Московский Государственный Университет
им. М.В. Ломоносова
Российская Академия Наук
Академия Технологических Наук России
Российская Академия Естественных Наук

Интеллектуальные Системы.

Теория и приложения

ТОМ **21** ВЫПУСК **3** * **2017**

МОСКВА

ISSN 2411–4448

УДК 519.95; 007:159.955
ББК 32.81

Издается с 1996 г.*

В журнале «Интеллектуальные системы. Теория и приложения» публикуются научные достижения в области теории и приложений интеллектуальных систем, новых информационных технологий и компьютерных наук.

Издание журнала осуществляется под эгидой МГУ им. М. В. Ломоносова, Научного Совета по комплексной проблеме «Кибернетика» РАН, Отделения «Математическое моделирование технологических процессов» АТН РФ, Секции «Информатики и кибернетики» РАЕН.

Учредитель журнала: ООО «Интеллектуальные системы».

Журнал входит в список изданий, включенных ВАК РФ в реестр публикаций материалов по кандидатским и докторским диссертациям по математике и механике.

Спонсором издания является:

ООО «Два Облака»
Разработка корпоративных информационных систем
http://www.dvaoblaka.ru

Индекс подписки на журнал: 64559 в каталоге НТИ «Роспечать».

Адрес редакции: 119899, Россия, Москва, Воробьевы Горы, МГУ, ГЗ, механико-математический факультет, комн. 12-01.
Адрес издателя: 115230, Россия, Москва, Хлебозаводский проезд, д. 7, стр. 9, офис 9. Тел. +7 (495) 939-46-37, e-mail: mail@intsysjournal.org

*) Прежнее название журнала: «Интеллектуальные системы».

ОГЛАВЛЕНИЕ

Часть 1.
Общие проблемы теории интеллектуальных систем

О некоторых результатах теории алгебраической сложности

Алексеев В.Б.

В данной работе приведен обзор некоторых результатов о вычислительной сложности алгебр, в частности, результатов, полученных на кафедре математической кибернетики МГУ им. М.В. Ломоносова автором и его учениками: Поспеловым А.Д., Чокаевым Б.В., Лысиковым В.В.

Ключевые слова: алгебраическая сложность, алгебра, ранг алгебры, билинейная сложность, мультипликативная сложность, сложность умножения матриц.

Понятие алгебраической сложности связано с алгебраическими моделями вычислений. Входными элементами в таких моделях являются переменные и константы — элементы какой-нибудь алгебраической структуры, обычно, элементы кольца или поля. В качестве элементарных операций в этом случае рассматриваются 4 алгебраические операции (сложение, вычитание, умножение, деление). Каждая операция может применяться к входным элементам или к уже построенным выражениям. Задача состоит в построении семейства заданных алгебраических выражений. Сложностью вычисления (алгоритма) называют число примененных операций (аналог сложности схем из функциональных элементов). Сложностью задачи называют минимум сложности алгоритмов, вычисляющих заданное семейство выражений. Хорошим введением в алгебраическую теорию сложности может служить книга [1]. Одна из центральных задач в алгебраической теории сложности — сложность умножения (вычисления произведений) в алгебрах. Среди важнейших задач — сложность умножения матриц и полиномов.

Пусть $\|a_{ij}\|_{m\times n}$ обозначает матрицу размера $m \times n$ над некоторым кольцом. Задача умножения матрицы $\|a_{ij}\|_{m\times n}$ на матрицу $\|b_{kl}\|_{n\times p}$ — это задача вычисления системы из mp билинейных форм вида $\sum_{j=1}^{n} a_{ij}b_{jl}$. При этом элементы a_{ij} и b_{kl} рассматриваются как отдельные независимые входные переменные, а на каждом шаге вычисления

разрешается применить любую из 4 арифметических операций к уже построенным выражениям и элементам кольца. Число арифметических операций в алгоритме называют арифметической сложностью (обычно просто сложностью) алгоритма, а наименьшую сложность алгоритмов, вычисляющих систему билинейных форм $\sum_{j=1}^{n} a_{ij}b_{jl}$, называют сложностью задачи умножения матрицы размера $m \times n$ на матрицу размера $n \times p$. Стандартный алгоритм («строка на столбец») для умножения матриц размера $n \times n$ использует порядка n^3 арифметических операций. Первый асимптотически более быстрый алгоритм умножения матриц размера $n \times n$ (с числом арифметических операций $O\left(n^{\log_2 7}\right)$) построил Ф. Штрассен [2] в 1969 году. В последующие 20 лет верхняя оценка сложности умножения двух матриц размера $n \times n$ была понижена до $O\left(n^{2.38}\right)$ [3], но с тех пор существенных продвижений в этой задаче нет.

Чтобы лучше понять проблемы, возникающие при поиске быстрых алгебраических вычислений, математики начали рассматривать более общую задачу — вычисления в произвольных алгебрах. Напомним, что алгебра — это линейное пространство, на котором задана операция умножения, обладающая свойством линейности по каждому из сомножителей. С учетом линейности, для задания алгебры достаточно в рассматриваемом линейном пространстве рассмотреть какой-нибудь базис $e_1, \ldots, e_n$ и задать произведения базисных элементов: $e_i \cdot e_j = \sum_{k=1}^{n} t_{ijk}e_k$. Тогда произведение произвольных элементов $\sum_{i=1}^{n} a_i e_i$ и $\sum_{j=1}^{n} b_j e_j$ будет вычисляться по формуле:

$$\left(\sum_{i=1}^{n} a_i e_i\right) \cdot \left(\sum_{j=1}^{n} b_j e_j\right) = \sum_{k=1}^{n} \left(\sum_{i=1}^{n}\sum_{j=1}^{n} t_{ijk} a_i b_j\right) e_k.$$

Таким образом, если сомножители задаются коэффициентами при разложении по некоторому базису, то коэффициенты произведения являются билинейными формами от коэффициентов сомножителей. И мы приходим к важному классу задач в теории алгебраической сложности — сложности вычисления систем билинейных форм. В рекурсивных алгоритмах для умножения матриц (таким является алгоритм Штрассена) элементы a_{ij} и b_{kl} сами могут являться матрицами, которые не коммутируют между собой. При этом основную роль для оценки сложности рекурсивных алгоритмов играет число умножений. В связи с этим важное значение придается изучению билинейной сложности умножения матриц, а вместе с этим и изучению билинейной сложности умножения в произвольных алгебрах.

Пусть U, V, W — конечномерные линейные пространства над некоторым полем F.

Определение 1. Отображение $\varphi\colon U \times V \to W$ называется *билинейным*, если

$$\varphi(a_1x_1 + a_2x_2, y) = a_1\varphi(x_1, y) + a_2\varphi(x_2, y),$$
$$\varphi(x, a_1y_1 + a_2y_2) = a_1\varphi(x, y_1) + a_2\varphi(x, y_2).$$

Если $u_1, u_2, \ldots, u_n$ — базис U, $v_1, v_2, \ldots, v_m$ — базис V, $w_1, w_2, \ldots, w_l$ — базис W, то билинейное отображение задается набором коэффициентов t_{ijk}:

$$\varphi(\sum_{i=1}^{n} a_iu_i, \sum_{j=1}^{m} b_jv_j) = \sum_{k=1}^{l} c_kw_k \Leftrightarrow c_k = \sum_{i=1}^{n}\sum_{j=1}^{m} t_{ijk}a_ib_j.$$

Определение 2. *Билинейным алгоритмом сложности* r для вычисления билинейного отображения φ называется набор r троек f_s, g_s, z_s, где $f_s \in U^*$, $g_s \in V^*$, $z_s \in W$, такой, что

$$\varphi(x, y) = \sum_{s=1}^{r} f_s(x)g_s(y)z_s$$

(Здесь U^*, V^* — пространства, двойственные к U и V, то есть f_s и g_s — это линейные формы на U и V.)

Минимально возможная сложность билинейного алгоритма называется *билинейной сложностью* или *рангом* φ (обозначение: $R(\varphi)$).

Определение 3. *Квадратичным алгоритмом сложности* r для вычисления φ называется набор r троек f_s, g_s, z_s, где $f_s \in (U \times V)^*$, $g_s \in (U \times V)^*$, $z_s \in W$, такой, что

$$\varphi(x, y) = \sum_{s=1}^{r} f_s(x, y)g_s(x, y)z_s.$$

(Здесь все линейные формы f_s и g_s могут зависеть как от x, так и от y.)

Минимально возможная сложность квадратичного алгоритма называется *мультипликативной сложностью* φ (обозначение: $C(\varphi)$).

Нетрудно получить следующие неравенства для любого билинейного отображения (семейства билинейных форм) φ:

$$rg(\varphi) \le R(\varphi) \le dimX \cdot dimY,$$

$$C(\varphi) \le R(\varphi) \le 2C(\varphi),$$

где $rg(\varphi)$ — размерность образа при отображении φ (число линейно независимых билинейных форм в заданной системе φ).

Существуют семейства билинейных форм P такие, что

$$C(P) < R(P).$$

Один из простейших нетривиальных примеров — задача умножения матрицы размера 2×2 на матрицу размера 2×3. Обозначим семейство из 6 билинейных форм в этой задаче как $< 2, 2, 3 >$. Тогда

$$C(< 2, 2, 3 >) = 10 \text{ (Waksman A., 1970, [4])},$$

$$R(< 2, 2, 3 >) = 11 \text{ (Алексеев В.Б., 1985 [5])}.$$

Для любых семейств билинейных форм P и Q выполняются следующие неравенства для ранга и мультипликативной сложности:

$$R(P \cup Q) \leq R(P) + R(Q),$$

$$C(P \cup Q) \leq C(P) + C(Q).$$

Известна следующая гипотеза.

Гипотеза о прямой сумме. *Если в семействах билинейных форм P и Q нет общих переменных, то*

$$R(P \cup Q) = R(P) + R(Q).$$

Фактически, Штрассен установил, что билинейная сложность умножения двух квадратных матриц порядка 2 не превосходит 7, откуда вытекал и его общий результат.

Установить точное значение билинейной сложности редко удается даже в задачах перемножения двух матриц достаточно малого размера. Например, для задачи перемножения двух матриц размера 3×3 к настоящему моменту известно только, что билинейная сложность заключена между 19 и 23 [6, 7]. Для задачи перемножения двух матриц размера 4×4 верхняя оценка 49 на число умножений (вместо обычных 64) получается двукратным использованием алгоритма Штрассена, и эта оценка пока не понижена. Для задачи перемножения двух матриц размера 5×5 наилучшим остается алгоритм из [8] с числом умножений 100 вместо обычных 125. Из недавних результатов интересен результат А.В. Смирнова [9], который построил билинейный алгоритм для умножения матрицы размера 3×3 на матрицу размера 3×6 с 40 умножениями (вместо обычных 54).

Обозначим через $R_F < m, n, p >$ билинейную сложность задачи умножения матрицы размера $m \times n$ на матрицу размера $n \times p$ над некоторым полем F. Теорема о двойственности [10] утверждает, что $R_F < m, n, p >$ не изменяется при любой перестановке чисел m, n, p.

Нетрудно показать, что $R_F < m, 1, p >= mp$. В работе [11] показано, что $R_F < 2, 2, 2 >= 7$ для любого поля F. (При этом в работе [12] доказано, что все билинейные алгоритмы билинейной сложности 7 для умножения матриц порядка 2 в определенном смысле эквивалентны друг другу). Применяя результат Штрассена, можно несложно получить, что $R_F < m, 2, 2 >\leq \lceil \frac{7m}{2} \rceil$ для произвольного поля F. В работе [13] получен более общий результат:

$$R_F(< 2, m, n >) \leq \lceil \frac{3mn + max(m, n)}{2} \rceil.$$

Там же получена и такая же нижняя оценка, но только для поля из 2 элементов. Автором в работе [5] был рассмотрен случай $m = 3$ и доказано, что $R_F < 3, 2, 2 >= 11$ для произвольного поля F. В статье [14] доказано, что $R_F < 4, 2, 2 >= 14$ для произвольного поля F. Пока только для этих параметров и двойственных к ним установлено точное значение для $R_F < m, n, p >$ над произвольным полем F.

В работе [15] рассмотрена величина $R_F < 5, 2, 2 >$, для которой получена нижняя оценка $R_F < 5, 2, 2 > \geq 17$ над произвольным полем F. (Отметим, что наилучшая известная верхняя оценка для этой задачи равна 18.) В работе [16] этот результат обобщен: а именно, показано, что $R_F < m, 2, 2 > \geq 3m + 2$ над произвольным полем F для всех $m \geq 3$.

Одним из первых общих результатов о нижних оценках сложности вычислений в алгебрах явилась теорема Алдера - Штрассена [17]. Напомним, что двухсторонним идеалом в алгебре A называется подпространство, замкнутое относительно умножения на любой элемент из A как слева, так и справа. Максимальный двухсторонний идеал в алгебре A — это двухсторонний идеал, отличный от A и не содержащийся ни в каком другом двухстороннем идеале, кроме A.

Теорема 1. *(Alder A., Strassen V., 1981.) Для ранга умножения в ассоциативной алгебре A с единицей справедлива нижняя оценка*

$$R(A) \geqslant 2 \dim A - t(A),$$

где $t(A)$ — количество максимальных двухсторонних идеалов в A.

Поскольку в алгебре матриц порядка n ровно один двухсторонний идеал, отличный от A (нулевая матрица), то для билинейной сложности умножения в алгебре матриц порядка n теорема Алдера - Штрассена дает нижнюю оценку:

$$R(< n, n, n >) \geq 2n^2 - 1.$$

Теорема Алдера - Штрассена породила следующие новые понятия.

Определение 4. Ассоциативная алгебра A с единицей называется *алгеброй минимального ранга*, если $R(A) = 2\dim A - t(A)$, и алгеброй *почти минимального ранга*, если $R(A) = 2\dim A - t(A) + 1$, где $t(A)$ — количество максимальных двухсторонних идеалов в A.

Одна из задач, которая при этом возникла, — описать все алгебры минимального ранга. Вскоре эту задачу удалось решить для локальных алгебр. *Локальная* алгебра — это ассоциативная алгебра с единицей, в каждом базисе которой найдется обратимый элемент. Поскольку в локальной алгебре ровно один двухсторонний идеал $\{0\}$, то теорема Алдера - Штрассена для произвольной локальной алгебры A дает нижнюю оценку:

$$R(A) \geqslant 2\dim A - 1.$$

Описание всех локальных алгебр минимального ранга, то есть тех, для которых $R(A) = 2\dim A - 1$, было получено в 1985 году (*Büchi, Clausen* [18]).

Задача полного описания произвольных алгебр минимального ранга с точки зрения их алгебраической структуры решалась многими математиками в течение почти 20 лет. В 2002 году Маркус Блезер получил полное описание всех алгебр минимального ранга над произвольными полями [19].

Теорема 2. *(Bläser M.) Алгебра A над полем k является алгеброй минимального ранга тогда и только тогда, когда*

$$A \cong C_1 \times C_2 \ldots \times C_s \times k^{2\times 2} \times \ldots \times k^{2\times 2} \times B,$$

где $C_1, \ldots, C_s$ — локальные алгебры минимального ранга, $k^{2\times 2}$ — алгебра матриц порядка 2 над полем k, а B — сверхосновная алгебра минимального ранга.

Для билинейной сложности умножения в алгебре матриц порядка 3 теорема Алдера - Штрассена дает нижнюю оценку 17. Однако с помощью

результата Блезера можно показать, что алгебра матриц порядка 3 не является алгеброй минимального ранга, что повышает эту оценку до 18.

Еще одна из алгебр, которая не является алгеброй минимального ранга, — это алгебра кватернионов. Это 4-мерная ассоциативная алгебра, которая в базисе $\{1, i, j, k\}$ задается следующей таблицей умножения:

	1	i	j	k
1	1	i	j	k
i	i	-1	k	$-j$
j	j	$-k$	-1	i
k	k	j	$-i$	-1

Теорема Алдера - Штрассена дает для билинейной сложности умножения в этой алгебре нижнюю оценку 7. Однако еще в 1975 году было доказано [20], что билинейная сложность умножения в алгебре кватернионов равна 8, то есть алгебра кватернионов — это алгебра почти минимального ранга. Вопрос об описании всех алгебр почти минимального ранга пока не решен. Но для алгебры матриц порядка 3 Блезер [7] доказал, что она не является алгеброй почти минимального ранга, что повышает оценку билинейной сложности умножения в этой алгебре до 19 (напомним, что наилучшая верхняя оценка для этой сложности — 23).

Одним из камней преткновения при описании алгебр почти минимального ранга оказались алгебры обобщенных кватернионов над произвольным полем отличной от 2 характеристики. Это 4-мерные ассоциативные алгебры, которые в базисе $\{1, i, j, k\}$ задаются следующей таблицей умножения (p, q — ненулевые скаляры):

	1	i	j	k
1	1	i	j	k
i	i	p	k	pj
j	j	$-k$	q	$-qi$
k	k	$-pj$	qi	$-pq$

Известно, что любая алгебра обобщенных кватернионов над полем F отличной от 2 характеристики либо изоморфна алгебре матриц порядка 2 над F (и тогда ее билинейная сложность равна 7), либо является алгеброй с делением, то есть алгеброй, в которой все ненулевые элементы обратимы. В последнем случае из результата Блезера (теорема 2) следует, что алгебра обобщенных кватернионов не является алгеброй минимального ранга и, следовательно, ее билинейная сложность не меньше 8. В 2012 году Лысиков В.В. [21] получил следующий результат.

Теорема 3. *(Лысиков В.В.) Пусть F — поле характеристики, отличной от 2, H — алгебра обобщенных кватернионов с делением над F. Тогда $R_F(H) = 8$.*

Билинейный алгоритм сложности 8 в этой теореме был построен с использованием более общего результата Лысикова В.В. [21].

Теорема 4. *(Лысиков В.В.) Пусть F — поле, A — локальная алгебра над F, $\dim A = n$. Пусть известно, что $R_F(A) > 2n - 1$, то есть A не является алгеброй минимального ранга. Тогда $R(A) = 2n$ (то есть A — алгебра почти минимального ранга) в том и только в том случае, когда в A существуют пара базисов $u_1 = 1, u_2, \ldots, u_n$ и $v_1 = 1, v_2, \ldots, v_n$ и пара наборов элементов $z'_1, \ldots, z'_n$ и $z''_1, \ldots, z''_n$ такие, что*

$$u_i v_j = \lambda_{ij} z'_i + \mu_{ij} z''_j$$

для некоторых $\lambda_{ij}, \mu_{ij} \in F$.

Решение задачи о сложности алгебры обобщенных кватернионов позволило полностью завершить описание алгебр почти минимального ранга для случая полупростых алгебр. *Полупростыми* называются алгебры, которые можно представить в виде прямого произведения $D_1^{n_1 \times n_1} \times D_2^{n_2 \times n_2} \times \cdots \times D_t^{n_t \times n_t}$, где все D_i — алгебры с делением, и $D_i^{n_i \times n_i}$ — алгебра матриц порядка n_i над D_i. Поскольку в такой алгебре ровно t максимальных двухсторонних идеалов, то для билинейной сложности умножения в такой алгебре теорема Алдера - Штрассена дает нижнюю оценку: $R(A) \geqslant 2 \dim A - t$.

Известно описание полупростых алгебр почти минимального ранга над полем действительных чисел [22].

Теорема 5. *(Bläser, de Voltaire, 2009). Любая полупростая алгебра почти минимального ранга над $\mathbb{R}$ имеет вид $\mathbb{H} \times \mathbb{R}^{2\times 2} \times \cdots \times \mathbb{R}^{2\times 2} \times \mathbb{C} \times \cdots \times \mathbb{C} \times \mathbb{R} \times \cdots \times \mathbb{R}$.*

Лысикову удалось обобщить этот результат на произвольные поля характеристики, отличной от 2.

Теорема 6. *(Лысиков В.В. [21]) Пусть F — бесконечное поле характеристики, отличной от 2. Любая полупростая алгебра почти минимального ранга над F имеет вид H или $H \times M$, где H — алгебра обобщенных кватернионов с делением, M — алгебра минимального ранга.*

Выше отмечалось, что мультипликативная сложность семейства билинейных форм (при которой предполагается, что все переменные коммутируют между собой) не превосходит билинейной сложности этого же семейства, причем существуют семейства билинейных форм, для которых мультипликативная сложность строго меньше, чем билинейная сложность. Из доказательства теоремы Алдера - Штрассена легко следует, что нижняя оценка этой теоремы справедлива не только для ранга алгебры, но и для мультипликативной сложности. Алгебры, для которых мультипликативная сложность совпадает с этой оценкой, называют *алгебрами минимальной мультипликативной сложности*. Таким образом, любая алгебра минимального ранга является и алгеброй минимальной мультипликативной сложности. Однако в принципе могли бы существовать алгебры, в которых ранг не достигает нижней оценки, а мультипликативная сложность досигает. На самом деле оказывается, что это невозможно. Еще в 1981 году Фейг получил такой результат для алгебр с делением [23].

Теорема 7. *(Feig E., 1981) Алгебра с делением D является алгеброй минимальной мультипликативной сложности тогда и только тогда, когда она является алгеброй минимального ранга.*

Окончательно, для произвольных алгебр этот вопрос разрешил Чокаев в совместных исследованиях с Блезером [24].

Теорема 8. *Произвольная алгебра является алгеброй минимальной мультипликативной сложности тогда и только тогда, когда она является алгеброй минимального ранга.*

Теорема Алдера-Штрассена дает нижнюю оценку на ранг алгебры с коэффициентом 2 при размерности алгебры. Если ограничить семейство алгебр, то удается получать более высокие оценки. Например, для алгебр матриц над алгебрами с делением Блезером [25] был получен следующий результат.

Теорема 9. *(Bläser M., 2005). Пусть F — поле, D — алгебра с делением над F, $D^{n\times n}$ — алгебра матриц порядка n над D и $A \cong D^{n\times n}$. Тогда $R(A) \geqslant \frac{5}{2}\dim A - 3n$.*

Лысикову удалось немного усилить эту оценку для случая, когда D — расширение основного поля.

Теорема 10. *(Лысиков В.В. [21]). Пусть F — поле, K — расширение F, $A \cong K^{n\times n}$. Тогда $R(A) \geqslant \frac{5}{2}\dim A - 3n + 1$.*

Эта оценка лучше других известных оценок при $\dim K = 2$, $n = 3$ и $\dim K = 3$, $n = 2$.

При рассмотрении сложности вычисления систем билинейных форм интересен вопрос о влиянии на эту сложность поля или кольца, над которым эти формы рассматриваются.

Пусть $\varphi\colon U \times V \to W$ — билинейное отображение. Если $x = (a_1, \ldots, a_n)$, $y = (b_1, \ldots, b_m)$, $\varphi(x, y) = (c_1, \ldots, c_l)$ в некоторых базисах, то

$$c_k = \sum_{i=1}^{n}\sum_{j=1}^{m} t_{ijk} a_i b_j.$$

Если коэффициенты t_{ijk} являются целыми числами, то отображение φ можно рассмотреть при a_i и b_j, принадлежащих произвольному кольцу. Такое отображение называется $\mathbb{Z}$-*билинейным*.

