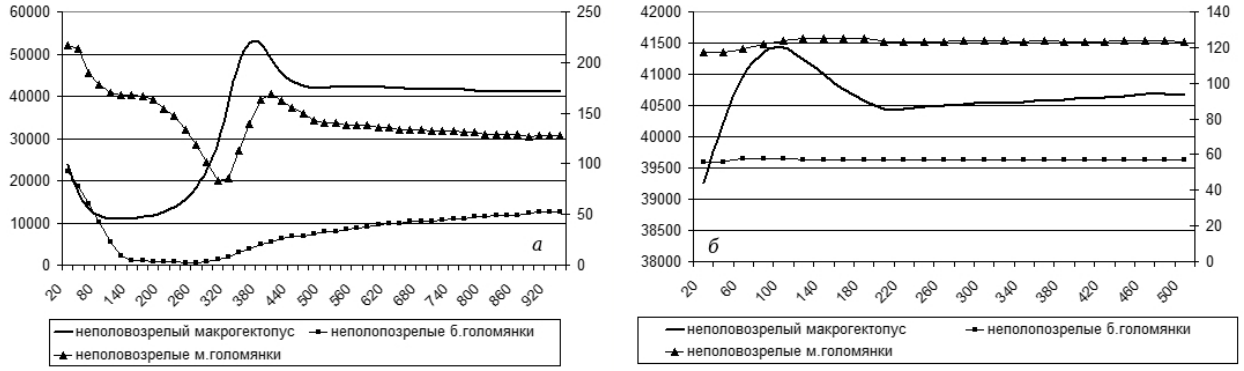


Рис. 4. Динамика численности некоторых видов в точке в центре области (*а*) и в точке на левой границе (*б*). Ось абсцисс — номер итерации, ось ординат — средняя численность на клетку. По левой оси ординат отмечена численность рачков, по правой — голомянок

Устойчивое состояние — равномерное распределение особей по области моделирования. Отношения общей численности особей в устойчивом состоянии составляют

$$\frac{\text{макрогектопус}}{\text{малая голомянка}} = 5{,}92; \qquad \frac{\text{макрогектопус}}{\text{большая голомянка}} = 24{,}06. \tag{3}$$

Таким образом, при начальном неравномерном распределении особей по области моделирования модель приходит к устойчивому состоянию, представляющему собой равномерное распределение особей по озеру. В устойчивом состоянии колебаний не наблюдается.

3.2. Второй эксперимент. Загрязнение

В этом эксперименте введена зависимость коэффициентов смертности от точки в области моделирования. Цель эксперимента — исследовать эволюцию КА-модели в случае неравномерного распределения коэффициентов взаимодействия между группами организмов. Эксперименту может соответствовать загрязнение определенной территории озера.

Начальные состояния клеток для всех групп организмов — равномерное распределение по области моделирования со случайным отклонением в каждой клетке от устойчивого состояния в пределах 30 % (рис. 5).

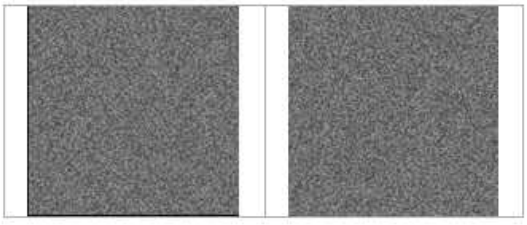


Рис. 5. Начальное состояние КА для неполовозрелых рачков макрогектопуса (слева) и неполовозрелых особей большой голомянки (справа)

Пусть «загрязнение» действует перманентно в круге радиуса 200/3 с координатами центра (100, 100) (размеры области моделирования 200×200). Пусть загрязнение увеличивает смертность рыб. Коэффициенты смертности голомянок увеличены на 30 % в эпицентре загрязнения и линейно убывают при отдалении от эпицентра к краю области загрязнения вплоть до их значений в первом эксперименте. «Загрязнение» действует на протяжении всего эксперимента.

Некоторые состояния КА для неполовозрелых рачков макрогектопуса представлены на рис. 6.