Примеры $\mathbb{Z}$-билинейных отображений: умножение матриц; умножение полиномов.

Для умножения матриц известно, например, следующее утверждение [26].

Теорема 11. *Если сложность умножения матриц порядка n над некоторым полем не превосходит $O(n^\alpha)$, то она также не превосходит $O(n^\alpha)$ над любым полем той же характеристики.*

Вопрос о связи сложности $\mathbb{Z}$-билинейных отображений над полями разной характеристики более сложен. Здесь интересный результат получил Лысиков В.В. [27].

Пусть $R_F(\varphi)$ — ранг φ, рассматриваемого как билинейное отображение над полем F. Пусть $\bar{\mathbb{Q}}$ — поле алгебраических чисел (алгебраически замкнутое поле характеристики 0), $\bar{\mathbb{F}}_p$ — алгебраически замкнутое поле характеристики p.

Теорема 12. *(Лысиков В.В. [27]) Пусть φ — $\mathbb{Z}$-билинейное отображение. Тогда*

$$R_{\bar{\mathbb{Q}}}(\varphi) = R_{\bar{\mathbb{F}}_p}(\varphi)$$

для всех простых характеристик за исключением, может быть, конечного числа.

В 2003 году Генри Коэн и Кристофер Уманс [28] предложили новый подход для получения верхних оценок сложности умножения матриц, основанный на вложениях в групповые алгебры. В частности, было показано, что установление сложности умножения в групповых алгебрах влечет определение сложности умножения матриц.

Определение 5. Алгебра A над полем F называется *групповой алгеброй*, если существует такой базис $g_1, g_2, \ldots, g_n$ этой алгебры, что множество $\{g_1, g_2, \ldots, g_n\}$ образует некоторую группу G относительно умножения в A. В этом случае алгебра A обозначается $F[G]$.

Элементы групповой алгебры можно рассматривать как формальные суммы

$$a_1 g_1 + \ldots + a_n g_n,$$

где $g_1, g_2, \ldots, g_n$ — это все n элементов группы, упорядоченные некоторым образом, а коэффициенты при них — произвольные элементы рассматриваемого поля. При этом элементы групповой алгебры умножаются с учетом дистрибутивности, g_i перемножаются как в группе, а коэффициенты при них перемножаются как элементы поля. Интерес к изучению групповых алгебр в связи с изучением сложности умножения матриц обусловлен следующим результатом из теории представлений групп.

Теорема 13. *Каждая групповая алгебра над полем комплексных чисел является прямым произведением матричных алгебр.*

Поспелов А.Д., Чокаев Б.В. и автор изучали билинейную и мультипликативную сложность умножения в групповых алгебрах для различных групп и полей. Достаточно глубоко был изучен случай групповых алгебр для коммутативных групп. Если рассматривать коммутативные групповые алгебры над алгебраически замкнутыми полями характеристики 0, то ситуация очень проста.

Теорема 14. *Пусть A — коммутативная групповая алгебра над алгебраически замкнутым полем k характеристики 0. Тогда $A \cong k^{\dim A}$ и $R(A) = C(A) = \dim A$. При этом A является алгеброй минимального ранга.*

Для случая алгебраически замкнутых полей простой характеристики p удалось получить следующий результат.

Теорема 15. *(Поспелов А.Д.) Пусть A — коммутативная групповая алгебра размерности n над алгебраически замкнутым полем k простой характеристики p, и $n = p^d t$, $p \nmid t$. Тогда существует такая сверхосновная алгебра B над k минимального ранга, что $A \cong B^t$ и $R(A) = C(A) = 2 \dim A - t$. При этом A является алгеброй минимального ранга.*

Над алгебраически незамкнутыми полями ситуация оказывается намного сложнее. В частности, для поля $\mathbb{R}$ вещественных чисел Поспелов [29] получил следующие результаты.

Теорема 16. *(Поспелов А.Д.) Пусть $G \cong \mathbb{Z}_{n_1} \times \ldots \times \mathbb{Z}_{n_s}$, $n = n_1 \times \ldots \times n_s$ — порядок группы и m — число четных чисел среди $n_1, \ldots, n_s$. Тогда*

$$\mathbb{R}[G] \cong \mathbb{R}^{2^m} \times (\mathbb{R}[X]/(X^2+1))^{\frac{n-2^m}{2}}$$

и

$$R(\mathbb{R}[G]) = C(\mathbb{R}[G]) = \frac{3}{2} \dim \mathbb{R}[G] - 2^{m-1}.$$

Теорема 17. *(Поспелов А.Д.) Пусть $A_1, A_2, \ldots$ — последовательность коммутативных групповых алгебр над полем вещественных чисел.*

1) *Если существует предел $c_A = \lim_{n\to\infty} \frac{R(A_n)}{\dim A_n}$, то он равен одному из чисел $c_0 = \frac{3}{2}$, $c_i = \frac{3}{2} - \frac{1}{2i}$, $i \geq 1$.*

2) *Если $c_A = c_i$, $i \geq 1$, то, начиная с некоторого N, для всех $n \geq N$ выполняется равенство $R(A_n) = c_i \dim A_n$. Если $c_A = c_0$, то всегда $R(A_n) < c_0 \dim A_n$.*

3) *Для любого c_i, $i \geq 0$, существует последовательность коммутативных групповых алгебр над полем вещественных чисел, константа асимптотики сложности которой равна c_i.*

Полностью завершил исследование сложности коммутативных групповых алгебр над полями характеристики 0 Чокаев Б.В. [30]

Теорема 18. *(Чокаев Б.В.) Пусть $A = F[G]$ — групповая алгебра коммутативной группы порядка n над произвольным полем F характеристики 0. Тогда алгебра A является алгеброй минимального ранга и*

$$R(A) = 2n - \sigma_F,$$

где σ_F определяется (довольно сложно) по параметрам разложения группы G в прямое произведение примарных групп и параметрам некоторых неприводимых многочленов над полем F. В частности, для любого поля F характеристики 0 выполняется $\sigma_F \geq \sigma_Q$, где σ_Q — значение параметра σ для поля рациональных чисел.

При этом Чокаевым для любого поля F характеристики 0 описано разложение групповой алгебры коммутативной группы в прямое произведение неразложимых алгебр.

Блезеру удалось построить специальную последовательность алгебр, для которой в нижней оценке мультипликативной сложности (а значит и ранга) коэффициент при размерности алгебры может быть сколь угодно близким к 3. Для полей ненулевой характеристики Чокаеву удалось показать [31], что последовательность алгебр с такой нижней оценкой мультипликативной сложности можно выбрать и среди коммутативных групповых алгебр.

Теорема 19. *(Чокаев Б.В.) Пусть $F[G]$ — групповая алгебра коммутативной группы $G \cong \mathbb{Z}_p \times \mathbb{Z}_p \times \ldots \times \mathbb{Z}_p$ (n сомножителей) над произвольным полем характеристики p. Тогда*

$$C(F[G]) \geq (3 - o(1))dimF[G], \text{ при } n \to \infty.$$

С использованием результатов о сложности групповых алгебр Поспеловым А.Д. получены интересные результаты о сложности умножения полиномов многих переменных [32].

Теорема 20. *(Поспелов А.Д.) Существует алгоритм умножения полиномов над алгебраически замкнутым полем характеристики 0 от m переменных степени N, имеющий билинейную сложность N.*

Теорема 21. *(Поспелов А.Д.) Существует алгоритм умножения полиномов над алгебраически замкнутым полем простой характеристики p от m переменных степени N, имеющий билинейную сложность $2N - t$, где t — наибольший натуральный делитель N, не делящийся на p.*

Соответствующие алгоритмы строятся путем сведения к умножению в коммутативных групповых алгебрах.

Из некоммутативных групп нами были рассмотрены 2 группы — группа подстановок третьего порядка и группа симметрий квадрата [33, 34]. Получены следующие результаты.

Теорема 22. *(Поспелов А.Д., Алексеев В.Б.) Пусть S_3 — полная группа подстановок третьего порядка. Тогда $\mathbb{C}[S_3] \cong \mathbb{C}^{2\times 2} \times \mathbb{C}^2$, $\mathbb{R}[S_3] \cong \mathbb{R}^{2\times 2} \times \mathbb{R}^2$, $R(\mathbb{C}[S_3]) = C(\mathbb{C}[S_3]) = 9$, $R(\mathbb{R}[S_3]) = C(\mathbb{R}[S_3]) = 9$, причем в $\mathbb{C}[S_3]$ (в $\mathbb{R}[S_3]$) существует единственная подалгебра, изоморфная $\mathbb{C}^{2\times 2}$ (соответственно, изоморфная $\mathbb{R}^{2\times 2}$).*

Теорема 23. *(Поспелов А.Д., Алексеев В.Б.) Пусть Q — группа симметрий квадрата. Тогда $\mathbb{C}[Q] \cong \mathbb{C}^{2\times 2} \times \mathbb{C}^4$, $R(\mathbb{C}[Q]) = C(\mathbb{C}[Q]) = 11$. При этом $\mathbb{C}[Q] \cong \mathbb{C}[H]$, где H — группа кватернионов (порядка 8) с элементами ± 1, $\pm i$, $\pm j$, $\pm k$, $H \ncong Q$.*

Нижние оценки в теоремах 22 и 23 легко вытекали бы из гипотезы о прямой сумме (см. первую половину статьи), однако ее справедливость пока не установлена. Поэтому во всех случаях нижние оценки получены независимо с использованием теоремы Алдера-Штрассена.

Для других некоммутативных групп получение точных значений билинейной сложности проблематично, поскольку в их разложениях в прямое произведение матричных групп начинают появляться алгебры матриц порядка большего чем 2. А как отмечено в начале статьи, точное значение билинейной сложности умножения в таких алгебрах пока неизвестно даже для алгебры матриц порядка 3.

Одну из таких групп — группу четных подстановок четвертого порядка — исследовал А.Д. Поспелов [32]. Он установил интересные связи между сложностью соответствующей групповой алгебры и сложностью алгебры матриц порядка 3. Из этих связей он получил очень интересный результат, который гласит, что либо хотя бы в одной из этих алгебр переход от поля вещественных чисел к полю комплексных чисел уменьшает сложность, либо гипотеза о прямой сумме неверна.

Теорема 24. *(Поспелов А.Д.) Пусть A_4 — группа четных подстановок четвертого порядка. Тогда $\mathbb{C}[A_4] \cong \mathbb{C}^{3\times 3} \times \mathbb{C}^3$, $\mathbb{R}[A_4] \cong \mathbb{R}^{3\times 3} \times \mathbb{R} \times \mathbb{R}[X]/(X^2+1)$, $R(\mathbb{C}^{3\times 3}) = R(\mathbb{C}[A_4]) - 3$, $R(\mathbb{R}^{3\times 3}) \geq R(\mathbb{R}[A_4]) - 4$.*

Справедливо, по крайней мере, одно из следующих утверждений:

1) $R(\mathbb{C}^{3\times 3}) < R(\mathbb{R}^{3\times 3})$.

2) Гипотеза о прямой сумме неверна.

3) $R(\mathbb{C}[A_4]) < R(\mathbb{R}[A_4])$.

В заключение отметим, что задача о наименьшей асимптотической сложности умножения матриц порядка n, которая полвека назад дала толчок развитию алгебраической теории сложности, остается пока одной из важнейших нерешенных задач этой теории.

Работа выполнена при финансовой поддержке РФФИ (проект 17-01-00782-а).

Список литературы

[1] Burgisser P., Clausen M. Shokrollahi M.A. Algebraic Complexity Theory. Berlin: Springer-Verlag, 1997.

[2] Strassen V. Gaussian elimination is not optimal // Numer. Math. 1969. Vol. 13. P. 354-356. [Имеется перевод: Штрассен В. Алгоритм Гаусса не оптимален // Кибернетический сборник, вып. 7. М.: Мир, 1970. С. 67-70].

[3] Coppersmith D., Winograd S. Matrix Multiplication via Arithmetic Progressions // J. Symbolic Computation. 1990. Vol. 9, no. 3. P. 251-280.

[4] Waksman A. On Winograd's algorithm for inner products // IEEE Trans. Comput. 1970. Vol. C-19, no. 4. P. 360-361.

[5] Alekseyev V.B. On the complexity of some algorithms of matrix multiplication // Journal of Algorithms. 1985. Vol. 6, no. 1. P. 71-85.

[6] Laderman J.D. A noncommutative algorithm for multiplying 3×3 matrices using 23 multiplications // Bull. Amer. Math. Soc. 1976. Vol. 82, no. 1. P. 126-128.

[7] Bläser M. On the complexity of the multiplication of matrices of small formats // J. Complexity. 2003. Vol. 19. P. 43-60.

[8] Макаров О.М. Некоммутативный алгоритм умножения квадратных матриц пятого порядка, использующий сто умножений // Журн. выч. матем. и матем. физики. 1987. Т. 27, вып. 2. С. 311-315.

[9] Смирнов А.В. О билинейной сложности и практических алгоритмах умножения матриц // Журн. выч. матем. и матем. физики. 2013. Т. 53, вып. 12. С. 1970-1984.

[10] Hopcroft J.E., Musinski J. Duality applied to the complexity of matrix multiplication and other bilinear forms // SIAM J. Comput. 1973. Vol. 2, no. 3. P. 159-173.

[11] Winograd S. On multiplication of 2×2 matrices // Linear Algebra and Appl. 1971. Vol. 4. P. 381-388.

[12] de Groote H.F. On varieties of optimal algorithms for the computation of bilinear mappings. II. Optimal algorithms for 2×2 matrix multiplication // Theoret. Comput. Sci. 1978. Vol. 7, no. 2. P. 127-148.

[13] Hopcroft J.E., Kerr L.R. On minimizing the number of multiplications necessary for matrix multiplication // SIAM J. Appl. Math. 1971. Vol. 20, no. 1. P. 127-148.

[14] Алексеев В.Б., Смирнов А.В. О точной и приближенной билинейных сложностях умножения матриц размеров 4×2 и 2×2 // Современные проблемы математики. 2013. Вып. 17. С. 135-152.

[15] Алексеев В.Б. О билинейной сложности умножения матриц размеров 5×2 и 2×2 // Ученые записки Казанского университета. Серия Физико-математические науки. 2014. Т. 156, вып. 3. С. 19-29.

[16] Алексеев В.Б. О билинейной сложности умножения матриц размеров $m \times 2$ и 2×2 // Чебышевский сборник. 2015. Т. 16, вып. 4. С. 11-27.

[17] Alder A., Strassen V. On the Algorithmic Complexity of Associative Algebras // Theor. Comput. Sci. 1981. Vol. 15. P. 201-211.

[18] Büchi W, Clausen M. On a class of primary algebras of minimal rank // Lin. Alg. Appl. 1985. Vol. 69. P. 249-268.

[19] Bläser M. A Complete Characterization of the Algebras of Minimal Bilinear Complexity // SIAM J. Comput. 2004. Vol. 34, no. 2. P. 277-298.

[20] De Groote H. F. On the complexity of quaternion multiplication // Information Processing Letters. 1975. Vol. 3, no. 6. P. 177-179.

[21] Лысиков В. В. Об алгебрах почти минимального ранга // Дискретная математика. 2012. Т. 24, вып. 4. С. 3-18.

[22] Bläser M., de Voltaire A.M. Semisimple algebras of almost minimal rank over the reals // Theor. Comput. Sci. 2009. Vol. 410, no. 50. P. 5202-5214.

[23] Feig E. On systems of bilinear forms whose minimal division-free algorithms are all bilinear // J. Algorithms. 1981. Vol. 2, no. 3. P. 261-281.

[24] Bläser M., Chokaev B. Algebras of minimal multiplicative complexity. // Proc. 27th Ann. IEEE Computational Complexity Conference (CCC). 2012. P. 224-234.

[25] Bläser M. Beyond the Alder-Strassen bound // Theor. Comput. Sci. 2005. Vol. 331, no. 1. P. 3-21.

[26] Schönhage A. Partial and total matrix multiplication // SIAM J. Comput. 1981. Vol. 10, no. 3. P. 434-455.

[27] Лысиков В. В. О билинейных алгоритмах над полями различных характеристик // Вестник Московского Университета. Серия 15: Вычислительная математика и кибернетика. 2013, вып. 4. С. 33-38.

[28] Cohn H., Umans C. A Group-Theoretic Approach to Fast Matrix Multiplication // Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science. 2003. P. 438-449.

[29] Поспелов А. Д. Ранг коммутативных групповых алгебр над полями комплексных и вещественных чисел // Проблемы теоретической кибернетики. Тезисы докладов XIV Международной конференции (Пенза, 23-28 мая2005 г.) Под редакцией О. Б. Лупанова. М.: Изд-во мех.-мат. ф-та МГУ, 2005. С. 125.

[30] Чокаев Б. В. Сложность умножения в коммутативных групповых алгебрах над полями характеристики 0 // Вестник Московского Университета. Серия 15: Вычислительная математика и кибернетика. 2010, вып. 4. С. 30-40.

[31] Чокаев Б. В. Сложность умножения в коммутативных групповых алгебрах над полями простой характеристики // Дискретная математика. 2010. Т. 22, вып. 4. С. 121-137.

[32] Поспелов А. Д. Сложность умножения в ассоциативных алгебрах // Диссертация на соискание ученой степени кандидата физико-математических наук / Московский государственный университет им. М.В. Ломоносова. М., 2008.

[33] Алексеев В. Б., Поспелов А. Д. Сложность умножения в некоторых групповых алгебрах // Дискретная математика. 2005. Т. 17, вып. 1. С. 3-17.

[34] Алексеев В. Б., Поспелов А. Д. Сложность умножения в групповой алгебре симметрий квадрата // Труды 6-ой Международной конференции « Дискретные модели в теории управляющих систем», 7-11 декабря 2004 г. М.: Изд. отдел ф-та ВМиК МГУ, 2004. С. 8-11.

On some results in algebraic complexity theory

Alekseev V.B.

In this paper we give a survey of some results on the computational complexity of algebras, in particular, obtained at the Department of Mathematical Cybernetics of the M.V. Lomonosov Moscow State University by the author and his students: Pospelov A.D., Chokaev B.V., Lysikov V.V.

Keywords: algebraic complexity, algebra, rank of algebra, bilinear complexity, multiplicative complexity, complexity of matrix multiplication.

Особенности расчётов обобщенной модели "чёрной нефти" вблизи критической точки раствора

Колдоба Е.В.

Для моделирования фазовых переходов в многокомпонентных растворах часто используется обобщенная модель «черной нефти». При изучении численных неустойчивостей модели было обнаружено, что они могут возникать из-за термодинамического рассогласования функций в окрестности критической точки раствора, которая, как известно, физически неустойчива по своей природе. Все измеряемые величины из-за флуктуаций, происходящих в этой области, имеют значительные погрешности измерений, что приводит к существенному рассогласованию параметров модели и соответственно к численным неустойчивостям.
Предлагается физический подход к моделированию таких областей. Объяснены некоторые причины неточности моделей «черной нефти».

Введение

Для прогнозирования добычи углеводородов на нефтяных и газовых месторождений широко применяются методы численного моделирования. Расчеты могут идти от нескольких часов до нескольких месяцев в зависимости от сложности и точности модели, а также производительности компьютера. В нефти и газовом конденсате содержатся сотни компонент, причем, присутствие даже небольших концентрации одного из них может изменить количество фаз. Для повышения нефте и газоотдачи в пласт закачивают воду и газ, т.е. фаз в системе становится еще больше. К тому же, на фазовых переходах при изменении давления постоянно меняются концентрации в фазах. Сложное фазовое поведение такой системы описывается физикой растворов и термодинамикой. Существует два возможных способа описания фазового состояния в растворах нефти, воды и газа:

1) с помощью коэффициентов объемного расширения фаз и функций растворимости компонентов в фазах, измеренных на скважинах (месторождениях) или в лабораториях,

2) с помощью фазовых диаграмм и уравнений состояния; к сожалению, фазовые диаграммы для N-компонентного флюида становятся N-мерными.

Первый способ менее точный, не учитывает ряд эффектов, обусловленных
многокомпонентностью растворов. Традиционно применяется в моделях «черной нефти». Название модели взято в кавычке для того, чтобы подчеркнуть, что модель, первоначально созданная для описания фильтрации с фазовыми переходами реальной черной нефти, затем была модифицирована для моделирования фильтрации летучей нефти (не черной), газированной воды и т.д. В модели «черной нефти» для описания фильтрации используется M уравнений переноса (M – количество фаз): если фазы две, соответственно моделируются только два уравнения фильтрации. Такая модель требует относительно небольших вычислительных ресурсов, поэтому она широко используется при моделировании месторождений. Данные, полученные с месторождений или лабораторий, содержат ценную интегральную информацию о поведении реальных растворов в пластах. Однако, экспериментальные данные требуют иногда существенной компьютерной обработки перед использованием, т.к. погрешности измерений могут быть значительными особенно вблизи критической точки раствора, которая, как известно, физически неустойчива по своей природе и все измеряемые величины флуктуируют. По мере приближения к критической точке бесконечно растет восприимчивость системы (раствора) ко всем внешним воздействиям: неоднородности среды, капиллярных сил и т.д., при этом ошибки измерений могут быть ~100%.

Второй способ более точно описывает сложное многокомпонентное фазовое поведение, обычно используется в так называемых композиционных моделях фильтрации. Однако, для моделирования фильтрации N-компонентного флюида необходимо числено решать N нелинейных дифференциальных уравнений в частных производных, что требует значительных вычислительных ресурсов. Кроме того, подход содержит ряд приближений (идеализаций), поэтому иногда плохо описывает реальное месторождение, но модель может быть настроена на реальный флюид (если есть необходимые данные с месторождения).

Заметим, что оба эти способа описания, на первый взгляд, должны давать одинаковые результаты при моделировании фазовых переходов, однако, на практике это обычно не так. Даже для двухкомпонентной системы с фазовыми переходами для некоторых типов диаграмм результаты отличаются значительно. В статье второй подход (теоретический) используется для объяснения некоторых эффектов, снижающих точность первого подхода. Используемые фазовых диаграмм могут дать наглядную информацию о фазовом состоянии, её особых точках и т.д. К сожалению, в общем случае N-мерные фазовые диаграммы сложно анализировать. Однако, в данной работе рассматривается только двух компонентный флюид. Т.к. в ряде теоретических работ показано, что модель «черной нефти» по сути сводится к модели с двумя псевдокомпонентами, называемыми «нефтяной» и «газовый».

В зависимости от типа залежи при моделировании могу возникнуть те или иные проблемы, а также возможны те или иные приближения. Природные залежи по особенностям фазового поведения растворов условно делятся на четыре класса: нелетучая (черная) нефть, летучая нефть, газовый конденсат и газ. Тип залежи определяется прежде всего по тому, как далеко находится рассматриваемый диапазон давлений и температур (p, T) от критической точки флюида (p_c, T_c). Так флюид будет считаться нелетучей (черной) нефтью, если диапазон (p, T) находится достаточно далеко от точки (p_c, T_c), а если близко, то такой флюид будет считаться летучей нефтью или газовым конденсатом, и в моделях »черной нефти» могут возникать численные неустойчивости обусловленные близостью критической точки растворов.