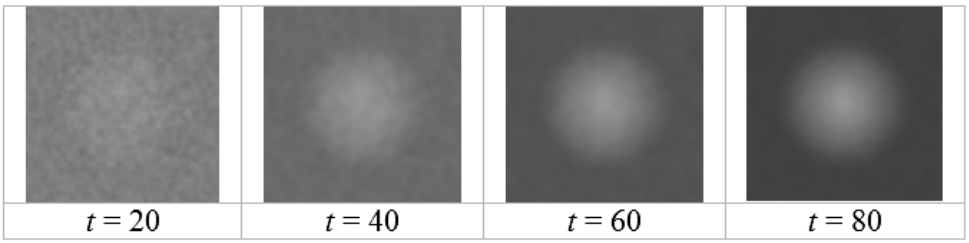


Рис. 6. Некоторые итерации для неполовозрелых особей макрогектопуса

Динамика численности неполовозрелых особей макрогектопуса и голомянок в центре «загрязнения» и на краю области приведена на рис. 7. В точке на границе области, как и в первом эксперименте, изменения численности особей незначительны и влияние загрязнения практически незаметно. На границе области численности особей изменяются в пределах 7 % от первоначальных значений. В центре области происходят более сильные изменения численности. Так, сперва происходит скачок численности неполовозрелых рачков макрогектопуса на 75 % от начального состояния в силу уменьшения числа рыб. Затем в силу большего числа пищи растет число хищников-рыб и соответственно уменьшается число рачков. Далее система приходит к устойчивому состоянию. Несмотря на то, что «загрязнение» действует на всех рыб, в устойчивом состоянии число малых голомянок окажется больше, чем в устойчивом состоянии при отсутствии «загрязнения», а число больших голомянок — меньше. Это явление объясняется тем, что большие голомянки больше предпочитают молодь голомянок в качестве пищи. Поэтому при уменьшении количества рыб большие голомянки гибнут активнее малых, и меньшее количество молоди голомянок оказывается съедено. При уменьшении количества рыб также увеличивается численность рачков, что в совокупности создаёт благоприятные условия для малых голомянок.

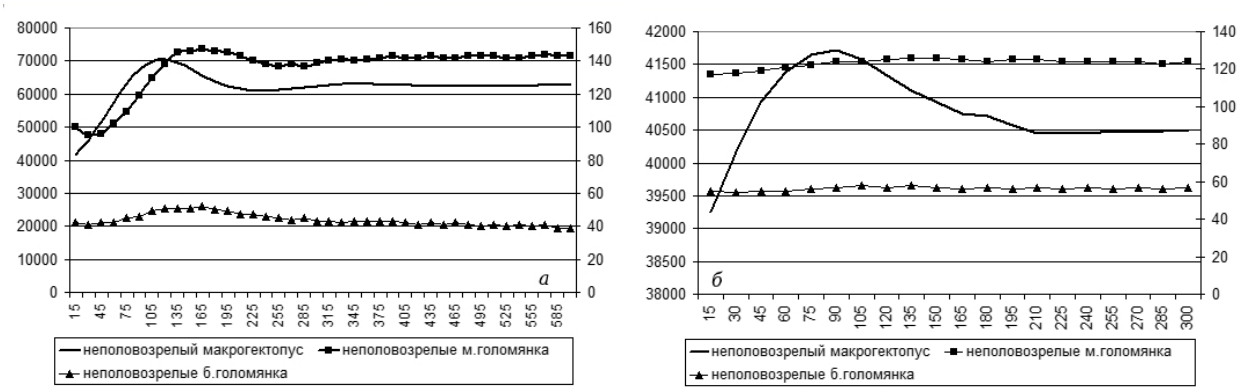


Рис. 7. Динамика численности организмов в центре «загрязнения» ($а$) и в клетке на границе области моделирвания ($б$). Ось абсцисс — номер итерации, ось ординат — число особей в клетке. Численность рачков отложена по левой шкале, численность голомянок — по правой

Устойчивое состояние не является равномерным. На краю области, где коэффициенты смертности всех видов организмов совпадают с коэффициентами в первом эксперименте, в устойчивом состоянии соотношения численностей близки к результатам первого эксперимента (3):

$$\frac{\text{макрогектопус}}{\text{малая голомянка}} = 6{,}01, \qquad \frac{\text{макрогектопус}}{\text{большая голомянка}} = 22{,}62.$$

В эпицентре загрязнения соотношения численностей особей в устойчивом состоянии составляют

$$\frac{\text{макрогектопус}}{\text{малая голомянка}} = 6{,}67, \qquad \frac{\text{макрогектопус}}{\text{большая голомянка}} = 43{,}90.$$

Таким образом, устойчивое состояние не является однородным. В устойчивом состоянии колебаний не наблюдается. В области загрязнения снижена численность больших голомянок и повышена численность малых голомянок и рачков по сравнению с результатами первого эксперимента; за пределами области загрязнения устойчивое состояние совпадает с результатом первого эксперимента.