1. Термодинамическая модель.

Рассмотрим термодинамическую модель углеводородного раствора, так как именно в таких растворах критическая точка может оказаться в моделируемой области или рядом с ней. Пусть N –компонентный раствор с полной молярной концентрацией $\{z_i\}$ в двухфазной области расслаивается на газовую и жидкую фазы с концентрациями $\{y_i\}, \{x_i\}$ соответственно, тогда для молярных концентраций выполняются условия:

$$\sum_i z_i = 1, \qquad \sum_i y_i = 1, \qquad \sum_i x_i = 1, \qquad i = 1, 2..N \tag{1}$$

где i - номер компоненты. В дальнейшем будем рассматривать двухкомпонентный, состоящий из более легкой («газовой» или летучей) компо-

ненты и более тяжелой («нефтяной» или менее летучей). Из (1.1) следует, что для такого раствора в формулах можно отказаться от индексов, обозначающих номер компонента: пусть z – полная молярная концентрация первого компонента («газового») компонента, тогда (1-z) - полная молярная концентрация второго («нефтяного») компонента. Если x – молярная концентрация первого компонента («газового») компонента в жидкой фазе, тогда (1-x) - полная молярная концентрация второго («нефтяного») компонента компонента в жидкой фазе и т.д.

На Рис.1 для изотермического случая схематически нарисована фазовая диаграмма «давление - состав» двухкомпонентного раствора, состоящего из «газового» и «нефтяного»
компонентов. По оси абсцисс откладываются все три молярные концентрация более легкого компонента: x, y, z. Кривая кипения $x = x_{g,O}(p)$ и кривая конденсации $y = y_{g,G}(p)$ ограничивают двухфазную область (**O**+**G**), причем в критической точке C они сходятся. Над кривой кипения раствор находится в однофазном жидком состоянии (нефть **O**). Справа и внизу от кривой конденсации раствор находится в однофазном газовом состоянии (газ **G**). В двухфазной области концентрации компонент в фазах зависят от давления и задаются кривыми $x = x_{g,O}(p)$ и $y = y_{g,G}(p)$. Здесь и далее индексы G, O обозначают фазу (газ, нефть), а индексы g, o обозначают компонент (газовый, нефтяной). Так обозначение $x_{g,O}$ означает концентрацию газового компонента в нефтяной фазе. Ясно что концентрация $y_{g,G}$ газового компонента в газовой фазе будет больше, т.е. $y_{g,G} > x_{g,O}$.

Рассмотрим флюид состава z_A при давлении p_D, на фазовой диаграмме (точка D на Рис.1) видно, что флюид находится в однофазном жидком (нефть) недонасыщенном состоянии (undersaturated oil), т.е. «газовый» компонент полностью растворен в жидкой фазе (отсутствует газовая фаза). При понижении давления $p < p_A$ раствор попадает в двухфазную область, в которой и нефть и газ будут насыщенными (saturated oil and saturated gas). Согласно фазовой диаграмме при давлении p_B раствор расслаивается на две фазы, в нефтяной фазе концентрация первого компонента станет x_B, а в газовой y_B.

На фазовой диаграмме (Рис.1) видно, что концентрация флюида z_A близка к концентрации критической точки, но меньше, такую нефть согласно принятой классификации называют «летучей» нефтью, если диапазон давлений задачи включает и близок давлению p_c. А если концентрация флюида z_{A1}, то этот раствор будет называться газовым конденсатом.

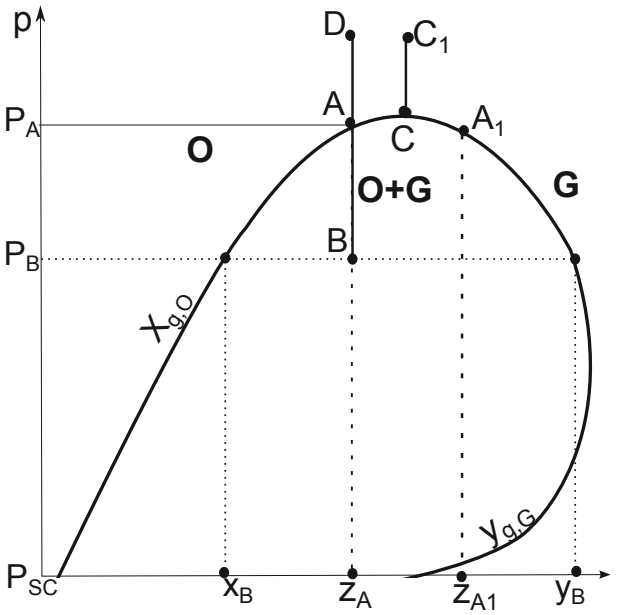


Рис. 1. Фазовая диаграмма «давление-состав».

Пусть n_O, n_G, n- молярные плотности нефтяной, газовой фазы и суммарного флюида соответственно, а S_G- насыщенность газовой фазой (доля объема, которую занимает газ), тогда S_O - насыщенность нефтяной фазой, для насыщенностей выполняется равенство: $S_O + S_G = 1$.

В некоторой области в окрестности критической точки раствора (Рис.1) физические характеристики (концентрации, плотности, вязкости, фазовые проницаемости и т.д.) жидкости должны непрерывным образом переходить в характеристики газовой фазы. А в критической точке «С» они должны быть тождественно равны:

$$n_O \equiv n_G \quad x_{g,O} \equiv y_{g,G} \quad \mu_O \equiv \mu_G$$

Но из-за больших погрешностей измерения в окрестности критической точки тождества не выполняются.

Луч CC_1 на Рис.1 условно разделяет однофазную области на нефть и газ. В реальных растворах этой границы не существует: в закритической области все характеристики жидкой фазы непрерывным образом переходят в характеристики газовой фазы. Однако, функции для газа и нефти в модели «черной нефти» разные, и обычно термодинамически не

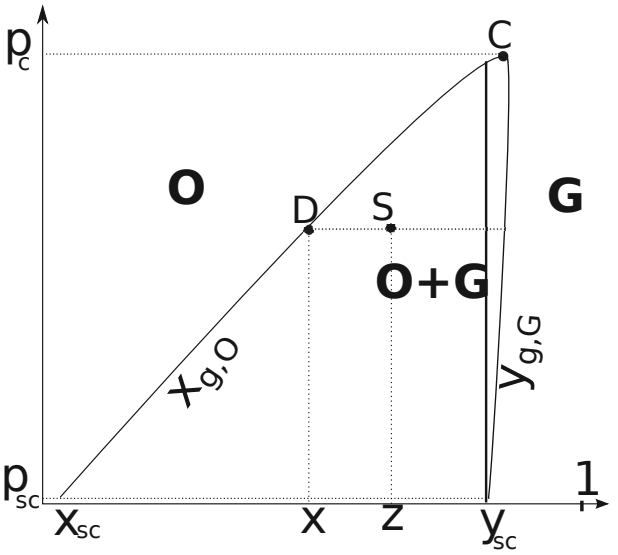


Рис. 2. Треугольная фазовая диаграмма «давление-состав».

согласованны, поэтому в алгоритме традиционно вводится линия, разделяющая область на жидкую и газовую. При пересечении этой линии на счете возникают нефизические скачки величин. Такие скачки также устраняются способом, предложенным в работе.

2. Классическая модель нелетучей или «черной нефти»

Классическая модель применима, если фазовая диаграмма «состав-давление» флюида имеет треугольный вид (Рис.2), т.е. во всем рассматриваемом диапазоне давлений состав газовой фазы практически постоянный $y_{g,G}(p) \approx y_{sc} \approx Const$, в то время как состав жидкой фазы $x_{g,O}(p) = x(p)$ меняется произвольным образом.

Обозначим концентрации в нефти и в газе при стандартных условиях (standart condition: $p = 1bar \approx 1, t = 0^oC$) соответственно x_{sc}, y_{sc} (Рис.2). Флюид состава x_{sc} считается чистым «нефтяным» псевдо-компонентом, а флюид состава y_{sc} - чистым «газовым» псевдо-компонентом. Предполагается, что все возможные концентрации в пласте представляются в виде линейной комбинации x_{sc} и y_{sc}. Рассмотрим несколько примеров. Фазовая диаграмма «метан-декан» (Рис.3а) удовлетворяет условия клас-

сической модели «черной нефти»: $y_{g,G}(p) \approx y_{sc} \approx Const$. А фазовая диаграмма «метан-гексан» (Рис.3б) нет, т.к. $y_{g,G}(p) \neq y_{sc}$.

В классической модель нелетучей или «черной» нефти [1] задаются следующие функции:

- растворимость газового компонента в нефти R_O,
- коэффициенты (функции) объемного расширения нефти и газа B_O, B_G.

Растворимость газового компонента в нефти вычисляется следующим образом. Пусть флюид состава z при давлении p_S находится в двухфазной области и расслаивается на нефть с концентрацией x и газ с концентрацией y. Нефть с концентрацией x (точка D на Рис.2) «извлекается» на поверхность (точка X). Из нефти выделяется газ. Измеряются объемы выделившегося газа $V_{G,sc}$ и оставшейся жидкости $V_{O,sc}$ и вычисляется растворимость газа в нефти:

$$R_O = R_O(p, z) = \frac{V_{G,sc}}{V_{O,sc}},$$

Для вычисления B_O, B_G вычисляются объёмы нефти и газа соответсвенно при стандартных условиях и в пласте $V_{O,sc}, V_{O,D}, V_{G,sc}, V_{G,D}$ и находится их отношение:

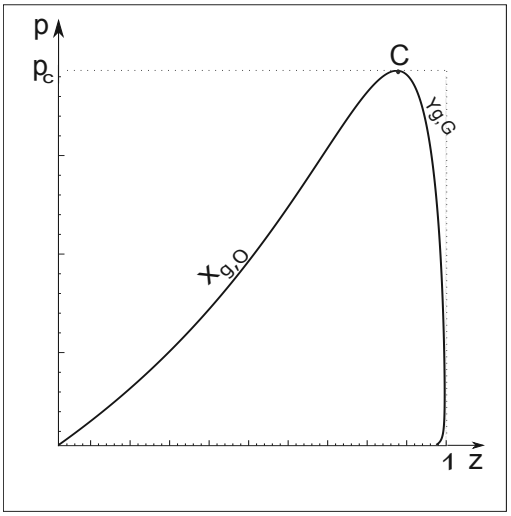
$$B_O = B_O(p, z) = \frac{V_{O,D}}{V_{O,sc}}, \qquad B_G = B_G(p, z) = \frac{V_{G,D}}{V_{G,sc}}$$

В классической модели «черной нефти» делаются явно и неявно следующие предположения:

- процесс равновесный и изотермический,
- извлеченная на поверхность нефть находится в двухфазном состоянии,
- двумя «поверхностными» псевдо-компонентами можно описать фазовое поведение многокомпонентной смеси во всем рассматриваемом диапазоне давлений [6],
- нефть - нелетучая, т.е. «нефтяной» компонент содержится только в нефтяной фазе, фазовая диаграмма имеет треугольный вид $y_{g,G}(p) \approx y_{sc} = Const$;

- отношение объемов газовой и нефтяной фазы, измеренные на поверхности при стандартных условиях, дают полную информацию о растворимости газа в нефти.

Если предположения не выполняются, то классическая модель «черной нефти» дает значительные ошибки.



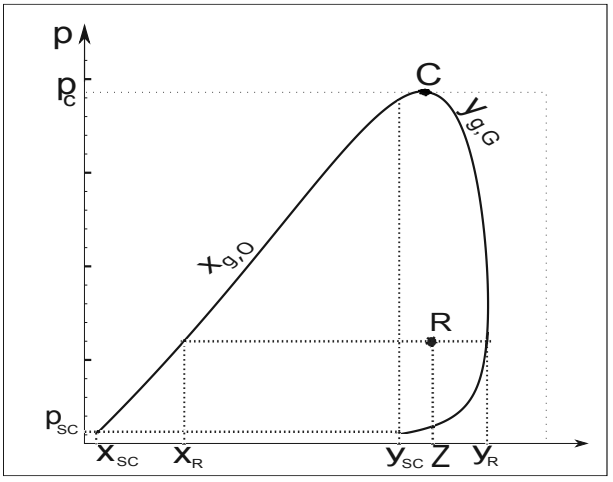


Рис.3 Фазовая диаграмма «давление-состав» при $T = 100^0 C$
а) для $C_1H_4 - C_{10}H_{22}$, б) для $C_1H_4 - C_6H_{14}$

Известно, что функция $R_O(p)$ однозначно связана с молярной концентрацией газовой компоненты (в двухфазной области) в нефти $x(p)$ следующим образом:

$$x = \frac{R_O \cdot \xi_{G,sc}}{R_O \cdot \xi_{G,sc} + \xi_{O,sc}} \tag{2}$$

где $\xi_{G,sc}, \xi_{O,sc}$- молярные плотности газовой и нефтяной фазы соответственно, измеренные на поверхности при стандартных условиях.

Таким образом по формуле (4) устанавливается связь первого и второго способа описания фазового состояния флюида.

Введем обозначения $\alpha = \xi_{G,sc}/\xi_{O,sc}$, где α- константа для конкретной задачи. Тогда взаимно-однозначную связь функций x и R_O можно записать в виде:

$$x = 1 - \frac{1}{\alpha R_O + 1}, \qquad R_O = \frac{1}{\alpha}\left(\frac{1}{1-x} - 1\right) \tag{3}$$

Из (2.2) следует, что если $x(p) = Const$, то $R_O(p) = Const$, что если $x(p)$ в некотором диапазоне аппроксимируется линейной функцией, то $R_O(p)$ аппроксимируется гиперболической функцией и наоборот. В классической модели «черной нефти» при необходимости на счете выполняется экстраполяция функций [4-5]:

- линейная экстраполяция функции $R_O(p)$ в двухфазной области, функция может принимать только такие значения, чтобы выполнялось следующее неравенство
  $x_{g,O}(p) < 1$,
- гармоническая экстраполяция функций $B_O = B_O(p), B_G = B_G(p)$.

На фазовой диаграмме раствора C1-C6 видно, что для описания фазовых переходов в этом случае недостаточно знания функции $R_O(p)$, необходимо учитывать функцию растворимости «нефтяного» компонента в газе $R_G(p)$, именно эта функция вводится далее в обобщенной модели «черной нефти».

3. Обобщенная модель «черной нефти» или модель летучей нефти (volatile oil).

Пусть флюид состава z в пласте при давлении p_S находится в двухфазной области и расслаивается на нефть с концентрацией x и газ с концентрацией y. Из пласта на поверхность (стандартные условия) извлекается

нефть «по одной трубе», а по «другой трубе» извлекается газ. На Рис.4а проиллюстрировано это действие: из пласта (точка B) на поверхность извлекается нефть (точка X), также извлекается на поверхность газ (точки D и Y соответственно). При стандартных условиях измеряются объемы, выделившегося газа и жидкости в первом и втором случае, затем вычисляются растворимости [2-3]:

- растворимость «газового» компонента в нефтяной фазе $R_S = R_S(p,z) = \left(\frac{V_{G,sc}}{V_{O,sc}}\right)_O$

- растворимость «нефтяного» компонента в газовой фазе $R_V = R_V(p,z) = \left(\frac{V_{O,sc}}{V_{G,sc}}\right)_G$

Индексы у скобок означают какой флюид извлекается на поверхность. Понятно, что для извлеченного на поверхность нефтяного флюида, величины объёмов $V_{G,sc}, V_{O,sc}$ будут другие чем для извлеченного газа. Соответственно функции $R_S \cdot R_V \neq 1$.

Кроме того вычисляются коэффициенты объемного расширения нефтяной и газовой фазы:

$$B_O = B_O(p,z) = \frac{V_{O,S}}{V_{O,sc}}, \qquad B_G = B_G(p,z) = \frac{V_{G,S}}{V_{G,sc}}$$

где $V_{O,S}, V_{G,S}$ - объемы нефтяной и газовой фазы при давлении $p = p_S$, а $V_{O,sc}, V_{G,sc}$ - объемы нефтяной и газовой фазы, измеренные при стандартных условиях.

На Рис.4 изображены две фазовые диаграммы «давление-состав». На первой: извлеченные нефть и газ при стандартных условиях оказываются в двухфазной области, это означается, что для них можно вычислить R_O и R_G. На второй диаграмме извлеченный газ (точка D) при стандартных условиях оказывается не в двухфазной области, а в газовой, т.е. $V_{O,sc} = 0$ и нельзя определить R_G. Для такой фазовой диаграммы обобщенная модель не работает и необходимо использовать классическую модель «черной нефти».

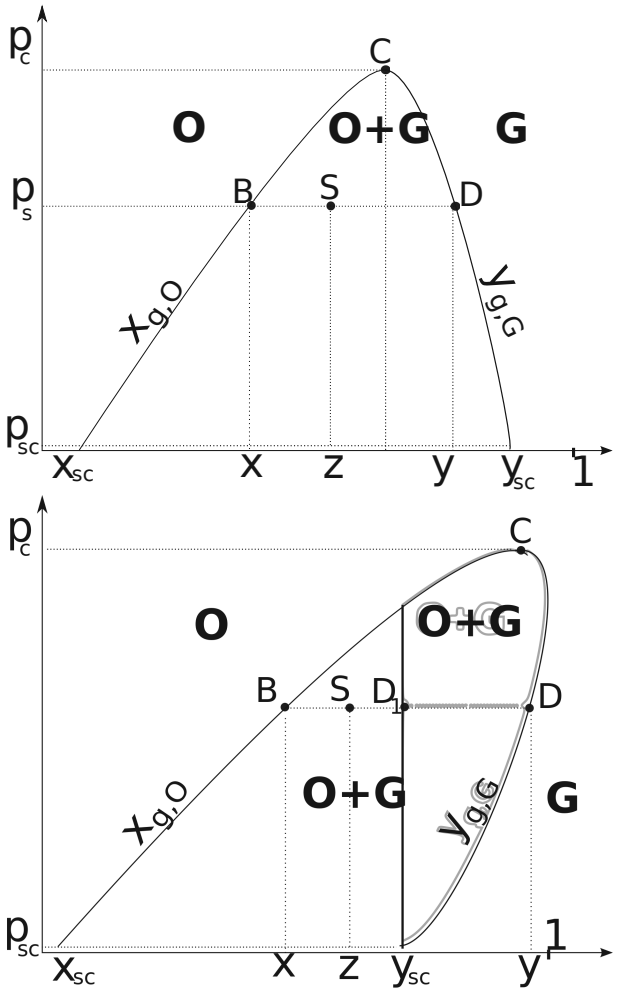


Рис.4 Фазовые диаграммы «давление-состав».

Если R_O и R_G вычислены, то по ним можно найти исходные молярные концентрации в фазах по следующим формулам:

$$x = \frac{\alpha R_O}{\alpha R_O + 1}, y = \frac{1}{R_G/\alpha + 1} \tag{4}$$

Таким образом, уравнения (3.1) устанавливают связь между молярными концентрациями газового компонента в нефти x и в газе y и растворимостями R_O и R_G.

В обобщенной модели рассматриваются и растворимость «газового» компонента в нефти и растворимость «нефтяного» компонента в газе.

Казалось бы такая модель более точная чем классическая модель «черной нефти». Но иногда эта модель дает более значительные ошибки, к тому же она в некоторых областях становится численно неустойчивой. Исследуем эти области.

Молярные плотности фаз выражаются через R_O и R_G следующим образом:

$$\xi_O = \frac{\xi_{O,SC}}{B_O}(\alpha R_O + 1), \qquad \xi_G = \frac{\xi_{G,SC}}{B_G}(R_G/\alpha + 1)$$

Количество молей нефтяной N_o и газовой N_g компоненты в нефти и газе выражаются через фазовые насыщенности S_O и $S_G = 1 - S_O$:

$$N_o = \frac{\xi_{O,SC}}{B_O}S_O + R_G\frac{\xi_{G,SC}}{\alpha B_G}S_G, \quad N_g = \alpha R_O\frac{\xi_{O,SC}}{B_O}S_O + \frac{\xi_{G,SC}}{B_G}S_G$$

разрешая которые относительно фазовых насыщенностей, получаем:

$$S_O = \frac{B_O}{\xi_{O,SC}}\frac{(N_o - N_g R_G/\alpha)}{(1 - R_O R_G)}, \qquad S_G = \frac{B_G}{\xi_{G,SC}}\frac{(N_g - \alpha N_o R_O)}{(1 - R_O R_G)} \tag{5}$$

В формулах (3.2) содержится деление на $(1 - R_O R_G)$, если $R_O R_G = 1$ и все функции термодинамически согласованы, то одновременно обращаются в «0» и знаменатель и числитель:

$$N_o - N_g R_G/\alpha = 0, \qquad N_g - \alpha N_o R_O = 0$$

Возникает неопределенность типа «0/0». Теоретически такая ситуация наблюдается в критической точке раствора и означает выход из двухфазного состояния. В алгоритме такая точка считается однофазным состоянием, например, жидким $S_O = 1,\ S_G = 0$ и тем самым неопределенность не исчезает.

Отметим, что в двухфазном состоянии всегда $R_O R_G < 1$ и $S_O, S_G > 0$.

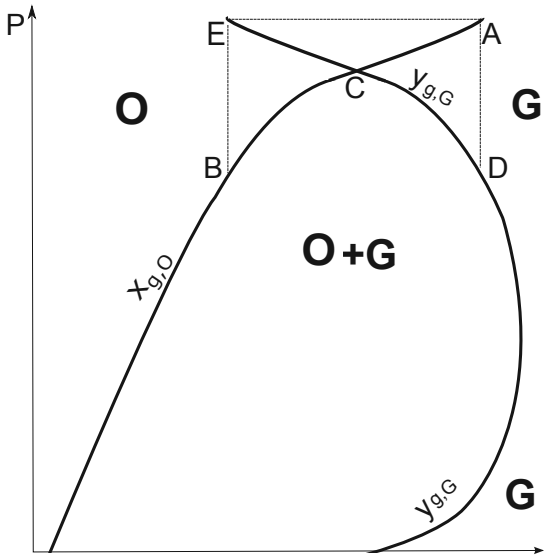
Действительно, в двухфазной области из (2.3) следует:

$$R_O \cdot R_G = \frac{x}{1-x} \cdot \frac{1-y}{y} = \frac{x}{y} \cdot \frac{1-y}{1-x} < 1$$

Это строгое неравенство выполняется, т.к. газового (более легкого) компонента в газе больше чем в нефти: $x < y$, а нефтяного компонента в нефти больше чем в газе
$(1 - x) > (1 - y)$. И только в критической точке выполняется равенство компонент в фазах и $R_O \cdot R_G = 1$.

Так как экспериментальные данные имеют всегда некоторую ошибку измерений, то возможны ситуации, когда: $R_O R_G > 1$.

Покажем, что физический смысл для углеводородных растворов имеют только значения, когда выполняется условие $R_O R_G \leq 1$. Построим фазовую диаграмму для случаев $R_O R_G \leq 1$ и $R_O R_G > 1$, пересчитав R_O, R_G в кривую кипения $x_{g,O}(p)$ и кривую конденсации $y_{g,G}(p)$.



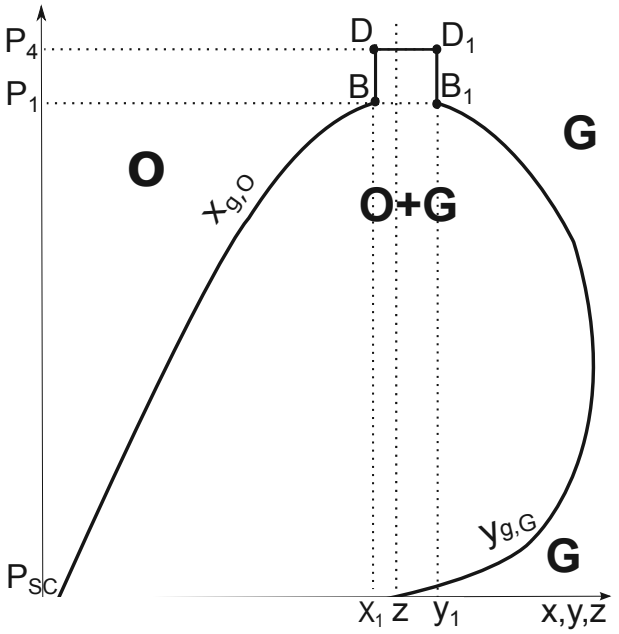


Рис.5-6 Фазовые диаграммы.

Получившийся график схематически изображен на Рис.5. Точка C может восприниматься как точка эвтектики, однако она таковой не является. Действительно луч CE - это продолжение кривой конденсации $y_{g,G}(p)$, т.е пространство под ним это двухфазное состояние. С другой стороны это же пространство находится над кривой кипения $x_{g,O}(p)$ и это означает, что это однофазное жидкое состояние. Таким образом возникает противоречие. Кроме того, по фазовой диаграмме на Рис.5 следует что при давлении p_E концентрация легкого компонента в газовой фазе ниже чем в нефтяной фазе. И все эти несоответствия возникают из-за больших ошибок в измерениях.

Ситуация $R_O R_G > 1$ не возможна в равновесном случае и не имеют физического смысла. Такие данные необходимо удалять и формировать область для сшивки всех функции, описывающих в моделях «черной нефти» раздельно газовую и жидкую фазы: плотности ξ_O, ξ_G, насыщенности фаз S_O, S_G, вязкости μ_O, μ_G, относительные проницаемости k_O, k_G.

Рассмотрим один из возможных случаев сшивки всех функции. Предположим, что при p_1 выполняется $R_O R_G < 1$, а при давлении p_2 наблюдается уже неравенство $R_O R_G > 1$, которое противоречит физическому смыслу. Это значит, что функции R_O, R_G надо обрезать для давлений $p > p_1$. Будем считать условие $R_O R_G < 1$ главным определяющим на-

личие двух фаз. Пересчитаем R_O, R_G в плотности x_O, y_G по формулам (3.1). Вычислим значение величины $\Delta = y_G - x_O$. В диапазоне давлений $[p_1, p_{max}]$ на фазовой диаграмме сформируем Δ-канал (см. Рис.6). Давление p_{max} введем позже так, чтобы согласованными оказались все функции. В диапазоне давлений $[p_1, p_{max}]$ концентрации в жидкости и газе постоянные $x_O \equiv x_O(p_1), y_G \equiv y_O(p_1)$, а это значит, что функции R_O, R_G тоже постоянны $R_O = R_O(p_1), R_G = R_O(p_1)$.

Согласуем плотности фаз. Предположим, что при p_3 молярная плотность жидкой фазы больше или равна газовой $\xi_O \geq \xi_G$, а при p_4 выполняется неравенство $\xi_O < \xi_G$ (противоречит физическому смыслу), тогда:

- если моделируется летучая нефть, то при выходе из 2-х фазной области, флюид будет в жидком состоянии, т.е. для $p > p_3$ плотность будет $\xi = \xi_O(p)$, если еще существует две фазы, то полагаем $\xi_O \equiv \xi_G$;
- если моделируется газовый конденсат, то при выходе из 2-х фазной области, флюид будет в газовом состоянии, т.е.для $p > p_3$ плотность газа $\xi = \xi_G(p)$, если еще существует две фазы, то полагаем $\xi_O \equiv \xi_G$.

Также согласуем вязкости фаз. Предположим , что при p_5 для вязкостей жидкой фазы и газовой выполняется $\mu_O \geq \mu_G$, а при p_6 выполняется неравенство $\mu_O < \mu_G$
(противоречит физическому смыслу), тогда:

- если моделируется летучая нефть $\mu_O = \mu_O(p)$, если еще существует две фазы , полагаем $\mu_O \equiv \mu_G$;
- если моделируется газовый конденсат, то для $p > p_5$ плотность газа $\mu_G = \mu_G(p)$, если еще существует две фазы, полагаем $\mu_O \equiv \mu_G$.

Выберем максимальное давление p_{max} из давлений p_1, p_3, p_5. Если $p_{max} = p_1$, то Δ-канал имеет нулевую длину.

При выходе из Δ-канала насыщенности фаз S_G, S_O и относительные фазовые проницаемости k_G, k_O равны:

- $S_O = 1, k_O = 1, S_G = 0, k_G = 0$, если моделируется летучая нефть ;
- $S_G = 1, k_G = 1, S_O = 0, k_O = 0$, если моделируется газовый конденсат.

Выводы.

В модели «черной нефти» данные часто рассогласованны и содержат значительные ошибки, поэтому на счете могут возникать нефизические скачки функций и численные неустойчивости. В работе предлагается физический подход для расчетов около-критического состояния (near-critical fluid): удаление нефизических данных и создание некоторой области, в которой сшиваются все функции модели (плотности, вязкость, концентрации, относительные проницаемости).

Литература

Список литературы

[1] Азиз Х., Сеттари Э., *Математическое моделирование пластовых систем*, Институт компьютерных исследований, Москва-Ижевск, 2004, 416 стр.

[2] John A.Trangenstein, John B.Bell, *Mathematical Structure of the Black-Oil Model for Petroleum Reservoir Simulation*, SIAM Journal on Applied Mathematics, Vol.49, No. 3(Jun., 1989), pp. 749-783

[3] Chen Z., Huan G., Ma Y., *Computanional Methods for Multiphase Flows in Porous Media*, Southern Metodist University Dallas, Texas, SIAM, 2006.

[4] McCain W.D., Spivey J.P., *Extrapolation of Laboratory Measured Black Oil and Solution Gas Properties for Variable-point Simulation*, SPE Annual Technical Conference and Exhibition, 3-6 October 1999, Houston, Texas.

[5] Bobach T., Farin G., Hansford D., Umlauf G., *Natural neighbor extrapolation using ghost points,* Computer Aided Design, 41(2009), pp. 350-365.

[6] C.F. Leibovici, E.H. Stenby, K. Knudsen, *A Consistent Procedure for Pseudo-Component Delumping,* Fluid Phase Equilibr., 117 (1997), pp. 225–232 8.

Numerical simulation of the general black oil model near the critical point of solution
Koldoba E.V.

For numerical simulation of phase transition in multicomponent solutions it is often used general Black oil model. In the study of the numerical instabilities of the model it has been discovered that they can arise from the mismatch thermodynamic functions near the critical point of solutions, which is known to be physically unstable by nature. All measured values because of fluctuations occurring in this region have significant measurement errors, leading to significant misalignment of model parameters and, respectively, to the numerical instabilities. We propose a physical approach to simulation such problems. It is explained some of the reasons for inaccurate of Black oil models .

Часть 2.
Специальные вопросы теории интеллектуальных систем

Алгоритмы перевода конца цепочки в заданную точку

Бергер И.О.

В работе исследована задача о цепочках.

Приведены результаты об области существования цепочек, полученных из данной переводом конца цепочки в заданную точку; оценки минимума евклидова расстояния между цепочками, получаемыми друг из друга переводом конца в заданную точку; возможное количество цепочек, полученных переводом конца в заданную точку и отличающихся минимальным количеством звеньев от данной цепочки; возможное количество цепочек, находящихся на минимальном расстоянии от данной и полученных переводом конца цепочки в заданную точку, для $n = 2$ и $n = 3$.

Описаны алгоритмы перевода конца цепочки в заданную точку: экспоненциальный алгоритм, перебирающий все возможные цепочки с шагом ε, линейный алгоритм, дающий примерное решение для евклидова расстояния, и линейный алгоритм, дающий точный ответ для расстояния Хэмминга и примерный для евклидова расстояния.

Ключевые слова: цепочка, алгоритм, верхние оценки, нижные оценки, евклидово расстояние, расстояние Хэмминга.

1. Определения и результаты

1.1. Введение

Одна из активно развивающихся областей науки - интеллектуальные системы ([1]-[19]). Одним из её разделов является управление роботами.

Представьте себе робота, который берет некоторые элементы, лежащие на столе, и перемещает их. У этого робота есть одна рука, имеющая несколько суставов, допустим, n. Такая рука - это $n+1$ последовательно соединённых отрезков. Необходимо изучить, как может двигаться рука данного робота, и как можно быстро и с минимальными потерями энергии перевести конец руки робота в нужную точку.

1.2. Основные определения

Определение. *Цепочка* $A = A_0A_1...A_n$ *длины* n - это $n+1$ точек $A_0, A_1, ..., A_n$, $A_i \in \mathbb{R}^2, A_i \neq A_{i+1}$.

Мы представляем цепочку на плоскости как совокупность точек $A_0, A_1, ..., A_n$ и отрезков $A_0A_1, ..., A_{n-1}A_n$.

Определение. Точка A_0 называется *центром цепочки.*

Определение. Отрезок $A_iA_{i+1} \forall i$ называется *звеном.*

Обозначение. $d^A_{i,i+1} = |A_iA_{i+1}|, i = 0..n-1$ - *длина звена.*

Обозначение. $\alpha^A_{0,1} =$ - угол между вектором A_0A_1 и направлением оси OX. $\alpha^A_{i,i+1} = \overline{A_{i-1}A_i}\widehat{}\,\overline{A_iA_{i+1}}, i = 1..n-1$ - угол между векторами $\overline{A_{i-1}A_i}$ и $\overline{A_iA_{i+1}}$.

Определение. *Цепочка* $B = B_0B_1...B_n$ *получается из цепочки* $A = A_0A_1...A_n$ *перемещением конца цепочки*, если $B_0 = A_0$ и $|A_iA_{i+1}| = |B_iB_{i+1}|, i = 0..n-1$.

Обозначение. $\mathfrak{P}(A)$ - множество цепочек, получаемых перемещением конца цепочки из A.

Определение. *Область допустимых положений конца цепочки* $A = A_0A_1...A_n$ - множество точек $\mathfrak{D}(A) = \{P | P = B_n, B$ получена из A перемещением конца цепочки$\}$.

Определение. *Евклидово расстояние между цепочками* $A = A_0A_1...A_n$ *и* $B = B_0B_1...B_n$: $\rho(A,B) = \sum\limits_{i=0}^{n} |A_iB_i|$

Определение. *Расстояние Хемминга между цепочками* $A = A_0A_1...A_n$ *и* $B = B_0B_1...B_n$: $\rho'(A,B) = \sum\limits_{i=0}^{n-1} Ind(A_i \neq B_i or A_{i+1} \neq B_{i+1})$

Определение. Цепочка $B = B_0B_1...B_n$, удовлетворяющая условию U, *удовлетворяет условию* U *и находится на минимальном расстоянии от цепочки* $A = A_0A_1...A_n$, если $\rho(A,B) = \inf_{Csatisf.U} \rho(A,C)$

Определение. Цепочка $B = B_0B_1...B_n$, удовлетворяющая условию U, *удовлетворяет условию* U *и отличается от цепочки* $A = A_0A_1...A_n$ *минимальным количеством звеньев*, если $\rho'(A,B) = \inf_{Csatisf.U} \rho'(A,C)$

1.3. Метрическое пространство

Определение. *Метрическое пространство* есть пара (X,d), где X — множество, а d — числовая функция, которая определена на декартовом произведении $X \times X$, принимает значения в множестве вещественных чисел, и такова, что

1. $d(x,y) = 0 \Leftrightarrow x = y$ (аксиома тождества).

2. $d(x,y) = d(y,x)$ (аксиома симметрии).

3. $d(x,z) \leq d(x,y) + d(y,z)$ (аксиома треугольника или неравенство треугольника).

Функция $d(x,y)$ называется *метрикой*.

Утверждение. Множество цепочек длины n - метрическое пространство с метрикой $\rho(A,B)$.

Доказательство.

Первые два свойства очевидны. Третье вытекает из неравенства треугольника для точек на плоскости.

Конец доказательства.

Множество цепочек длины n не является метрическим пространством с метрикой $\rho^{'}(A,B)$.

Пример. Возьмем три цепочки длины 2: A и B пересекаются по одному звену, B и C не пересекаются, A и C не пересекаются. Для них неравенство треугольника не выполняется.

1.4. Область допустимых положений конца данной цепочки (существование цепочки с концом в данной точке)

Теорема. $A = A_0A_1...A_n$ - цепочка длины n. Область допустимых положений конца цепочки $\mathfrak{D}(A) = \{P : \max\{d^A_{0,1} - (d^A_{1,2} + ... + d^A_{n-1,n}), 0\} \leqslant |PA_0| \leqslant d^A_{0,1} + ... + d^A_{n-1,n}\}$

Доказательство.

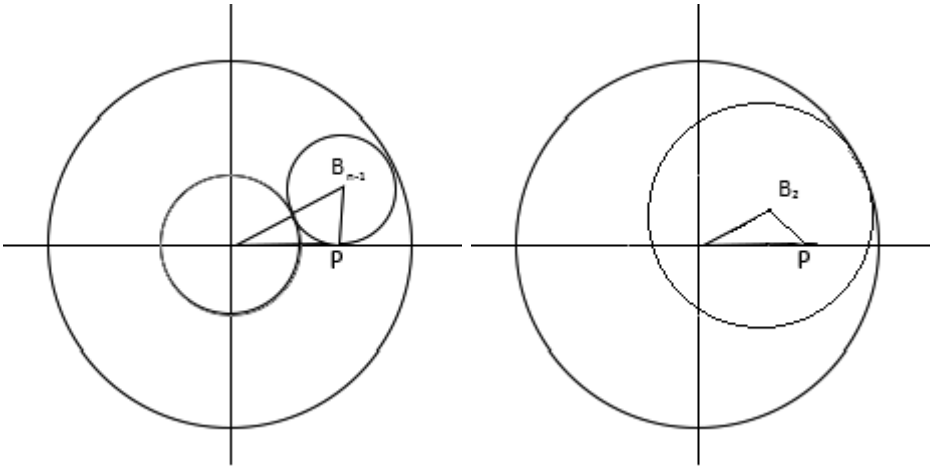


Рис 1. Область допустимых положений конца цепочки. Иллюстрация доказательства

Докажем утверждение для цепочки с центром в начале координат O. Для такой цепочки область равна $\mathfrak{D}(A) = \{P : \max\{d^A_{0,1} - (d^A_{1,2} + ... + d^A_{n-1,n}), 0\} \leqslant |P| \leqslant d^A_{0,1} + ... + d^A_{n-1,n}\}$. Для цепочек с центром в другой точке утверждение получается параллельным переносом.

Докажем, что для любой точки P из указанной области существует положение цепочки такое, что A_n и P совпадают.

Без ограничения общности будем считать, что точка лежит на оси OX, иначе совершим поворот вокруг начала координат.

Случай $\max\{d^A_{0,1} - (d_{1,2} + ... + d_{n-1,n}), 0\} > 0$: Проведём окружность радиуса $d_{1,2} + \ldots + d_{n-1,n}$ так, чтобы она касалась окружностей $|x| = d_{0,1} + \ldots + d_{n-1,n}$ и $|x| = \max\{d^A_{0,1} - (d^A_{1,2} + ... + d^A_{n-1,n}), 0\}$ и точки P. Для этого проведём следующие две вспомогательные окружности: окружность с центром в точке P и радиуса $d^A_{n-1,n}$, и окружность с центром в начале координат и радиуса $\frac{d^A_{0,1}+\ldots+d^A_{n-1,n}+\max\{d^A_{0,1}-(d^A_{1,2}+...+d^A_{n-1,n}),0\}}{2} = d^A_{0,1}$. Эти окружности пересекаются в одной или двух точках, либо совпадают. Возмём любую точку из их пересечения и примем за B_1. Точки $B_1, ...B_n$ лежат на отрезке PB_1. Точка B_n совпадает с P. Точка B_0 - начало координат. Таким образом, мы построили искомую цепочку.

Случай $\max\{d^A_{0,1} - (d^A_{1,2} + ... + d^A_{n-1,n}), 0\} = 0$: Проведём окружность радиуса $d^A_{1,2} + \ldots + d^A_{n-1,n}$ так, чтоб она проходила через точку P и касалась окружности $|x| = d^A_{0,1} + \ldots + d^A_{n-1,n}$. Центр этой окружности примем за B_1. Точки $B_1, ...B_n$ лежат на отрезке PB_1. Точка B_n совпадает с P. Точка B_0 - начало координат.

Покажем, что для любой точки вне данной области цепочки не существует.

Пусть $|P| > d^A_{0,1} + d^A_{1,2} + ... + d^A_{n-1,n}$. Это невозможно, т.к. $|A_0A_n| = |P| = |d^A_{0,1}\cos\beta^A_{0,1} + \ldots + d^A_{n-1,n}\cos\beta^A_{n-1,n}| \leqslant d^A_{0,1} + d^A_{1,2} + ... + d^A_{n-1,n}$, где $\beta^A_{i,i+1}$ - угол между отрезком ${}_iA_{i+1}$ и осью .

Пусть $\max\{d^A_{0,1} - (d^A_{1,2} + ... + d^A_{n-1,n}), 0\} > |P|$. Это противоречит теореме о том, что для любого треугольника ABC $|AC| < |AB| + |BC|$.

Конец доказательства.

Таким образом, для любой точки из области допустимых положений конца цепочки существует цепочка, полученная из данной перемещением конца в точку. Для остальных точек такой цепочки не существует.

1.5. Оценки

Теорема. $A = A_0A_1...A_n$ - цепочка длины n.

точка $P \in \mathfrak{D}(A)$
$B : B_n = P, B_0 = A_0$

$$\Downarrow$$

$$\theta_A(P,i) = \max\{0, PA_i - \sum_{j=i}^{n-1} d^A_{i,i+1}\}$$

$$\rho(A,B) \geq \sum_{i=1}^{n} \theta_A(P,i)$$

Доказательство.

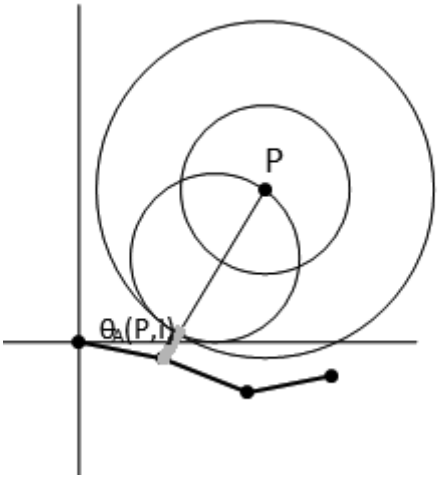


Рис 2. Величина $\theta_A(P,i)$

Фиксируем i.

Рассмотрим область допустимых положений конца цепочки C с началом в точке P и длины $n-1-i+1$, с длинами отрезков $d^A_{n-1,n}, d^A_{n-2,n-1}, ...$ $\theta_A(P,i)$ - расстояние до ближайшей точки на $\mathfrak{D}(C)$. Меньше, чем $\theta_A(P,i)$, расстояние между точками A_i и B_i быть не может, так как таких цепочек B не существует. Значит, $|B_iA_i| \geq \theta_A(P,i)$.

Следовательно, $\sum\limits_{i=1}^{n} |B_iA_i| \geq \sum\limits_{i=1}^{n} \theta_A(P,i)$.

Конец доказательства.

Теорема. $A = A_0A_1...A_n$ - цепочка длины n.

точка $P \in \mathfrak{D}(A)$
$B : B_n = P, B_0 = A_0$

$$\Downarrow$$

$$\phi_A(P,i) = \max\{0, PA_i + \sum_{j=i}^{n-1} d^A_{i,i+1}\}$$

$$\rho(A,B) \le \sum_{i=1}^{n} \phi_A(P,i)$$

Доказательство.

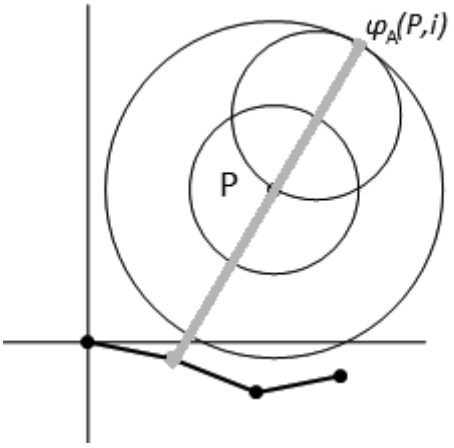


Рис 3. Величина $\phi_A(P,i)$

Фиксируем i.

Рассмотрим область допустимых положений конца цепочки C с началом в точке P и длины $n-1-i+1$, с длинами отрезков $d^A_{n-1,n}, d^A_{n-2,n-1}, \ldots$. $\phi_A(P,i)$ - расстояние до дальней точки на $\mathfrak{D}(C)$. Больше, чем $\phi_A(P,i)$, расстояние между точками A_i и B_i быть не может, так как таких цепочек B не существует. Значит, $|B_iA_i| \le \phi_A(P,i)$.

Следовательно, $\sum\limits_{i=1}^{n} |B_iA_i| \le \sum\limits_{i=1}^{n} \phi_A(P,i)$.

Конец доказательства.

1.6. Количество цепочек, отличающихся минимальным количеством звеньев от данной и полученных перемещением конца в заданную точку

Теорема. $A = A_0A_1...A_n$ - цепочка длины n.

точка $P \in \mathfrak{D}(A) \Rightarrow$

$\exists$ цепочка B, полученная переводом конца цепочки A в точку P и отличающая минимальным количеством звеньев от цепочки A.

Доказательство.

$\rho'(A, B)$ принимает конечное количество значений на множестве цепочек, полученных из данной перемещением конца. Значит, минимум достигается.

Конец доказательства.

$A = A_0A_1...A_n$ - цепочка длины n. Определим D_i следующим образом:

$D_{n-1} = \mathfrak{D}(A_{n-1}A_n)$

...

$D_k = \mathfrak{D}(A_k...A_n)\backslash D_{n-1}...\backslash D_{k+1}$

...

$D_0 = \mathfrak{D}(A)\backslash D_{n-1}...\backslash D_1$

Γ_k - граница D_k.