Заключение

Предложена КА-модель динамики численности организмов озера Байкал, позволяющая учитывать особенности пространственного распределения организмов и пространственной зависимости внешних условий, а также зависимость смертности хищников от количества пищи.

Компьютерное моделирование показало, что при условии пространственной независимости коэффициентов взаимодействия между группами организмов устойчивым состоянием является равномерное распределение особей по области моделирования. Если коэффициенты взаимодействия зависят от точки области моделирования, то устойчивое состояние неоднородно.

В отличие от модели, в реальности наблюдаются сезонные колебания численности организмов в озере Байкал. В КА-модели не отражена зависимость коэффициента рождаемости и смертности по сезонам, поэтому КА-модель приходит к устойчивому состоянию. В дальнейшем для развития КА-модели планируется ввести зависимость коэффициентов рождаемости и смертности по сезонам.

ЛИТЕРАТУРА

1. *Хакен Г.* Синергетика. М.: Мир, 1980. 406 с.
2. *Wolfram S.* A new kind of science. USA: Wolfram Media Inc., 2002. 1197 p.
3. *Chua L. O.* CNN: A Paradigm for Complexity. Berkely: World Scientific Series on Nonlinear Science. University of California, 1998. 320 p.
4. *Ванаг В. К.* Диссипативные структуры в реакционно-диффузионных системах. Ижевск: ИКИ, 2008. 300 с.
5. *Madore B. and Freedman W.* Computer simulation of the Belousov — Zhabotinski reaction // Science. 1983. V. 222. P. 615–618.
6. *Bandman O.* Simulation of complex phenomena by Cellular Automata composition // Ibid. P. 9–20.
7. *Афанасьев И.* Исследование эволюции клеточных автоматов, моделирующих процесс «разделения фаз» на треугольной сетке // Прикладная дискретная математика. 2010. № 4. С. 79–90.
8. *Bandman O.* Parallel Composition of Asynchronous Cellular Automata Simulating Reaction Diffusion Processes // LNCS. 2010. V. 6350. P. 395–398.
9. *Свирежев Ю. М., Логофет Д. О.* Устойчивость биологических сообществ. М.: Наука, 1978. 352 с.
10. *Базыкин А. Д.* Нелинейная динамика взаимодействующих популяций. Ижевск: ИКИ, 2003. 368 с.
11. *Зоркальцев В. И., Казазаева А. В., Мокрый И. В* Модель взаимодействия трех пелагических видов организмов озера Байкал // Современные технологии. Системный анализ. Моделирование. Иркутский государственный университет путей сообщения. 2008. № 1. С. 182–193.
12. *Бандман О. Л.* Клеточно-автоматные модели пространственной динамики // Системная информатика. 2006. № 10. С. 58–113.
13. *Medvedev Y. G.* Multi-particle Cellular Automata Models For Diffusion Simulation // Methods and tools of parallel programming multicomputers. 2011. V. 6083/2011. P. 204–211.

СВЕДЕНИЯ ОБ АВТОРАХ

АБРОСИМОВ Михаил Борисович — доцент, кандидат физико-математических наук, доцент Саратовского государственного университета им. Н. Г. Чернышевского, г. Саратов. E-mail: **mic@rambler.ru**

АЛЕКСЕЙЧУК Антон Николаевич — доктор технических наук, профессор кафедры Института специальной связи и защиты информации Национального технического университета Украины «Киевский политехнический институт», г. Киев.
E-mail: **alex-crypto@mail.ru**

АНОХИН Михаил Игоревич — кандидат физико-математических наук, старший научный сотрудник Института проблем информационной безопасности Московского государственного университета им. М. В. Ломоносова, г. Москва.
E-mail: **anokhin@mccme.ru**