Лемма. $A = A_0A_1...A_n$ - цепочка длины n.

точка $P \in D_k$

$B = B_0B_1...B_n$ получена перемещением конца цепочки A в P.

Тогда:

$\rho'(A, B) = n - k$; A, B отличаются звеньями $A_kA_{k+1}, ..., A_{n-1}A_n$.

$\Leftrightarrow$

A, B отличаются минимальным количеством звеньев.

Доказательство.

$\Rightarrow$

$\rho'(A, B) \geq n - k$, т.к. $P \in D_k$ и любая цепочка, полученная перемещением конца цепочки A в P, отличается от A звеньями $A_kA_{k+1}, ..., A_{n-1}A_n$. $\rho'(A, B) = n - k$, значит, B отличается от A минимальным количеством звеньев.

$\Leftarrow P \in D_k$, поэтому A и B отличаются звеньями $A_kA_{k+1}, ..., A_{n-1}A_n$ ($|PA_{k+1}| > \sum_{i=k+1}^{n-1} |A_iA_{i+1}|$, то есть цепочки, где одно из этих звеньев совпадает, не существует) и только ими, потому что цепочки отличаются минимальным количеством звеньев. Значит, $\rho'(A, B) = n - k$

Теорема. $A = A_0A_1...A_n$ - цепочка длины n.

$P \in D_{n-1} \Rightarrow$ существеут единственная цепочка, полученная из A перемещением конца в точку P и отличающаяся минмальным количеством звеньев.

$P \in D_{n-2} \backslash \Gamma_{n-2} \Rightarrow$ существет две таких цепочки.

$P \in D_k \backslash \Gamma_k \backslash \Gamma_{k-1} (k \neq n-1, n-2) \Rightarrow$ существует континуум таких цепочек.

$P \in \Gamma_k \Rightarrow$ - существует единственная такая цепочка.

Доказательство.

По лемме, цепочка, полученная из A перемещением конца в точку P и отличающаяся минимальным количеством звеньев, отличается только своими конечными звеньями.

Если $P \in D_{n-1}$, мы меняем положение только последнего звена, и мы можем сделать это единственным образом.

Если $P \in D_{n-2} \tilde{} \Gamma_{n-2}$, то проведём две окружности: с центром в точке P и радиусом $|A_{n-1}A_n|$, и центром в точке A_{n-2} и радиусом $|A_{n-2}A_{n-1}|$. Поскольку точка не принадлежит границе области, существует две точки пересечения окружностей, определяющих две цепочки.

Если $P \in D_k \backslash \Gamma_k \backslash \Gamma_{k-1} (k \neq n-1, n-2)$, мы можем построить континуум цепочек, так как дуга окружности с центром в точке P и радиусом $|A_{n-1}A_n|$ имеет континуум точек, для каждой из которых можно построить как минимум одну цепочку.

Если $P \in \Gamma_k$, то существует единственная цепочка, у котрой последние звенья вытянуты в прямую.

1.7. Количество цепочек, находящихся на минимальном расстоянии от данной и полученных перемещением конца в заданную точку

Теорема. $A = A_0A_1...A_n$ - цепочка длины n.

точка $P \in \mathfrak{D}(A) \Rightarrow$

$\exists$ цепочка B, полученная переводом конца цепочки A в точку P и находящаяся на минимальном расстоянии от цепочки A.

Доказательство.

Рассмотрим множество цепочек C, полученных из A перемещением в точку P.

$C_0 = A_0$;

$C_1 \in \mathfrak{D}(A_0A_1) \cap \mathfrak{D}(PA_n...A_1)$;

...

$C_i \in \mathfrak{D}(A_0A_1...A_i) \cap \mathfrak{D}(PA_n...A_i)$;

Множество таких цепочек - компакт, евклидово расстояние и расстояние между центрами масс - непрерывная функция, следовательно, на нём достигаются минимум и максимум. Следовательно, существует цепочка, минимально удалённая от данной и полученная перемещением конца данной цепочки в P.

Теорема доказана.

Теорема. $A = A_0A_1A_2$ - цепочка длины 2.

точка $P \in \mathfrak{D}(A)$.

Если $P \in$ прямой A_0A_1, $P \notin$ границе $\mathfrak{D}(A)$, то $\exists$ ровно две цепочки, получаемых из A перемещением конца цепочки в точку P, и находящихся на минимальном расстоянии от цепочки A.

Иначе $\exists$ ровно одна такая цепочка.

Доказательство.

Проведём две окружности: Окружность с центром в точке A_0 и радиусом $d^A_{0,1}$; Окружность с центром в точке P и радиусом $d^A_{1,2}$.

Эти две окружности пересекаются по одной точке тогда и только тогда, когда $P \in$ границе $\mathfrak{D}(A)$. В этом случае существует только одна цепочка, получаемых из A перемещением конца цепочки в точку P, и находящихся на минимальном расстоянии от цепочки A. У неё оба звена находятся на одной прямой.

Иначе эти две окружности пересекаются по двум точкам M, N, и существуют ровно две цепочки, получаемые из A перемещением конца цепочки в точку P: A_0MP и A_0NP.

Нам требуется доказать, что $A_1M = A_1N \Leftrightarrow P \in$ прямой A_0A_1.

Необходимость: $A_1M = A_1N$. OP - высота равнобедренных треугольников ΔOMN и ΔMNP. ΔMNP построен на том же основании, и OP - также и его высота. Следовательно, $A_1 \in OP$.

Достаточность: В случае, если $P \in$ прямой A_0A_1, все точки и прямые симметричны относительно прямой A_0A_1. Поэтому $A_1M = A_1N$.

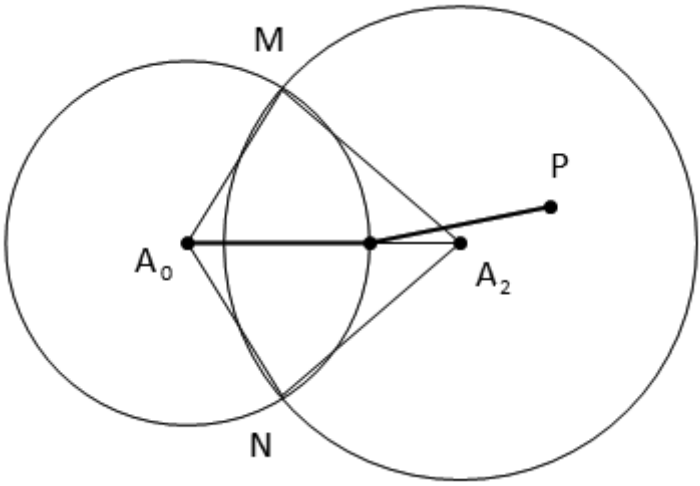


Рис. 4. Иллюстрация доказательства теоремы о количестве цепочек, полученных из данной перемещением в заданную точку и находящихся на минимальном расстоянии от данной.

Теорема доказана.

Теперь рассмотрим цепочки длины 3.

Пусть дана цепочка $A = A_0A_1A_2A_3$ и точка $P \in \mathfrak{D}(A)$. Мы будем рассматривать только цепочки с центром в начале координат и со звеном A_0A_1, лежащим на положительной части оси OX, потому что все остальные цепочки получаются из таких параллельным переносом и поворотом.

Построим цепочку B, которая получается из цепочки A перемещением конца в точку P.

Введём параметр t, обозначающий угол наклона B_0B_1 к оси OX. За t_0 обозначим угол наклона A_0A_1.

$$\begin{cases} x_{B_1} = d^A_{0,1} \cos t \\ y_{B_1} = d^A_{0,1} \sin t \end{cases}$$

Попробуем выразить положение точки A_2 через параметр t. Эта точка находится на пересечении следующих окружностей:

$$\begin{cases} (x - x_{B_1})^2 + (y - y_{B_1})^2 = d^A_{1,2} \\ (x - x_P)^2 + (y - y_P)^2 = d^A_{2,3} \end{cases}$$

Вычтем одно выражение из другого. Квадраты x и y сократятся и получится выражение:

$$y = a(t) + b(t)x$$

где

$a(t) = \frac{(d^A_{1,2})^2 - (d^A_{2,3})^2 - x_{B_1}(t)^2 + x_P^2 - y_{B_1}(t)^2 + y_P^2}{-2y_{B_1}(t) + 2y_P}$

$b(t) = \frac{-2x_{B_1}(t) + 2x_P}{-2y_{B_1}(t) + 2y_P}$

Подставим $y = a(t) + b(t)x$ в одно из изначальных уравнений. Получим:

$$A(t)x^2 - B(t)x - C(t) = 0$$

где

$A(t) = 1 + b(t)^2$

$B(t) = 2x_{B_1}(t) + 2a(t)b(t) - 2b(t)y_{B_1}(t)$

$C(t) = (d^A_{1,2})^2 - x_{B_1}(t)^2 - y_{B_1}(t)^2 - a(t)^2 + 2a(t)y_{B_1}(t)$

Дискриминант равен:

$$D(t) = B(t)^2 + 4A(t)C(t)$$

Уравнения имеют два решения, которые могут совпадать:

$$\begin{cases} x_{B_2^{'}}(t) = \frac{B(t) + \sqrt[2]{D(t)}}{2A(t)} \\ y_{B_2^{'}}(t) = a(t) + b(t)x \end{cases}$$

$$\begin{cases} x_{B_2^{''}}(t) = \frac{B(t) - \sqrt[2]{D(t)}}{2A(t)} \\ y_{B_2^{''}}(t) = a(t) + b(t)x \end{cases}$$

Это координаты, которые может иметь точка B_2 в зависимости от параметра t.

$Dist^A_1(t) = |A_1B_1(t)| + |A_2B_2^{'}(t)| + |A_3P|$

$Dist^A_2(t) = |A_1B_1(t)| + |A_2B_2^{''}(t)| + |A_3P|$

$Dist^A_{min}(t) = \min\{Dist^A_1(t), Dist^A_2(t)\}$ - расстояние между цепочкой A и цепочкой, получаемой из A перемещением конца цепочки в точку P и находящейся на минимальном расстоянии от A при данном t.

Гипотеза. $A = A_0A_1A_2A_3$ - цепочка длины 3. $P \in \mathfrak{D}(A)$.

1. Если A_0, A_1, A_2 лежат на одной прямой и точка P лежит на этой же прямой, но не на границе $\mathfrak{D}(A)$, то существует ровно два минимума $Dist^A_{min}(t)$, расположенных симметрично относительно t_0;

2. Иначе существует ровно один минимум $Dist^A_{min}(t)$.

Гипотеза подтверждается эмпирически построением соответствующих графиков программой в Wolfram Mathematica.

Теорема. $A = A_0A_1A_2A_3$ - цепочка длины 3. $P \in \mathfrak{D}(A)$.

Выполняется Гипотеза $\Rightarrow$ Существует не больше чем две цепочки, полученные переведением конца цепочки A в точку P и находящиеся на минимальном расстоянии от A.

Доказательство.

Разберём два случая:

1. Существует ровно один минимум $Dist^A_{min}(t)$. Тогда существует как максимум две цепочки длины 2 с центром в точке B_1, оканчивающиеся в P, и со звеньями длины $d^A_{1,2}$ и $d^A_{2,3}$. Поэтому всего существует не более двух цепочек, соответствующих условию.

2. Существует два симметричных минимума $Dist^A_{min}(t)$. Этим минимумам соответствуют точки $B_2^{'}$ и $B_2^{''}$. Для каждой из этих точек существует ровно одна цепочка длины 2 с центром в ней, оканчивающиеся в P, и со звеньями длины $d^A_{1,2}$ и $d^A_{2,3}$, так как точка P не лежит на прямой, содержащей её первое звено. Поэтому всего существует не более двух цепочек, соответствующих условию.

Теорема доказана.

Пример. В этом случае есть две симметричные цепочки, получаемые из A перемещением конца цепочки в точку P и находящиеся на минимальном расстоянии от A.

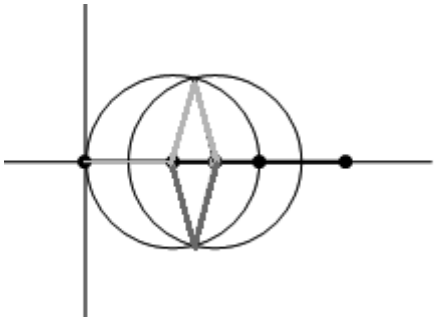


Рис.5. Пример цепочек, полученных перемещанием конца цепочки и находящихся на минимальном расстоянии от данной.

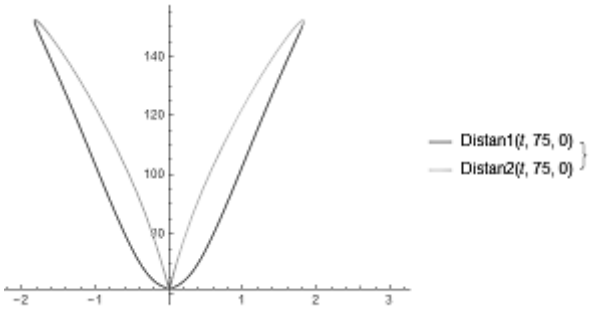


Рис.6. Функции $Dist_1^A(t), Dist_1^A(t)$ для этих цепочек.

Пример. В этом случае соответствующие цепочки не симметричны.

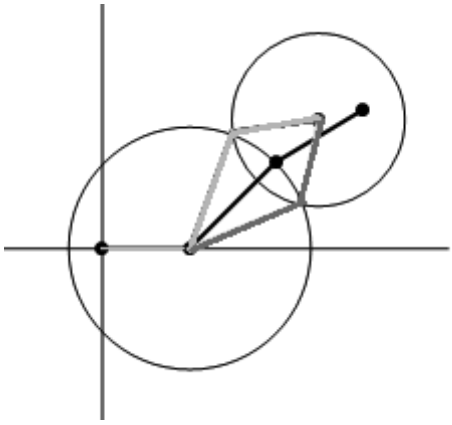


Рис.7. Пример цепочек, полученных перемещанием конца цепочки и находящихся на минимальном расстоянии от данной.

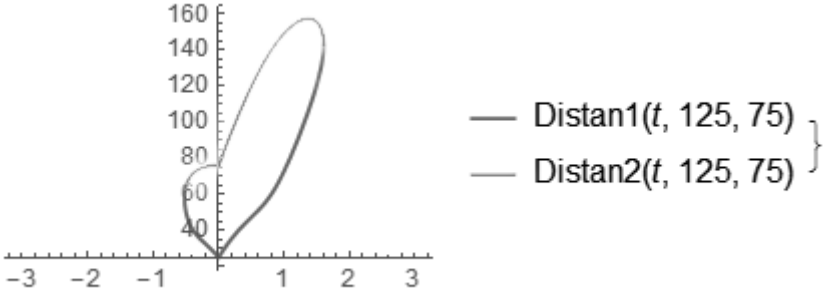


Рис.8. Функции $Dist_1^A(t), Dist_1^A(t)$ для этих цепочек.

1.8. Неэквивалентность задач минимизации расстояния и углового расстояния

Определение. *Угловое расстояние между цепочками* $A = A_0A_1...A_n$ *и* $B = B_0B_1...B_n$: $\rho_\alpha(A,B) = \sum\limits_{i=0}^{n-1} |\alpha^A_{i,i+1} - \alpha^B_{i,i+1}|$

Пусть дана цепочка A. Сводится ли задача минимизации функционала $\rho^A_\alpha(B) = \rho_\alpha(A,B)$ к минимизации функционала $\rho^A(B) = \rho(A,B)$?

Пример. Дана цепочка из большого количества звеньев, и её конец переходит в точку P путём поворота первого звена на небольшой угол. При этом все звенья сместятся на некоторое расстояние, которое в сумме будет довольно большим, так как звеньев много. Очевидно, чтоб минимизировать расстояние между цепочками, нужно оставить на месте почти все звенья и повернуть несколько последних. Но в этом случае звенья отклоняются на достаточно большой угол.

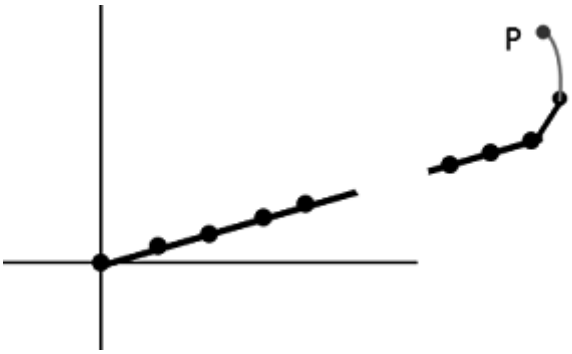


Рис.9.Пример цепочки, для которой задачи о минимизации углового расстояния и евклидова расстояния не эквивалентны.

Таким образом, задачу об минимизации углового расстояния следует рассматривать отдельно.

2. Алгоритмы

2.1. Приближённый алгоритм для евклидова расстояния

Пусть дана цепочка A длины и точка P. Этот алгоритм перебирает возможные цепочки, получаемые из A перемещением конца в точку P, с шагом угла ε. Из них он выбирает находящуюся на наименьшем евклидовом расстоянии от A.

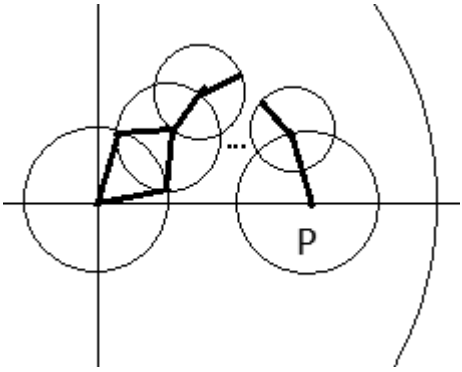


Рис.10. Иллюстрация одной итерации приближённого алгоритма.

`BuildChainLength2`(P_1, d_1, P_2, d_2) - функция, выдающая пару точек A и B таких, что P_1AP_2 и P_1BP_2 - цепочки с длинами звеньев d_1 и d_2.

`InAllowedArea(A,n,P)` - функция, выдающая $true$, если P находится в области допустимых положений конца цепочки A, и $false$ иначе.

`r(A,B)` - евклидово расстояние между цепочками.

`A1`(A, n, P, ε)

```
if !InAllowedArea(A, n, P)
    return (∞,∞)..(∞,∞)
minDist = ∞
 minChain=(∞,∞)..(∞,∞)
 B_n=P
 B_0 = A_0
 for t = (0,0,...0) to (2*π, 2*π, ...2*π) with step (0,0...ε)
     for i = 1 to n − 2
         B_i.x = B_{i−1}.x + d^A_{i−1,i} cos t
         B_i.y = B_{i−1}.y + d^A_{i−1,i} sin t
     {P_1, P_2}=BuildChainLength2(B_{n−2}, d^A_{n−2,n−1}, P, d^A_{n−1,n})
     if P_1A_1 < P_2A_2
         B_{n−1} = P_1
     else
         B_{n−1} = P_2
     minDist=r(A, B)
     minChain=B
```

Всего надо перебрать $2(\frac{2\pi}{\varepsilon})^{n-2}$ цепочек.

Алгоритм имеет сложность $O(e^n)$.

2.2. Линейный алгоритм

Данный алгоритм на каждом шагу выбирает точку B_i, максимально приближенную к A_i.

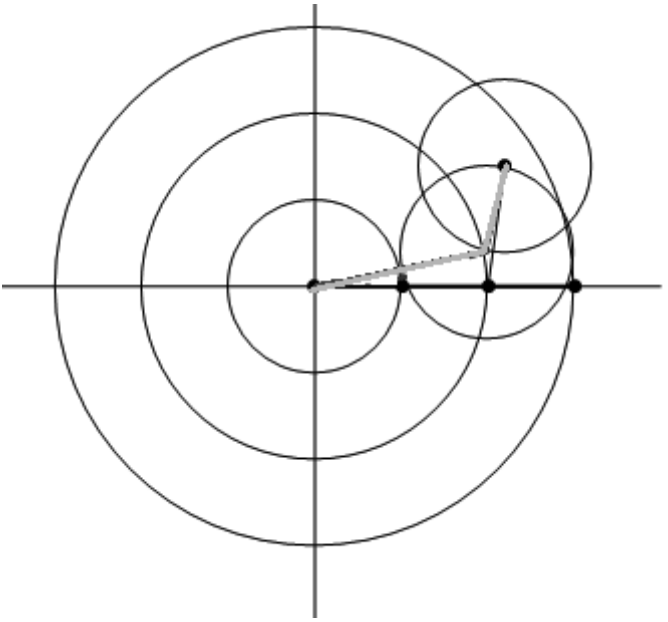


Рис.11. Решение, полученное с помощью линейного алгоритма.

A2(A, n, P)
    if !InAllowedArea(A, n, P)
        return $(\infty, \infty)..(\infty, \infty)$
    $B_n = P$
    $B_0 = A_0$
    for $i = n - 1$ to 2
        $Area_i$ = (окружность с центром в B_{i+1}, радиуса $d^A_{i,i+1} \cap \mathfrak{D}(A_0A_1...A_i)$)
        $B_i = arg\min_{C \in Area_i} |CA_i|$
    $\{P_1, P_2\}$=BuildChainLength2($B_2, d^A_{1,2}, A_0, d^A_{0,1}$)
    if $|P_1A_1| < |P_2A_1|$
        $B_1 = P_1$
    else
        $B_1 = P_2$
    return B

Сложность алгоритма - $O(n)$.

Сравним алгоритмы A1 и A2 с помощью программы. На следующих рисунках слева изображены цепочки, а справа - область таких P, что цепочки, получаемые в результате алгоритмов, совпадают с небольшой погрешностью:

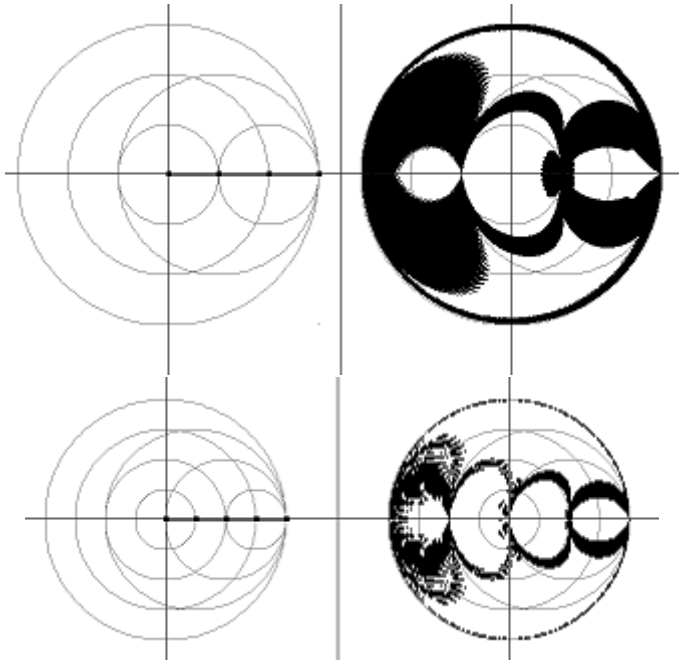
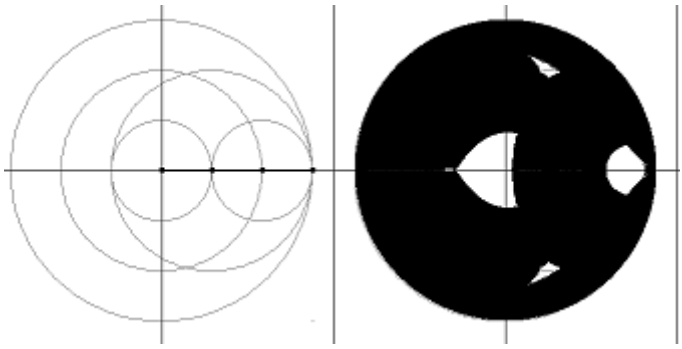


Рис.13. Область, где результаты работы алгоритмов A1 и A2 совпадают с небольшой погрешностью.

На следующих рисунках изображена область, на которой точки цепочек находятся в среднем на расстоянии половины звена друг от друга, то есть на небольшом расстоянии:



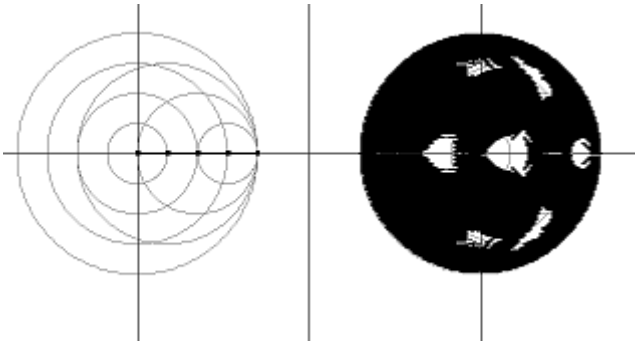
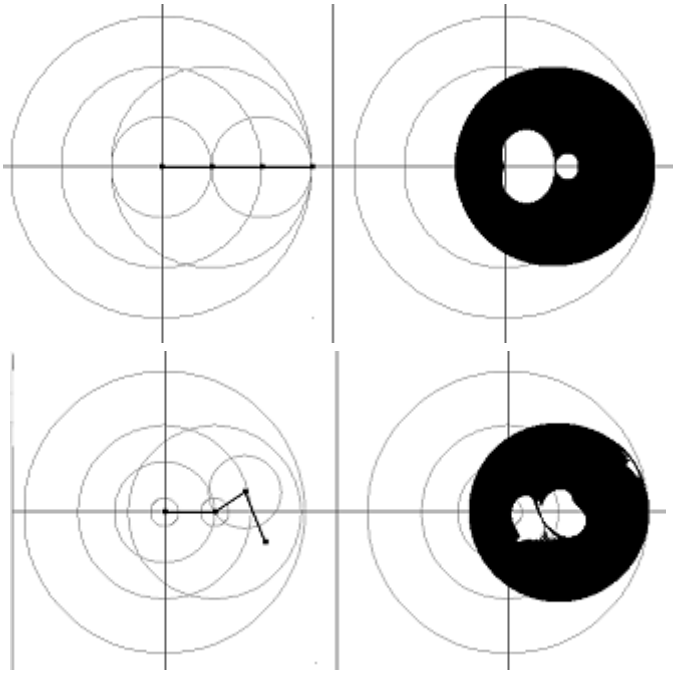


Рис.13. Область, где точки результатов работы алгоритмов A1 и A2 находятся на расстоянии половины звена в среднем.

2.3. Алгоритм для расстояния Хэмминга

Для начала внимательно рассмотрим результат работы алгоритма A1. Нарисуем область, в которой первое или первое и второе звенья остаются на месте:



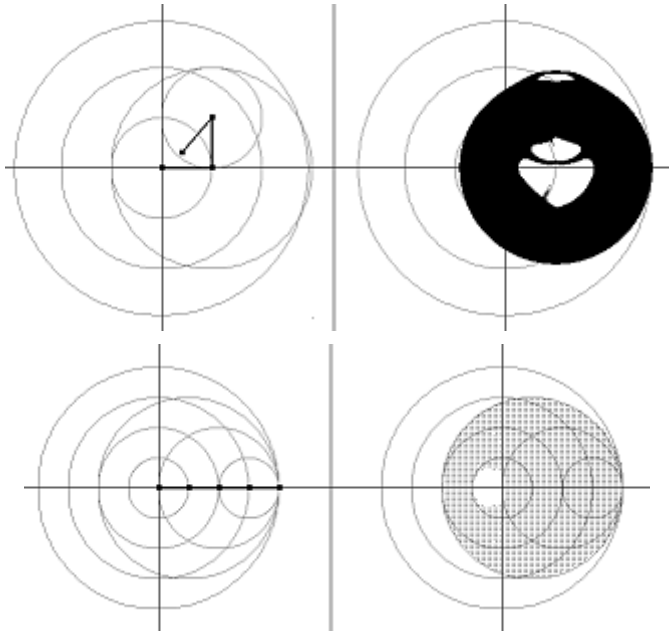


Рис.14. Область, где алгоритм A2 оставляет на месте первое звено.

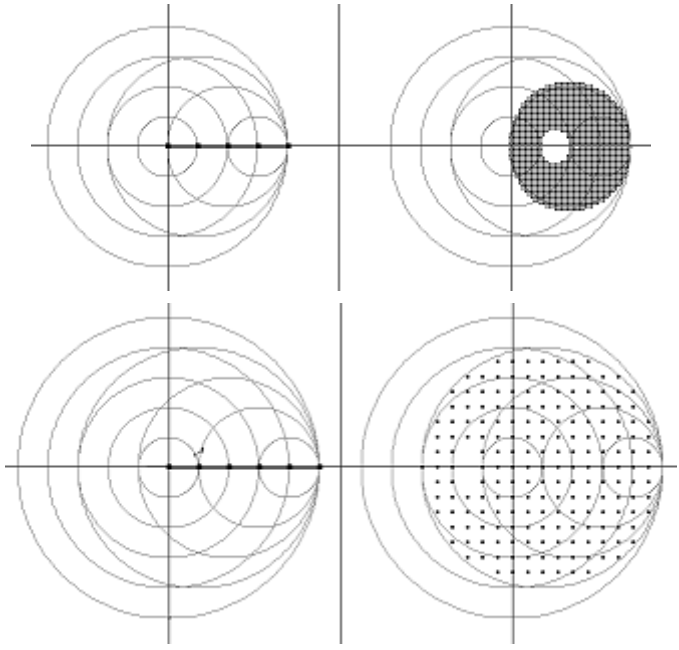


Рис.15. Область, где алгоритм A2 оставляет на месте первое и второе звено.

Как несложно заметить, эта область - почти вся область допустимых положений конца цепочки с центром в A_1 или A_2. Причём с ростом n эти области всё сильнее совпадают.

Попытаемся составить алгоритм, которые оставляет максимальное количество цепей на месте.

Обратим внимание, что на области вне области допустимых положений конца цепочки с центром в A_1 или A_2 алгоритм A2 достаточно эффективен.

Алгоритм оставляет максимально возможное количество звеньев. Оставшиеся звенья он находит по алгоритму A2.

`Intersection`(P_1, r_1, P_2, r_2) - выдаёт точки пересечения двух окружностей.

```
A3(A, n, P)
    if !InAllowedArea(A, n, P)
        return (∞, ∞)..(∞, ∞)
    B_n = P
    B_0 = A_0
    for i = n − 1 to 1
        if InAllowedArea(A_i...A_n, n, P)
            break
    for j = 0 to i
        B_j = A_j
    B_{i+1}...B_n = A2(A_{i+1}...A_n, n − i, P)
    return B
```

Сложность алгоритма - $O(n)$.

Теорема. Алгоритм для расстояния Хэмминга минимизирует расстояние Хэмминга.

Доказательство.

Построенная цепочка отличается от данной только совими последними звеньями, минимальным возможным их количеством. По лемме из параграфа про количество цепочек, отличающихся минимальным количеством звеньев, она отличается от данной минимальным количеством звеньев, т.е. она минимизирует расстояние Хэмминга между цепочками.

Теорема доказана.

Сравним результаты алгоритмов A1 и A3.

Область, где результаты работы алгоритмов находятся на небольшом расстоянии:

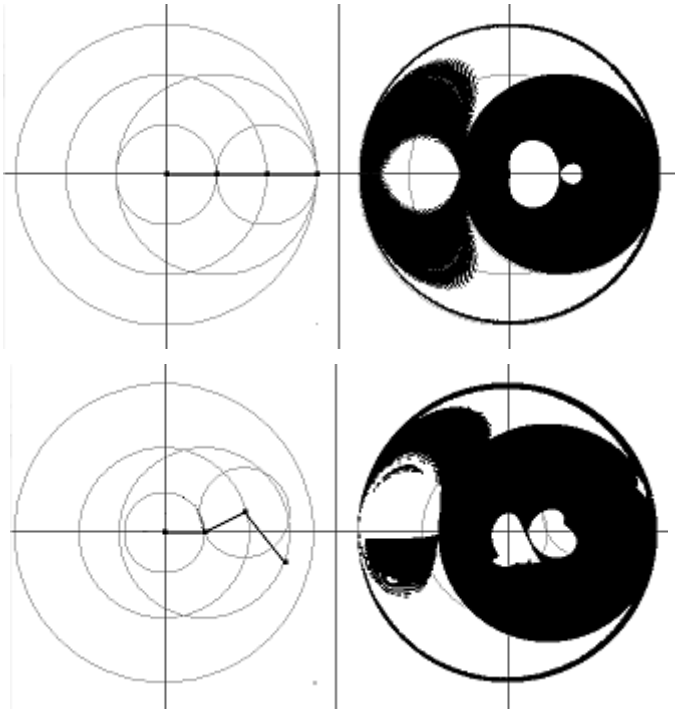
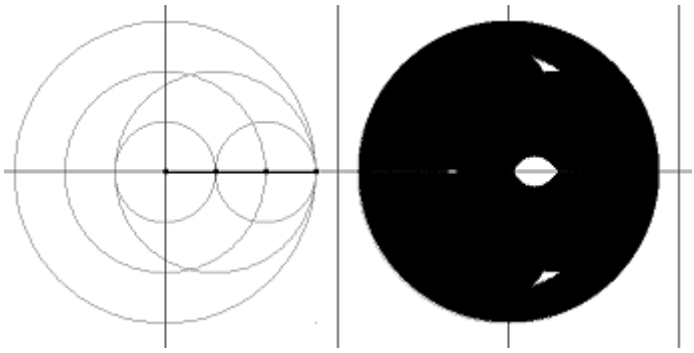


Рис.16. Область, где результаты работы алгоритмов A1 и A3 совпадают с небольшой погрешностью.

Область, где точки результатов находятся друг от друга в среднем на расстоянии половины звена:



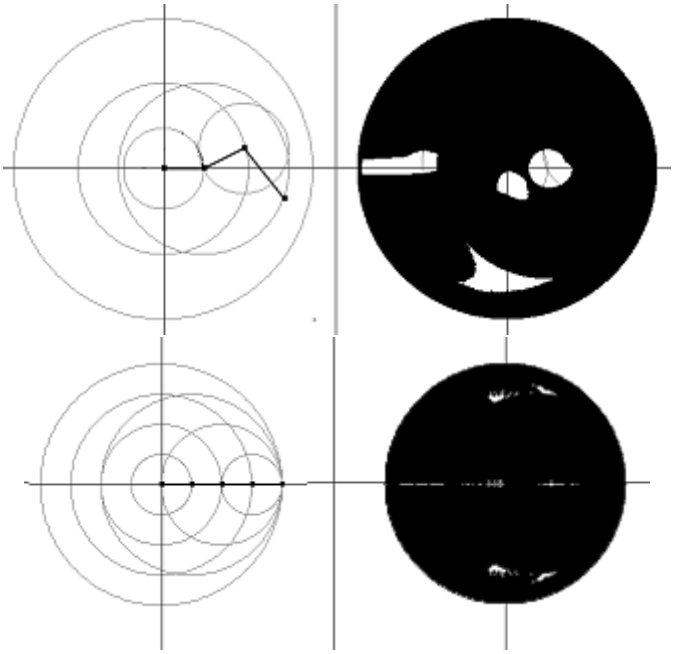


Рис.17. Область, где точки результатов работы алгоритмов A1 и A3 находятся на расстоянии половины звена в среднем.

Судя по наблюдаемым данным, алгоритм A3 для данных цепочек эффективней, чем алгоритм A2.

Список литературы

[1] Титова Е.Е. Конструирование движущихся изображений клеточными автоматами // Интеллектуальные системы. — 2014. — Т. 18, вып. 1. — С. 153–180.

[2] Иванов И.Е. О некоторых свойствах автоматов с магазинной памятью // Интеллектуальные системы. — 2014. — Т. 18, вып. 1. — С. 243–252.

[3] Кучеренко И.В. О минимизации монофункциональных классов бинарных клеточных автоматов с неразрешимым свойством обратимости. — 2014. — Т. 18, вып. 4. — С. 227–295.

[4] Якимец К.К. Об инвариантности характеристик конфигураций однородных структур. — 2014. — Т. 18, вып. 4. — С. 347–356.

[5] Иванов И.Е. О сохранении периодических последовательностей автоматами с магазинной памятью с однобуквенным магазином // Интеллектуальные системы. — 2015. — Т. 19, вып. 1. — С. 145–160.

[6] Летуновский А.А. Выразимость линейных автоматов относительно расширенной суперпозиции // Интеллектуальные системы. — 2015. — Т. 19, вып. 1. — С. 161–170.

[7] Гербус В.Г. О связи функций автомата и автоматной функции // Интеллектуальные системы. — 2015. — Т. 19, вып. 2. — С. 109–116.

[8] Миронов А.М. Критерий реализуемости функций на строках вероятностными автоматами Мура с числовым выходом // Интеллектуальные системы. — 2015. — Т. 19, вып. 2. — С. 149–160.

[9] Терехина И.Ю. Модель невлияния для квантовых автоматов // Интеллектуальные системы. — 2015. — Т. 19, вып. 2. — С. 183–190.

[10] Бабин Д.Н., Летуновский А.А. О возможностях суперпозиции, при наличии в базисе автоматов фиксированной добавки из булевых функций и задержки // Интеллектуальные системы. — 2015. — Т. 19, вып. 3. — С. 71–78.

[11] Бабин Д.Н. Автоматы с суперпозициями, пример нерасширяемости до предполного класса // Интеллектуальные системы. — 2015. — Т. 19, вып. 3. — С. 87–94.

[12] Э.Э.Гасанов, А.А.Мастихина Прогнозирование общерегулярных сверхсобытий автоматами // Интеллектуальные системы. — 2015. — Т. 19, вып. 3. — С. 127–154.

[13] А.А.Часовских. Критериальные системы в классах линейно-автоматных функций над конечными полями // Интеллектуальные системы. — 2015. — Т. 19, вып. 3. — С. 195–207.

[14] Гасанов Э.Э., Ефремов Д.В. Фоновый алгоритм решения двумерной задачи о доминировании // Интеллектуальные системы. — 2014. — Т. 18, вып. 3. — С. 133–158.

[15] Е. М. Перпер. Нижние оценки временной и объёмной сложности задачи поиска подслова // Дискретная математика, 2014, том 26:2, 58–70.

[16] Черемисин О. В. Об активности схем из клеточных элементов, реализующих систему всех конъюнкций // Дискретная математика. — 2003. — Т. 15, вып. 2. — С. 113—122

[17] Калачев Г. В. Порядок мощности плоских схем, реализующих булевы функции // Дискретная математика. — 2014. — Т. 26, № 1. — С. 49–74.

[18] Калачев Г. В. Нижние оценки мощности плоских схем, реализующих частичные булевы операторы // Интеллектуальные системы. Теория и приложения. — 2014. — Т. 18, № 2. — С. 279–322.

[19] Калачев Г. В. Об одновременной минимизации площади, мощности и глубины плоских схем, реализующих частичные булевы операторы // Интеллектуальные системы. Теория и приложения. — 2016. — Т. 20, № 2. — С. 203–266.

Algorithms of moving of the end of chain to the given point

Berger I.O.

The problem of chains is investigated.

Results on the existence of chains obtained from given chain by moving of the end of the chain to a given point; Bounds of the minimum of the Euclidean distance between chains obtained from each other by moving of the end to a given point; possible number of chains obtained by moving the end to a given point and differing by the minimum number of elements from a given chain; the possible number of chains that are at the minimum distance from the given and obtained by moving the end of the chain to a given point, for $n = 2$ and $n = 3$.

Algorithms for moving the end of a chain to a given point are described: an exponential algorithm that sorts out all possible chains with step ε, a linear algorithm giving an approximate solution for Euclidean distance, and a linear algorithm giving an exact answer for the Hamming distance and approximate for the Euclidean distance.

Keywords: chain, algorithm, upper bounds, lower bounds, Euclidean distance, Hamming distance.

Протоколы безопасности
часть 1

А.М. Миронов

Излагаются основные криптографические примитивы, используемые в протоколах безопасности (симметричные и асимметричные системы шифрования, хэш-функции, схемы разделения секрета), протоколы аутентификации, и алгоритмы цифровой подписи.
Ключевые слова: протоколы, безопасность, криптография, хэш-функции, аутентификация, цифровая подпись

1. Введение

Данная статья является первой частью обзорного текста по **протоколам безопасности** (**security protocols**), называемых также **криптографическими протоколами**. В ней изложены основные криптографические примитивы, используемые в протоколах безопасности (симметричные и асимметричные системы шифрования, хэш-функции, схемы разделения секрета), протоколы аутентификации, и алгоритмы цифровой подписи. Во второй части будут изложены протоколы распределения криптографических ключей, протоколы голосования, протоколы электронных платежей, и некоторые другие протоколы. Более подробное изложение основных протоколов безопасности можно найти в текстах [1]-[5].

1.1. Понятие протокола безопасности

Протокол безопасности (ПБ), называемый также просто **протоколом** – это распределённый алгоритм, определяющий порядок обмена сообщениями между несколькими **агентами**, целью которого является обеспечение безопасности передачи, обработки и хранения информации в небезопасной среде. В качестве агентов, принимающих участие в работе протокола (их называют также **участниками** этого протокола) могут выступать, например, люди, компьютерные программы, вычислительные комплексы, базы данных, сети связи, банковские карточки, и т.д.

Действия, выполняемые участниками протокола, имеют следующий вид: **посылка сообщения** другому участнику (или группе участников), **приём сообщения** от другого участника, **внутренние действия** (проверка логических условий, обновление значений переменных).

Свойства безопасности, которые должен обеспечивать ПБ, могут иметь, например, следующий вид:

- **целостность** передаваемых сообщений, которая заключается в том, что всякое изменение сообщений в процессе их передачи будет обнаружено в ходе выполнения протокола,

- **секретность** передаваемых сообщений, которая заключается в отсутствии неавторизованной утечки информации в процессе работы протокола.

В последующих пунктах этого параграфа мы излагаем основные концепции и конструкции, используемые при построении ПБ.

1.2. Шифрование сообщений

1.2.1. Понятие шифрования и дешифрования

Некоторые из сообщений, пересылаемых участниками ПБ, могут быть зашифрованными. Шифрование сообщений делается для того, чтобы противник, которому станут доступны пересылаемые сообщения, не смог ознакомиться с их содержанием.

Шифрование сообщения m представляет собой применение некоторого алгоритма $Encrypt$ (называемого **алгоритмом шифрования**) к паре (k, m), где k – битовая строка, называемая **ключом шифрования** (или просто **ключом**). Мы будем обозначать результат применения алгоритма $Encrypt$ к паре (k, m) записью $k(m)$ и называть её **шифртекстом (ШТ)** сообщения m на ключе k.

Для того, чтобы извлечь из ШТ $k(m)$ исходное сообщение m, должен быть задан алгоритм $Decrypt$, называемый **алгоритмом дешифрования**, и обладающий следующими свойствами:

- алгоритм $Decrypt$ получает на вход пару вида (d, u), где d – битовая строка, называемая **ключом дешифрования**, и u – дешифруемое сообщение, результат применения $Decrypt$ к паре (d, u) обозначается записью $d(u)$,

- каждому ключу шифрования k должен соответствовать ключ дешифрования d_k, такой, что для каждого сообщения m верно равенство $d_k(k(m)) = m$.

1.2.2. Системы шифрования

Система шифрования (СШ) представляет собой набор следующих данных:

- пара алгоритмов *Encrypt* и *Decrypt* шифрования и дешифрования соответственно, и

- набор ограничений, которым должны удовлетворять ключи шифрования и дешифрования, а также шифруемые сообщения.

Системы шифрования принято подразделять на два класса: симметричные и асимметричные.

1) В **симметричных** СШ **(ССШ)** каждый ключ шифрования k совпадает с соответствующим ему ключом дешифрования d_k. Как правило, алгоритмы шифрования и дешифрования в ССШ тоже совпадают.

   Если ССШ используется для связи между несколькими агентами, то ключ шифрования этой ССШ, который используется в текущий момент, обозначается записью $k_{a_1 \ldots a_n}$, где $a_1, \ldots, a_n$ – список всех агентов, которым известен этот ключ. Если после использования этого ключа агенты $a_1, \ldots, a_n$ перейдут на новый ключ, то этот новый ключ будет обозначаться записью $k'_{a_1 \ldots a_n}$.

2) В **асимметричных** СШ **(АСШ)** ключи шифрования могут отличаться от соответствующих им ключей дешифрования, причем

   - ключи шифрования и дешифрования в АСШ, как правило, связаны с конкретными агентами, мы будем обозначать эти ключи записями a^+ и a^- соответственно, где a – идентификатор агента, с которым связаны эти ключи, и
   - ключ a^+ является **открытым** (т.е. известен всем агентам), в то время как ключ a^- **закрыт**: он не должен быть известен никому кроме агента a.

Некоторые АСШ обладают следующим свойством: для каждого сообщения m верно равенство $a^+(a^-(m)) = m$. Такие АСШ можно использовать для **аутентификации** (т.е. проверки подлинности) агентов: если какой-либо агент a, использующий эту АСШ, хочет доказать, что он действительно является тем агентом, имя которого – a, то он может сделать это, если для любого предъявляемого ему сообщения m он сможет вычислить сообщение $m' \stackrel{\text{def}}{=} a^-(m)$. Подлинность a будет доказана, если выполнено условие

$$a^+(m') = m. \tag{1}$$

Данный метод доказательства подлинности обосновывается предположением о том, что

- без знания закрытого ключа a^- создать сообщение m', удовлетворяющее условию (1), невозможно, и
- никто, кроме агента a, не знает ключ a^-.