АФАНАСЬЕВ Иван Владимирович — аспирант Института вычислительной математики и математической геофизики СО РАН, г. Новосибирск.
E-mail: **ivafanas@gmail.com**

ДЕВЯНИН Петр Николаевич — доктор технических наук, доцент, декан факультета Института криптографии, связи и информатики, г. Москва.
E-mail: **peter_devyanin@hotmail.com**

КОГОС Константин Григорьевич — студент кафедры криптологии и дискретной математики Национального исследовательского ядерного университета (МИФИ), г. Москва. E-mail: **k.kogos@mail.ru**

КОНЮШОК Сергей Николаевич — кандидат технических наук, доцент кафедры Института специальной связи и защиты информации Национального технического университета Украины «Киевский политехнический институт», г. Киев.
E-mail: **3tooth@mail.ru**

КРУЧИНИН Дмитрий Владимирович — аспирант Томского государственного университета систем управления и радиоэлектроники, г. Томск.
E-mail: **KruchininDm@gmail.com**

СОЛОВЬЁВ Тимофей Михайлович — студент Национального исследовательского Томского государственного университета, г. Томск. E-mail: **tm.solovev@gmail.com**

СТЕФАНЦОВ Дмитрий Александрович — аспирант Национального исследовательского Томского государственного университета, г. Томск.
E-mail: **d.a.stefantsov@isc.tsu.ru**

ФОМИЧЕВ Владимир Михайлович — старший научный сотрудник, доцент, доктор физико-математических наук, профессор кафедры криптологии и дискретной математики Национального исследовательского ядерного университета (МИФИ), г. Москва. E-mail: **fomichev@nm.ru**

ЧЕРНЯК Роман Игоревич — студент Национального исследовательского Томского государственного университета, г. Томск. E-mail: **r.chernyack@gmail.com**

АННОТАЦИИ СТАТЕЙ НА АНГЛИЙСКОМ ЯЗЫКЕ

Alekseychuk A. N., Konyushok S. N. **NONLINEARITY STATISTICAL PROPERTIES OF BOOLEAN FUNCTION RESTRICTIONS ON A RANDOMLY CHOSEN SUBSPACE.** It is shown that for all sufficiently large natural n, the relative nonlinearity of any Boolean function in n variables can be statistically approximated by the relative nonlinearity of its restriction on a random subspace (possibly without the zero vector), whose dimension is independent on n.

Keywords: *Boolean functions, nonlinearity, random subspace, statistical estimation.*

Anokhin M. I. **THEORY OF COMPLETE ORTHOGONAL DIRECT DECOMPOSITIONS OF VECTOR SPACES.** A theory is constructed for the complete orthogonal (with respect to generalized orthogonalities defined by certain partial symmetric bilinear functions) direct decompositions of vector spaces V such that the quotient of V by a certain particular subspace is finite-dimensional. The main result of the theory is a description of all such decompositions. This theory has an application to the theory of direct decompositions of p-ary functions, where p is a prime.

Keywords: *vector space, orthogonality, partial symmetric bilinear function, complete decomposition, Abelian group.*

Kogos K. G., Fomichev V. M. **ABOUT BRANCHINGS OF CRYPTOGRAPHIC FUNCTIONS ON TRANSFORMATIONS WITH THE PRESCRIBED SIGN.** The conceptions connected with a branching of functions into linear subfunctions is generalized for branchings of special kind functions into the transformations having a prescribed sign. Characteristic estimations for branchings of the functions realized by the generators "δ–τ steps" and with alternating step are obtained.

Keywords: *branching degree, sign in semigroup, "δ–τ steps" generator, generator with alternating step.*

Kruchinin D. V. **PROPERTIES OF COEFFICIENTS IN SOME SUPERPOSITIONS OF GENERATING FUNCTIONS.** The generating function $\ln\left((1-F(x))^{-1}\right)$ where $F(x)$ is an ordinary generating function with the integer coefficients is considered. Some properties ot its coefficients allowing the construction of probabilistic primality tests are obtained. The connection of them with the existing primality tests is shown. Some new properties of Lucas numbers and binomial coefficients $\binom{2n-1}{n-1}$ are obtained too.