1.3. Формальные описания и примеры протоколов

1.3.1. Понятие формального описания протокола

Одним из простейших видов формального описания ПБ является список P записей вида

$$a \to b : [\![\varphi]\!]\, m, \tag{2}$$

$$a \to \{b_1, \ldots, b_n\} : [\![\varphi]\!]\, m, \tag{3}$$

или

$$a : [\![\varphi]\!] \text{ внутреннее действие}, \tag{4}$$

объединенных фигурной скобкой слева, где a, b, $b_1, \ldots, b_n$ – имена агентов, φ – условие, m – выражение, значением которого является сообщение (**сообщением** мы называем любой объект, который один из участников ПБ передаёт другому в процессе работы ПБ, этот объект может быть как символьной строкой, так и набором банкнот, товаром, и т.п.). Записи (2), (3) и (4) изображают действия, которые должны выполнять агенты в процессе работы ПБ.

Действия, входящие в список P, выполняются последовательно, их выполнение происходит следующим образом:

- (2) и (3) выполняется путем проверки φ, и

– если φ выполнено, то a посылает m агенту b (в случае (2)), или агентам $b_1, \ldots, b_n$ (в случае (3)),

– если φ не выполнено, то это действие пропускается, и выполняется следующее по списку действие,

- (4) выполняется аналогично, только в данном случае происходит не посылка сообщения от a, а вычисления и изменение значений переменных агента a.

Если φ выполнено всегда, то компонента $[\![\varphi]\!]$ в записи действий опускается.

Если текущее исполняемое действие не относится к какому-либо из участников ПБ, то этот участник не функционирует в момент исполнения этого действия.

В формальных описаниях протоколов мы будем использовать следующее соглашение: запись вида $x := e$, где x – переменная, и e – выражение, обозначает переменную x, значение которой равно значению выражения e.

1.3.2. Пример формального описания протокола

В этом пункте мы рассмотрим пример формального описания протокола, решающего задачу продажи компьютера агента a агенту b. Мы предполагаем, что у a есть компьютер, а у b есть деньги, и b хочет на эти деньги купить у a компьютер. Агенты a и b не доверяют друг другу, поэтому протоколы продажи компьютера, имеющие вид

$$\begin{cases} a \to b: \text{ компьютер} \\ b \to a: \text{ деньги} \end{cases} \quad \text{или} \quad \begin{cases} b \to a: \text{ деньги} \\ a \to b: \text{ компьютер} \end{cases}$$

для них неприемлемы: каждый из них не верит, что если он выполнит первое действие в первом или втором протоколе, то его коллега обязательно выполнит второе действие.

Одним из возможных решений задачи продажи компьютера агента a агенту b может быть протокол, в котором, помимо a и b, принимает участие доверенный посредник I. Данный протокол может иметь, например,

следующий вид:

$$\begin{cases} a \to I: \ \text{компьютер} \\ b \to a: \ \text{деньги} \\ a \to I: \begin{pmatrix} \text{подтверждение или опровержение} \\ \text{того, что полученная от } b \text{ сумма} \\ \text{соответствует стоимости компьютера} \end{pmatrix} \\ I \to b: \ [\![\text{от } a \text{ поступило подтверждение}]\!] \ \text{компьютер} \\ I \to a: \ [\![\text{от } a \text{ поступило опровержение}]\!] \ \text{компьютер} \end{cases}$$

a и b могут считать данный протокол приемлемым, например, по следующим причинам:

- a верит, что до окончания проверки денег I не передаст компьютер агенту b, и I вернёт компьютер a, если b передаст a недостаточную сумму,
- b верит, что пока a не пошлёт I подтверждение, компьютер будет находиться у I, и сразу после того, как a пошлёт I подтверждение, I передаст b компьютер.

Однако данный протокол некорректно работает в том случае, когда какой-либо из агентов ведёт себя нечестно (например, b посылает a правильную сумму, но a посылает опровержение, получает обратно свой компьютер, и не отдает b полученные от него деньги).

1.4. Уязвимости протоколов

1.4.1. Понятие уязвимости протокола

Нарушения свойств безопасности в процессе работы ПБ могут происходить по причине противодействия со стороны агентов, называемых **противниками**.

Противники подразделяются на следующие два класса:

- **пассивные противники**, они могут перехватывать сообщения, пересылаемые участниками ПБ, и анализировать их,
- **активные противники**, они могут делать то же, что и пассивные противники, а также модифицировать или удалять перехваченные сообщения, генерировать новые сообщения и посылать их участникам ПБ вместо перехваченных, выдавать себя за участников ПБ.

Также нарушения свойств безопасности ПБ возможны из-за действий участников ПБ, которые нарушают (умышленно или неумышленно) предписанные протоколом правила взаимодействия с другими участниками этого ПБ.

Возможности нарушения свойств безопасности в процессе работы ПБ называют **уязвимостями** этого ПБ.

При решении задач анализа уязвимостей ПБ используются следующие предположения о противнике:

- противник активен и полностью знает ПБ,
- противник не может извлечь из каждого перехваченного ШТ $k(m)$ сообщения m, если он не знает k.

1.4.2. Пример уязвимости протокола

В этом пункте мы рассмотрим пример уязвимости ПБ, который использовался много лет в банковских транзакциях. Данный протокол очень прост, он состоит всего из трёх действий, и сначала его правильность не вызывала сомнений. Однако после 15 лет его использования выяснилось, что он содержит уязвимость (которая была обнаружена автоматической системой верификации). Этот ПБ называется **протоколом Нидхема--Шредера** (Needham-Schroeder, 1979), мы будем обозначать его записью NS. Целью данного ПБ является взаимная аутентификация агентов a и b.

Предполагается, что a и b используют общую АСШ.

Протокол NS имеет следующий вид:

$$\begin{cases} a \to b : b^+(a, r_a) \\ b \to a : a^+(r_a, r_b) \\ a \to b : b^+(r_b) \end{cases} \tag{5}$$

где r_a и r_b – **уникальные значения**, называемые в литературе по протоколам безопасности **нонсами** (**nonces**, что является сокращением от **number used once**), каждое из которых представляет собой большую (несколько сотен битов) псевдослучайную строку, сгенерированную агентом a и b соответственно. Мы предполагаем что все нонсы, сгенерированные в различных сеансах выполнения любого ПБ, являются различными и отличаются от других значений. Нонсы обозначаются символом r (возможно, с индексами).

Действия, входящие в ПБ NS, имеют следующий смысл.

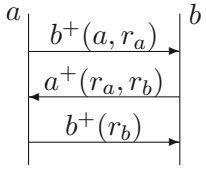
- Первое действие заключается в пересылке от a к b ШТ $b^+(a, r_a)$, получаемого шифрованием на ключе b^+ пары, состоящей из имени агента a и нонса r_a, который используется для того, чтобы дать возможность b доказать свою подлинность путем извлечения данного нонса из полученного ШТ.

- Второе действие заключается в пересылке от b к a ШТ $a^+(r_a, r_b)$, получаемого шифрованием на ключе a^+ пары, состоящей из нонса r_a, который b извлёк из полученного ШТ, и нонса r_b, сгенерированного агентом b. После его получения a убеждается в подлинности b, т.к. никто кроме b не может извлечь r_a. Нонс r_b предназначен для того, чтобы дать возможность a тоже доказать b свою подлинность.

- Третье действие заключается в пересылке от a к b ШТ $b^+(r_b)$. После получения r_b агент b убеждается в подлинности агента a.

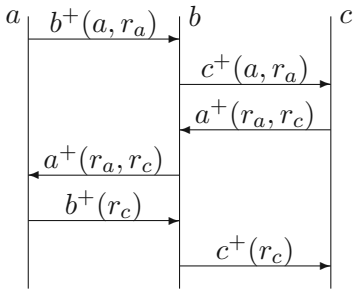
Таким образом, после успешного завершения данного протокола a и b будут убеждены в подлинности друг друга.

Для формальной записи ПБ используется также нотация в виде диаграмм. Каждое действие в ПБ, связанное с пересылкой сообщений, изображается в диаграмме горизонтальной стрелкой, помеченной пересылаемым сообщением. Диаграмма, соответствующая протоколу (5), имеет вид



Одна из уязвимостей данного ПБ связана с тем, что агент b может использовать свой статус участника ПБ NS для того, чтобы во взаимодействии с другими агентами выдавать себя за агента a. Использование агентом b своего статуса участника ПБ NS для совершения мошенниче-

ских действий можно изобразить следующей диаграммой:



Данную диаграмму можно рассматривать как объединение двух диаграмм, изображающих выполнение двух сеансов выполнения ПБ NS: в первом сеансе участвуют агенты a и b, а во втором – агенты b и c. Действия a, b и c во время выполнения этих сеансов выглядят следующим образом.

- После того, как a и b выполнят первое действие первого сеанса, агент b создает с использованием полученного нонса r_a ШТ $c^+(a, r_a)$, и посылает этот ШТ агенту c в качестве своего первого действия во втором сеансе ПБ NS (выдавая себя за a).

- Получив этот ШТ и дешифруя его, агент c приходит к выводу, что этот ШТ был послан агентом a. Согласно протоколу NS, агент c генерирует нонс r_c и посылает ответный ШТ $a^+(r_a, r_c)$.

- b пересылает полученный от c ШТ $a^+(r_a, r_c)$ агенту a.

- a рассматривает полученный от b ШТ как ответ, который b должен послать ему в соответствии со вторым шагом ПБ NS, т.е. a полагает, что извлечённый из полученного ШТ нонс r_c был сгенерирован агентом b.

- Согласно третьему шагу NS, a посылает b ШТ $b^+(r_c)$.

- b извлекает из полученного ШТ нонс r_c, и посылает агенту c ШТ $c^+(r_c)$.

После этого c верит, что тот агент, с которым он выполнял этот сеанс ПБ NS, является агентом a.

1.5. Другие примеры протоколов

1.5.1. Обедающие криптографы

Рассмотрим следующую задачу.

За круглым столом сидят три криптографа и обедают. После того, как они пообедали и хотят заплатить, официант сообщает им, что их обед уже оплачен, но не уточняет, кто именно платил. Возможен один из двух вариантов: обед оплатил один из криптографов, или обед оплатила ФСБ.

Криптографы хотят выяснить, какой именно из вариантов имеет место, причём, если имеет место первый вариант, то те криптографы, которые не платили, не должны узнать, кто же конкретно оплатил их обед.

Для решения этой задачи предлагается следующий ПБ.

Поскольку участники сидят за круглым столом, то каждая пара соседей может подбрасывать монету между собой, так, чтобы результат был известен только им двоим. Подбрасывание монеты двумя участниками можно рассматривать как получение ими сообщения, состоящего из одного случайным образом порождённого бита, от доверенного посредника. После того, как все три пары подбросили монету, каждый участник знает результаты двух подбрасываний (решка или орёл). Эти результаты могут быть

- либо одинаковыми (т.е. оба раза была решка, или оба раза был орёл),
- либо разными (т.е. при одном подбрасывании была решка, а при другом - орёл).

Каждый участник говорит другим “одинаково” или “по разному”, причём тот, кто заплатил, говорит противоположное (т.е. если надо сказать “одинаково” то он говорит “по-разному”, и наоборот). Если число ответов “по разному” чётно, то это значит, что обед оплатила ФСБ, иначе - один из них.

Одна из уязвимостей этого протокола заключается в отсутствии контроля честности участников.

1.5.2. Протокол с подтверждением приёма

Рассмотрим следующую ситуацию: агенты a и b используют общую АСШ для секретного обмена сообщениями, и для контроля правильности пере-

дачи каждое получаемое сообщение отсылается назад отправителю (чтобы отправитель был уверен, что сообщение получено в неискажённом виде), согласно следующему протоколу:

$$\begin{cases} a \to b : b^+a^-(m) \\ b \to a : a^+b^-(m) \end{cases}$$

(из полученного ШТ извлекается сообщение m и возвращается отправителю в зашифрованном виде в качестве подтверждения приёма).

Данный протокол уязвим к следующей атаке: активный противник e перехватывает первое сообщение (обозначим его m') и посылает его b (от своего имени). Согласно протоколу, b посылает e ответное сообщение $e^+b^-e^+b^-(m')$ (т.е. $e^+b^-e^+a^-(m)$), из которого e сможет извлечь m.

1.5.3. Вычисление суммы

Рассмотрим следующую задачу: агенты $a_1, \ldots, a_n$ имеют числа $x_1, \ldots, x_n$ соответственно. Они хотят вычислить $\sum\limits_{i=1}^{n} x_i$, причём каждый агент a_i не хочет разглашать своё число x_i. Один из протоколов для решения данной задачи имеет следующий вид:

$$\begin{cases} a_1 \to a_2 : {a_2}^+(x_1 + r), \text{ где } r \text{ – случайное целое число} \\ a_2 \to a_3 : {a_3}^+(x_2 + x_1 + r) \\ \ldots \\ a_n \to a_1 : {a_1}^+(x_n + \ldots + x_1 + r) \\ a_1 \text{ вычитает } r \text{ из полученного результата} \end{cases}$$

Недостаток этого протокола – нет контроля честности участников.

1.5.4. Сравнение двух чисел

Излагаемый ниже протокол предназначен для решения следующей задачи: агенты a и b имеют числа x и y соответственно (предполагается, что $x, y \in \{1, \ldots, 100\}$), они хотят узнать без разглашения чисел x и y, верно ли, что $x \leq y$.

Один из ПБ для решения этой задачи имеет следующий вид: (предполагается, что a и b используют общую АСШ)

$$\begin{cases} a \to b : z - x, \text{ где } z = b^{+}(r),\ r \text{ – случайное целое число} \\ b \text{ вычисляет } \{z_i := b^{-}(z - x + i) \mid i = 1, \ldots, 100\}, \\ \quad \text{и проверяет условие } \forall\, i \neq j \quad |z_i - z_j| \geq 2 \\ \quad \text{(если это условие не вып., то } b \text{ просит } a \text{ начать} \\ \quad \text{выполнение протокола заново и выбрать новое } r) \\ b \to a : \ z_1, \ldots, z_y, z_{y+1} + 1, \ldots, z_{100} + 1 \\ a \to b : \text{ответ} = (x \leq y), \text{ если } r = x\text{-й член этой посл–сти.} \end{cases}$$

Данный протокол тоже содержит уязвимости.

2. Необходимые математические сведения

2.1. Кольцо вычетов по модулю n

Ниже символ $\mathbf{Z}$ обозначает множество целых чисел.

Пусть n – положительное целое число. Обозначим записью $\mathbf{Z}_n$ множество $\{0, 1, \ldots, n-1\}$. $\forall\, a \in \mathbf{Z}$ существует единственное число $r \in \mathbf{Z}_n$, такое, что $\exists\, q \in \mathbf{Z} : a = nq + r$. Данное число называется **остатком** от деления a на n и обозначается записью $(a)_n$.

$\forall\, a, b \in \mathbf{Z}$ запись $a \underset{n}{=} b$ означает, что $(a)_n = (b)_n$.

Множество $\mathbf{Z}_n$ является коммутативным кольцом, операции сложения и умножения на котором обозначаются записями $\underset{n}{+}$ и $\underset{n}{\cdot}$ соответственно, и определяются следующим образом: $\forall\, a, b \in \mathbf{Z}_n$, $a \underset{n}{+} b \stackrel{\text{def}}{=} (a + b)_n$, $a \underset{n}{\cdot} b \stackrel{\text{def}}{=} (ab)_n$. $\forall\, a \in \mathbf{Z}_n$ запись $\underset{n}{-} a$ обозначает число $n - a$ (если $a \neq 0$) и 0, если $a = 0$. Ниже числа $a \underset{n}{+} b$, $a \underset{n}{+} (\underset{n}{-} b)$ и $a \underset{n}{\cdot} b$ могут обозначаться записями $a + b$, $a - b$ и ab соответственно.

Подмножество множества $\mathbf{Z}_n$, состоящее из чисел, взаимно простых с n, является группой относительно операции умножения на $\mathbf{Z}_n$, её нейтральным элементом является число 1. Данная группа обозначается записью $\mathbf{Z}_n^*$. Число элементов в $\mathbf{Z}_n^*$ называется **функцией Эйлера** числа n, оно обозначается записью $\varphi(n)$ и равно $n(1 - \frac{1}{p_1}) \ldots (1 - \frac{1}{p_k})$, где

$p_1, \ldots, p_k$ – различные простые делители n. Элемент $g \in \mathbf{Z}_n^*$ называется **примитивным**, если $\mathbf{Z}_n^* = \{g^i \mid i \geq 1\}$.

Элементы $\mathbf{Z}_2$ $(= \{0,1\})$ называются **битами**. Для каждого $n \geq 1$ числа из $\mathbf{Z}_{2^n}$ в двоичной записи содержат не более n бит, в некоторых случаях мы будем отождествлять их с битовыми последовательностями длины n, т.е. с элементами $\mathbf{Z}_2^n$. Если $n = kl$, то элементы $\mathbf{Z}_{2^n}$ можно отождествлять с последовательностями l битовых блоков длины k, т.е. с элементами $\mathbf{Z}_{2^k}^l$.

Для каждого положительного $n \in \mathbf{Z}$ запись $|n|$ обозначает количество битов в двоичной записи n. $\forall\, m, n \in \mathbf{Z}$ запись $m|n$ означает, что m является делителем n. Для каждого множества M запись $|M|$ обозначает число элементов множества M.

$\forall\, g \in \mathbf{Z}_n^*$ запись $ord(g)$ обозначает порядок элемента g в группе $\mathbf{Z}_n^*$, т.е. наименьшее целое $n > 0$, такое, что $g^n \underset{\varphi(n)}{=} 1$.

Элемент g группы $\mathbf{Z}_n^*$ называется **порождающим**, если $\mathbf{Z}_n^* = \{g^i \mid i \geq 0\}$ (т.е. $ord(g) = |\mathbf{Z}_n^*|$).

$\forall\, g \in \mathbf{Z}_n^*$ запись g^{-1} обозначает элемент группы $\mathbf{Z}_n^*$, являющийся обратным к g. $\forall\, g \in \mathbf{Z}_n^*, \forall\, m \in \mathbf{Z}$, $m > 0$, запись g^{-m} обозначает элемент $(g^{-1})^m$ группы $\mathbf{Z}_n^*$.

Если в какое-либо выражение входит число, изначально выбранное как элемент множества $\mathbf{Z}_n$ или $\mathbf{Z}_n^*$, и это число является аргументом операций сложения, вычитания, умножения, или основанием в операции возведения в степень, то по умолчанию предполагается, что операции в этом выражении выполняются как операции в $\mathbf{Z}_n$.

2.2. Автоматы Мура

Автомат Мура (называемый ниже просто **автоматом**) – это набор

$$M = (X, S, Y, s^0, \delta, \lambda) \tag{6}$$

где X, S и Y – множества, элементы которых называются соответственно **входными сигналами**, **состояниями**, и **выходными сигналами**, $s^0 \in S$ – **начальное состояние**, $\delta : S \times X \to S$ – **функция перехода**, $\lambda : S \to Y$ – **функция выхода**.

Автомат является моделью дискретной динамической системы, работа которой заключается в изменении состояний под воздействием входных сигналов, поступающих на её вход, и выдаче в каждый момент времени $t = 0, 1, 2, \ldots$ некоторого выходного сигнала.

В начальный момент времени ($t = 0$) автомат находится в состоянии s^0. В каждый момент времени $t = 0, 1, 2, \ldots$ автомат получает входной сигнал $x(t) \in X$, выдаёт выходной сигнал $y(t) \stackrel{\text{def}}{=} \lambda(s(t)) \in Y$, и в следующий момент времени $(t+1)$ переходит в состояние $s(t+1) \stackrel{\text{def}}{=} \delta(s(t), x(t))$.

Автомат называется **автономным**, если множество его входных сигналов состоит из одного элемента. Если M – автономный автомат с множеством состояний S, то можно считать, что его функция перехода δ имеет вид $S \to S$.

Для каждого множества X и каждого $n \geq 0$ запись X^n обозначает множество последовательностей длины n с компонентами из X (если $n = 0$, то X^n состоит из одной пустой последовательности, обозначаемой символом ε). Запись X^* обозначает множество $\bigcup_{n \geq 0} X^n$.

Пусть M – автомат вида (6). $\forall\, (s, u) \in S \times X^*$ запись su обозначает состояние, в которое перейдет автомат M из состояния s после поступления на его вход последовательности входных сигналов u, т.е. $s\varepsilon \stackrel{\text{def}}{=} s$, $sx \stackrel{\text{def}}{=} \delta(s, x)$, $sx_1 \ldots x_{n-1}x_n \stackrel{\text{def}}{=} (sx_1 \ldots x_{n-1})x_n$.

Автомат (6) называется **автоматом без выхода**, если $Y = S$, и $\lambda = id_S$. Если (6) – автомат без выхода, то будем обозначать его четверкой (X, S, s^0, δ).

3. Криптографические примитивы

Криптографические примитивы – это математические понятия и конструкции, используемые в качестве элементарных компонентов при построении ПБ. Криптографические примитивы предназначены для обеспечения различных свойств безопасности ПБ. К криптографическим примитивам относятся системы шифрования, хэш-функции, и некоторые другие понятия и конструкции.

3.1. Системы шифрования

3.1.1. Симметричные системы шифрования

Симметричные СШ в основном относятся к следующим двум классам: блочные и поточные.

В **блочных СШ** шифруемое сообщение разбивается на блоки одинакового размера, и каждый блок шифруется при помощи одного и того же алгоритма.

Шифрующие преобразования блоков заключаются в суперпозиции нескольких простых отображений, называемых **базовыми преобразованиями**.

Среди базовых преобразований блоков наибольшее распространение получили **преобразования Фейстеля**, которые заключаются в разделении обрабатываемого блока на левую и правую половины L и R, и преобразовании блока (L, R) в блок (L', R') (где L' и R' – левая и правая половины преобразованного блока) по следующему принципу:

$$L' := R, \quad R' := L \oplus f(R, k),$$

где $\oplus$ – побитовое сложение по модулю 2, и k – ключ.

Алгоритм шифрования реализуется несколькими итерациями преобразования Фейстеля, при этом очередная итерация использует в качестве входного блока результат предыдущей итерации.

Для дешифрования применяется обратное преобразование, которое вычисляется по той же схеме, как и исходное:

$$L = R' \oplus f(L', k), \quad R = L'.$$

В **поточных СШ** шифрование заключается в сложении по модулю 2 каждого бита открытого текста с соответствующим битом псевдослучайной последовательности, называемой **гаммой**. Дешифрование осуществляется по той же схеме, как и шифрование.

Для порождения гаммы используются **регистры сдвига с линейной обратной связью (РСЛОС)**. РСЛОС – это автономный автомат вида $(\{1\}, \mathbf{Z}_2^n, \mathbf{Z}_2, k, \delta, \lambda)$, функция переходов которого сопоставляет произвольному состоянию $(q_1, \ldots, q_n)$ состояние $(q_2, \ \ldots, \ q_n, \ \sum_{i=1}^{n} c_i q_i)$, где $c_1, \ldots, c_n$ – фиксированные элементы $\mathbf{Z}_2$.

Гамма, которую порождает РСЛОС, представляет собой последовательность его выходных сигналов (которые он выдает в моменты 0, 1, и т.д.). Для шифрования или дешифрования при помощи РСЛОС его начальное состояние полагается равным ключу.