Keywords: *generating functions, superposition of generating functions, composition of a natural number, primality test.*

Solovyev T. M., Chernyak R. I. **FINGERPRINTS RESISTANT TO COLLUSION ATTACKS.** The systems of fingerprints and collusion attacks on them are considered. The structure of fingerprint systems opposing such attacks is investigated. An algorithm for generating these systems is developed. A method for identifying counter-fitters through a fake fingerprint is proposed. In the set of all collective attacks, the riskiest one being the majorizing attack is identified.

Keywords: *digital watermarking, fingerprinting, collusion attack.*

Devyanin P. N. **THE ROLE DP-MODEL OF ACCESS AND INFORMATION FLOWS CONTROL IN OPERATING SYSTEMS OF *Linux* SETS.** In this article, the basic role DP-model of access and information flows control in operating systems (OS) is presented completing the role DP-model for OS of Linux set. New features of the basic role DP-model described in the article are the following: the names of entities, mandatory integrity attributes of entities-containers, and the function of de-facto ownership. The main difference of the model is the strong separation of the de-jure state transformation rules (requiring implementation in OS) and the de-facto rules (used only for the analysis of system security conditions). It is proved that the using only monotonic state transformation rules is sufficient for analysing conditions of transfering role access rights, of access to entities, and of realizing information flows in OS.
Keywords: *computer security, role DP-model, operating system Linux.*

Stefantsov D. A. **THE AspectTalk PROGRAMMING LANGUAGE.** The object-oriented (OOP) and aspect-oriented (AOP) programming language AspectTalk is described. The language consists of base language, metalanguage, and libraries for OOP and AOP. Properties of AOP in it are provided by metaprogramming, metaobject protocols and the mixins mechanism. The brief comparison of AspectTalk with similar programming languages is given.
Keywords: *programming language, metaprogramming, Smalltalk, AOP, OOP, metaobject protocol.*

Abrosimov M. B. **CHARACTERIZATION OF GRAPHS WITH A GIVEN NUMBER OF ADDITIONAL EDGES IN A MINIMAL 1-VERTEX EXTENSION.** A graph G^* is k-vertex extension of graph G if every graph obtained by removing any k vertices from G^* contains G. k-Vertex extension of graph G with $n + k$ vertices is called minimal if, among all k-vertex extensions of graph G with $n+k$ vertices, it has the minimum possible number of edges. The graphs whose minimal vertex 1-extensions have a specified number of additional edges are studied. A solution is given when the number of additional edges is equal to one, two or three.
Keywords: *graph, minimal vertex extension, exact vertex extension, fault tolerance.*

Afanasyev I. V. **A CELLULAR AUTOMATA MODEL FOR THE DYNAMICS OF ORGANISMS POPULATION IN BAIKAL.** The cellular automata model of Baikal organisms population dynamics is proposed and investigated. The results of computational experiments for the cases of predators gathering and area pollution are presented.
Keywords: *self-organization, cellular automata, composition, population dynamic model, prey — predator system.*

Журнал «Прикладная дискретная математика» включен в перечень ВАК рецензируемых российских журналов, в которых должны быть опубликованы основные результаты диссертаций, представляемых на соискание учёной степени кандидата и доктора наук, а также в перечень журналов, рекомендованных УМО в области информационной безопасности РФ в качестве учебной литературы по специальности «Компьютерная безопасность».

Журнал «Прикладная дискретная математика» распространяется по подписке; его подписной индекс 38696 в объединённом каталоге «Пресса России». Полнотекстовые электронные версии вышедших номеров журнала доступны на его сайте vestnik.tsu.ru/pdm и на Общероссийском математическом портале www.mathnet.ru. На сайте журнала можно найти также и правила подготовки рукописей статей в журнал.

Тематика публикаций журнала:

- *Теоретические основы прикладной дискретной математики*
- *Математические методы криптографии*
- *Математические методы стеганографии*
- *Математические основы компьютерной безопасности*
- *Математические основы надежности вычислительных и управляющих систем*
- *Прикладная теория кодирования*
- *Прикладная теория автоматов*
- *Прикладная теория графов*
- *Логическое проектирование дискретных автоматов*
- *Математические основы информатики и программирования*
- *Вычислительные методы в дискретной математике*
- *Дискретные модели реальных процессов*
- *Математические основы интеллектуальных систем*
- *Исторические очерки по дискретной математике и ее приложениям*