Существуют и другие способы порождения гаммы:

- первый заключается в использовании двух РСЛОС: если гаммы, порожденные ими, имеют вид $a_0, a_1, \ldots$ и $b_0, b_1, \ldots$ соответственно, то результирующая гамма получается из гаммы $a_0, a_1, \ldots$ удалением её компонентов с такими номерами i, что $b_i = 0$,

- другой способ заключается в том, что вместо РСЛОС используются автономные автоматы с состояниями из R^n (где R – конечное кольцо, элементы R^n рассматриваются как вектор-столбцы длины n над R), отображение переходов которых переводит состояние $q \in R^n$ в состояние $Aq+b$, где A – матрица порядка n над R, $b \in R^n$.

3.1.2. Асимметричные системы шифрования

В асимметричных СШ каждый агент a использует пару ключей a^+, a^-, где

- a^+ – **открытый ключ**, он используется для шифрования, и должен быть известен всем агентам,
- a^- – **закрытый ключ**, он используется для дешифрования, и должен быть известен только агенту a.

Одним из примеров ассиметричных СШ является **СШ RSA**. Её название является аббревиатурой, связанной с фамилиями её создателей (Rivest, Shamir и Adleman). Криптографическая стойкость данной СШ (т.е. сложность нахождения по ШТ $k(m)$ сообщения m без знания ключа дешифрования) основывается на вычислительной сложности задачи разложения на множители больших целых чисел.

Для задания конкретной реализации СШ RSA агент a должен сгенерировать два больших (несколько сотен битов) простых числа p и q, примерно одинаковых по размеру, и таких, что НОД $(p-1, q-1)$ – небольшое число.

Ключи a^+ и a^- имеют следующий вид:

- $a^+ = \{n, e\}$, где $n = pq$, $e \underset{r}{\in} \mathbf{Z}^*_{\varphi(n)}$
  (запись вида $x \underset{r}{\in} X$ означает, что x – случайно и равновероятно выбранный элемент множества X),
- $a^- = \{d, p, q\}$, где $ed \underset{\varphi(n)}{=} 1$.

Для шифрования шифруемое сообщение разбивается на блоки, каждый из которых можно рассматривать как двоичную запись числа из $\mathbf{Z}_n$. Каждый из этих блоков шифруется отдельно. Шифрование и дешифрование определяются следующим образом: (все вычисления – в $\mathbf{Z}_n$)

$$\forall\, m \in \mathbf{Z}_n \quad a^+(m) \stackrel{\text{def}}{=} m^e, \;\; a^-(m) \stackrel{\text{def}}{=} m^d.$$

Нетрудно доказать, что СШ RSA обладает свойством

$$\forall\, m \in \mathbf{Z}_n \;\; a^-(a^+(m)) = m, \quad a^+(a^-(m)) = m.$$

Другим примером асимметричной СШ является **СШ Эль-Гамаля**. Её криптографическая стойкость основывается на вычислительной сложности задачи дискретного логарифмирования, т.е. задачи нахождения по паре $a, b \in \mathbf{Z}_p$ (где p – простое число) такого $x \in \mathbf{Z}$, что $a^x \underset{p}{=} b$.

Для задания конкретной реализации СШ Эль-Гамаля агент a должен сгенерировать большое простое число p, и примитивный элемент $g \in \mathbf{Z}_p^*$. Ключи a^+ и a^- имеют следующий вид: $a^- = x \underset{r}{\in} \mathbf{Z}_p^*$, $a^+ = \{p, g, y\}$, где $y \underset{p}{=} g^x$.

Как и в RSA, для шифрования шифруемое сообщение разбивается на блоки, каждый из которых можно рассматривать как двоичную запись числа из $\mathbf{Z}_p$. Каждый из этих блоков шифруется отдельно. Шифрование и дешифрование определяются следующим образом:

$$a^+(m) = (g^z,\ my^z), \qquad \text{где } z \underset{r}{\in} \mathbf{Z}_p^*, \quad a^-(u, v) = u^{-x}v.$$

3.2. Хэш-функции

3.2.1. Понятие хэш-функции

Хэш-функция (ХФ) - это функция вида $h : D \to \mathbf{Z}_2^n$ (где $D \subseteq \mathbf{Z}_2^*$), удовлетворяющая условиям:

- h – **односторонняя функция**, т.е. не существует быстрого алгоритма нахождения по заданному $y \in \mathbf{Z}_2^n$ такого $x \in D$, что $y = h(x)$,
- h **устойчива к коллизиям**, т.е. сложно найти различные $x_1, x_2 \in D$, такие, что $h(x_1) = h(x_2)$.

Алгоритм вычисления ХФ должен быть общеизвестен. ХФ применяются для контроля целостности данных, аутентификации источников данных, и многих других целей.

Ниже символ h (возможно с индексами) обозначает ХФ.

3.2.2. Пример построения хэш–функции

Каждый автомат без выхода M вида $(\mathbf{Z}_2^n, \mathbf{Z}_2^n, s^0, \delta)$ определяет функцию h_M, которая вычисляется следующим образом: $\forall\, x \in \mathbf{Z}_2^* \;\; x$ дополняется

до размера, кратного n, представляется в виде конкатенации $x_1 \dots x_k$, где $\forall\, i = 1, \dots, k\ x_i \in \mathbf{Z}_2^n$, и $h_M(x) \stackrel{\text{def}}{=} s^0 x_1 \dots x_k$.

h_M может быть ХФ, если $\forall\, s, x \in \mathbf{Z}_2^n \quad \delta(s, x) = s(x) \oplus x$, где $s(x)$ – ШТ, получаемый шифрованием x на ключе s. Если же $\delta(s, x)$ имеет вид $k(s \oplus x)$, где k – фиксированный ключ, то h_M скорее всего не является ХФ.

3.2.3. Стандарт хэш–функции SHS

Стандарт ХФ SHS (Secure Hash Standard) был разработан в 1993 г., он основан на алгоритме MD4 Ривеста. Определяемая ниже ХФ h, построенная по данному стандарту, преобразует битовые строки длины $\leq 2^{64}$ в битовые строки длины 160. Значение $h(m)$ вычисляется следующим образом.

Пусть m состоит из n бит. Сначала к m справа приписывается битовая строка вида $10 \dots 0 l_1 \dots l_{64}$, такая, что $l_1 \dots l_{64}$ – двоичная запись числа n, и размер получившейся строки u делится на 512 ($= 32{\cdot}16$). Представим u в виде конкатенации блоков $u_1 \dots u_k$, где длина каждого блока u_i равна 512 (т.е. u_i можно рассматривать как число из $\mathbf{Z}_{2^{32}}^{16}$ $(i = 1, \dots, k)$). Искомое значение $h(m)$ равно состоянию $s^0 u_1 \dots u_k$ автомата без выхода $(\mathbf{Z}_{2^{32}}^{16}, \mathbf{Z}_{2^{32}}^{5}, s^0, \delta)$, где $\delta(s, x) = s + g(s, x)$, и

- s и $g(s, x)$ рассматриваются как последовательности из пяти блоков, которые $\in \mathbf{Z}_{2^{32}}$, сложение выполняется поблочно, блоки складываются в $\mathbf{Z}_{2^{32}}$, и

- $g(s, x)$ – состояние, в которое перейдёт автомат без выхода $(\mathbf{Z}_{2^{32}} \times \mathbf{Z}_{2^{32}} \times \mathbf{Z}, \mathbf{Z}_{2^{32}}^{5}, s, \delta')$ после поступления на его вход последовательности $(v_0, z_0, 0) \ \dots\ (v_{79}, z_{79}, 79)$, которая удовлетворяет условиям:

$$
\begin{aligned}
&(v_0, \dots, v_{15}) = x,\\
&v_i = v_{i-3} \oplus v_{i-8} \oplus v_{i-14} \oplus v_{i-16} \quad (i = 16, \dots, 79),\\
&z_{20i} = z_{20i+1} = \dots = z_{20i+19} \quad (i = 0, 1, 2, 3),
\end{aligned}
$$

и $\delta'((a, b, c, d, e), (v, z, i)) =$
$= (T^5(a) + f(b, c, d, i) + e + v + z, a, T^{30}(b), c, d)$, сложение – в $\mathbf{Z}_{2^{32}}$,

T – циклический сдвиг влево на 1 бит, и

$$f(b,c,d,i) = \begin{cases} (b \wedge c) \vee (\neg b \wedge d) & (i = 0, \ldots, 19) \\ b \oplus c \oplus d & (i = 20, \ldots, 39, 60, \ldots, 79) \\ (b \wedge c) \vee (b \wedge d) \vee (c \wedge d) & (i = 40, \ldots, 59) \end{cases}$$

($\neg$, $\wedge$, $\vee$, $\oplus$ – побитовые операции над блоками).

3.3. Цифровая подпись

Цифровая подпись является криптграфическим примитивом, предназначенным для доказательства подлинности передаваемых сообщений и их отправителей.

Алгоритм вычисления цифровой подписи преобразует пару (m, a), где m – подписываемое сообщение, и a – имя агента, подписывающего это сообщение, в строку $\langle m \rangle_a^s$, называемую **цифровой подписью (ЦП)** сообщения m, созданной агентом a. При вычислении $\langle m \rangle_a^s$ используется **закрытый ключ ЦП** агента a, который обозначается записью s_a, и должен быть известен только агенту a. Тройка $(m, a, \langle m \rangle_a^s)$ обозначается записью $\langle m \rangle_a$, и называется **подписанным сообщением**.

ЦП должна обладать следующими свойствами:

- **возможность проверки подлинности ЦП**: существует открытый алгоритм позволяющий по тройке (m, a, u) проверить истинность равенства $u = \langle m \rangle_a^s$,
- **невозможность подделки ЦП**: задача вычисления $\langle m \rangle_a^s$ без знания s_a является труднорешаемой,
- **невозможность подмены**: задача нахождения по $\langle m \rangle_a^s$ такого $m' \neq m$, что $\langle m' \rangle_a^s = \langle m \rangle_a^s$, является труднорешаемой.

Если подписывающий агент a и проверяющий агент b используют одну и ту же АШС, $\langle m \rangle_a^s$ может иметь, например, один из следующих видов:

$$a^-(h(m)), \quad b^+a^-(h(m)), \quad a^-(a, a^-(h(m)), t)$$

где h –ХФ, t – метка времени, $s_a = a^-$.

3.4. Схемы разделения секрета

Еще одним классом криптографических примитивов являются схемы разделения секрета.

3.4.1. Понятие схемы разделения секрета

Схема разделения секрета (СРС) – это способ распределения секретной информации между несколькими агентами, используя которую, они могут вычислить некоторое заданное значение s (называемое **секретом**). Информация, которую при этом получает каждый агент, называется **долей** этого агента. Например, секрет $s \in \mathbf{Z}_2^l$ распределяется между n агентами, доли которых имеют вид $s_1 \underset{r}{\in} \mathbf{Z}_2^l$, ..., $s_{n-1} \underset{r}{\in} \mathbf{Z}_2^l$, $s_n = s \oplus s_1 \oplus \ldots \oplus s_{n-1}$. Значение s могут вычислить только все агенты совместно, а если доля хотя бы одного из агентов неизвестна, то все остальные агенты, даже открыв друг другу свои доли, не могут извлечь из них никакой информации о значении s.

3.4.2. Схема разделения секрета Шамира

Излагаемая в этом пункте СРС Шамира относится к числу (n, k)-пороговых СРС (где $1 < k \leq n$), т.е. таких СРС, в которых секретное значение s распределяется между n агентами таким образом, что

- любая совокупность из k агентов может, используя свои доли, вычислить s, и
- любая совокупность из $< k$ агентов не сможет извлечь из своих долей никакой информации об s.

СРС Шамира используется в том случае, когда секретное значение s является элементом некоторого поля P.

Для распределения секрета s среди агентов $a_1, \ldots, a_n$ выбираются многочлен $f \in P[x]$ вида $s + c_1 x + \ldots + c_{k-1} x^{k-1}$ ($c_{k-1} \neq 0$) и различные элементы $x_1, \ldots, x_n \in P \setminus \{0\}$.

$\forall\, i = 1, \ldots, n$ доля агента a_i имеет вид пары $(x_i, f(x_i))$.

Для вычисления секрета используется интерполяционная теорема Лагранжа: для каждого k–элементного подмножества $\{y_1, \ldots, y_k\} \subseteq \{x_1, \ldots, x_n\}$ верно равенство

$$f(x) = \sum_{i=1}^{k} f(y_i) \prod_{j \neq i} \frac{x - y_j}{y_i - y_j}$$

из которого следует, что $s = f(0) = \sum\limits_{i=1}^{k} f(y_i) \prod\limits_{j \neq i} \frac{y_j}{y_j - y_i}$.

Если доли вычисляются и распределяются агентом s, которому агенты $a_1, \ldots, a_n$ не доверяют, и хотят проверить правильность своих долей, то в том случае, когда $P = \mathbf{Z}_p$, это можно сделать следующим образом. Выбирается несекретный элемент $g \in P$, и s посылает всем агентам множество $\{d_i := g^{c_i} \mid i = 0, \ldots, k-1\}$ $(c_0 = s)$. Агент a_i проверяет правильность своей доли z_i путем проверки равенства

$$g^{z_i} = d_0 d_1^{x_i} \ldots d_{k-1}^{\binom{x_i}{k-1}}.$$

3.4.3. Схема Карнина-Грини-Хеллмана

Другим примером (n, k)–пороговой схемы является схема Карнина-Грини-Хеллмана. В ней секрет s тоже является элементом поля P. Для вычисления долей выбираются различные числа $r_0, \ldots, r_n \in P$, и вычисляется множество строк $\vec{v}_0, \ldots, \vec{v}_n$, где $\forall\, i = 0, \ldots, n \quad \vec{v}_i = (1\, r_i\, r_i^2\, \ldots\, r_i^{k-1})$. Обозначим символом $u^{\downarrow}$ столбец из P^k с компонентами $u_0, \ldots, u_{k-1}$, где $u_1 \underset{r}{\in} P, \ldots, u_{k-1} \underset{r}{\in} P$, $u_0 = s - \sum_{i=1}^{k} r_0^i u_i$ (т.е. $s = \vec{v}_0 u^{\downarrow}$).

$\forall\, i = 1, \ldots, n$ доля агента a_i имеет вид $\vec{v}_i u^{\downarrow}$. Каждые k долей порождают невырожденную систему из k линейных уравнений, неизвестными в которой являются $u_0, \ldots, u_{k-1}$.

3.4.4. Схема разделения секрета с двумя группами агентов

Излагаемая ниже СРС предназначена для решения следующей задачи: имеется две группы агентов $\{a_1, \ldots, a_{n_1}\}$ и $\{b_1, \ldots, b_{n_2}\}$, и для вычисления секрета s требуются доли любых k_1 агентов из первой группы, и любых k_2 агентов из второй группы.

Если s можно представить в виде ненулевого элемента некоторого поля P, то для решения этой задачи можно использовать СРС, аналогичную СРС Шамира: в данном случае для распределения секрета s выбираются

- пара многочленов $f_1, f_2 \in P[x]$ степени $k_1 - 1$ и $k_2 - 1$ соответственно, причём $s = f_1(0) f_2(0)$, и
- пара подмножеств $\{x_1, \ldots, x_{n_1}\}, \{y_1, \ldots, y_{n_2}\}$ множества $P \setminus \{0\}$.

$\forall\, i = 1, \ldots, n_1$ доля агента a_i имеет вид пары $(x_i, f_1(x_i))$, и $\forall\, j = 1, \ldots, n_2$ доля агента b_j имеет вид пары $(y_j, f_2(y_j))$.

4. Протоколы аутентификации

4.1. Понятие протокола аутентификации

Важный класс протоколов безопасности составляют **протоколы аутентификации (ПА)**. Они предназначены для решения задачи **аутентификации** (т.е. доказательства подлинности) агентов, ключей, сообщений, времени создания сообщений, сеансов связи, и т.д.

Мы будем рассматривать в этом параграфе только ПА агентов. Проблема аутентификации агентов представляет большую актуальность, например, в том случае, когда агенты выражают желание получить доступ к ресурсам, безопасность которых представляет повышенный интерес (банковские счета, секретные базы данных, государственные здания, и т.д.). Как правило, в ПА агентов

- принимают участие два обычных агента, которых мы будем обозначать символами a и b (в литературе по ПБ их обычно называют Алиса и Боб),
- а также может принимать участие доверенный посредник I, в этом случае мы будем предполагать, что a, b и I могут использовать общую ССШ.

Как правило, аутентификация агента a перед другим агентом b заключается в том, что a доказывает b знание некоторого секретного значения s, причём во многих случаях a должен убедить b что он знает s, не раскрывая s. Если после завершения этого доказательства у b не появляется новой информации о том, в каком диапазоне может содержаться значение s, то такое доказательство называется **доказательством с нулевым разглашением (Д0Р)**.

ПА агентов включают в себя следующие два класса:

- протоколы **односторонней аутентификации (ПОА)**, в которых только один из агентов (a) доказывает свою подлинность другому агенту (b), и
- протоколы **двусторонней аутентификации (ПДА)**, в которых оба агента a и b доказывают свою подлинность друг другу.

Некоторые ПА предназначены для одновременного решения двух задач: аутентификации агентов, и выработки ими (или передачи им от I) нового сеансового ключа ССШ, который эти агенты могут использовать

для организации сеанса шифрованной связи друг с другом после завершения работы ПА.

В ПА часто используются **метки времени (МВ)**, они обозначаются символом t (возможно, с индексами), и символ t (возможно, с индексами) обозначает только МВ. Если какой-либо агент получает сообщение, содержащее МВ, то проводит дополнительную проверку того, что эта МВ принадлежит заданному промежутку $[t_{min}, t_{max}]$.

4.2. Простейшие протоколы аутентификации

4.2.1. Простейшие протоколы односторонней аутентификации

- $\{a \to b : k_{ab}(b, t)$, в этом протоколе $s = k_{ab}$.
- $\{a \to b : \langle b, t \rangle_a$, в этом протоколе $s = s_a$.

4.2.2. Протоколы односторонней аутентификации с использованием паролей

Пароль - это строка, являющаяся общим секретом a и b.

Примеры простейших ПОА с использованием паролей:

- $\{a \to b : a, \text{пароль}$.
- $\{a \to b : a, r, h(\text{пароль}, r)$, этот ПОА обеспечивает защиту от перехвата паролей.

Как правило, в системах аутентификации с использованием паролей выполняется регулярное обновление паролей, которое может происходить по одной из следующих схем.

1) Агенты a и b имеют общий список паролей, и заранее договариваются о порядке смены паролей.

2) Сначала a и b используют пароль p_0. Каждый сеанс аутентификации агентов a и b имеет свой порядковый номер $i = 0, 1, \ldots$. На сеансе аутентификации с номером i используется пароль p_i. Пароли $p_1, \ldots$ генерируются агентом a. Агенты a и b используют алгоритм, который по каждому паролю p_i вырабатывает ключ ССШ k_i. i–й сеанс ПОА имеет вид $a \to b : p_i, k_i(p_{i+1})$.

3) Агенты a и b выбирают число n, представляющее собой максимальное количество сеансов аутентификации, которые они собираются выполнить.

   a генерирует последовательность паролей $p_0, \ldots, p_n$, где $p_0 = r$, $p_i = h(p_{i-1})$ $(i = 1, \ldots, n)$.

   b каким-либо образом получает p_n. $\forall\, i = 1, \ldots, n$ в i–м сеансе аутентификации a посылает b пароль p_{n-i}, и b проверяет равенство $h(p_{n-i}) = p_{n-i+1}$.

4.2.3. Простейшие протоколы двусторонней аутентификации

1) $$\begin{cases} b \to a : r_b \\ a \to b : k_{ab}(b, r_a, r_b) \\ b \to a : k_{ab}(r_a, r_b) \end{cases}$$

2) $$\begin{cases} b \to a : r_b \\ a \to b : \langle b, r_a, r_b \rangle_a \\ b \to a : \langle a, r_a, r_b \rangle_b \end{cases}$$

3) $$\begin{cases} b \to a : r_b \\ a \to b : r_a, h(a, r_a, r_b) \\ b \to a : h(b, r_a) \end{cases}$$

4.3. Вопросно-ответные протоколы односторонней аутентификации

Работа **вопросно-ответного ПОА** состоит из нескольких раундов, в каждом из которых a посылает b доказательство знания секрета s, и b либо принимает это доказательство, либо не принимает. Как правило, в каждом раунде b посылает a некоторый вопрос, и доказательство знания s представляет собой ответ на этот вопрос.

a проходит аутентификацию только в том случае, если в каждом из раундов b принимает доказательство, которое ему прислал a. Если в одном раунде вероятность того, что b пример ответ a, не превосходит p, то вероятность того, что a правильно ответит в d раундах, не превосходит p^d (поскольку все раунды независимы).

Для защиты от атаки с повторной передачей (**replay**) в вопросы и ответы можно включать нонсы и МВ.

4.3.1. Вопросно-ответные однораундовые протоколы односторонней аутентификации

1) $\begin{cases} b \to a : r \\ a \to b : k_{ab}(b, r) \end{cases} \quad (s = k_{ab})$

2) $\begin{cases} b \to a : r \\ a \to b : \langle b, r \rangle_a \end{cases} \quad (s = s_a)$

3) $\begin{cases} b \to a : r \\ a \to b : a, a^-(r) \end{cases} \quad (s = a^-)$

4) $\begin{cases} b \to a : \ h(r), b, a^+(a, r) \\ a \to b : r \end{cases} \quad (s = a^-).$

4.3.2. Протокол аутентификации Шнорра

Открытые параметры:

- простые числа p и q, где $|p| \geq 512$, $|q| \geq 140$, $q \mid p - 1$,
- элемент $g \in \mathbf{Z}_p^*$, такой, что $\mathrm{ord}(g) = q$.

Секретным значением является число $s \underset{r}{\in} \mathbf{Z}_q^*$, для проверки знания s используется открытое значение $v = g^{-s}$.

ПОА Шнорра имеет следующий вид:

$$\begin{cases} a \to b : x := g^z, \text{ где } z \underset{r}{\in} \mathbf{Z}_q^* \\ b \to a : e \underset{r}{\in} \mathbf{Z}_q^*, \text{ где } |e| < 100 \\ a \to b : y := z + se \\ b \text{ принимает ответ, если } x = g^y v^e \end{cases}$$

Если a не знает s, но хочет послать то значение y, которое примет b, он должен уметь вычислять для каждого возможного e значение $y = z + se$, что равносильно знанию $s = (y - z)e^{-1}$.

Можно доказать, что данный ПА является Д0Р.

4.3.3. Протокол Шаума

Открытые параметры:

- целое число $n > 0$, и
- порождающий элемент g группы $\mathbf{Z}_n^*$.

Секретным значением является число $s \underset{r}{\in} \mathbf{Z}_{\varphi(n)}$, для проверки знания s используется открытое значение $v = g^s$.

Протокол состоит из d раундов, имеющих вид

$$\begin{cases} a \to b : x := g^z, \text{ где } z \underset{r}{\in} \mathbf{Z}_{\varphi(n)} \\ b \to a : e \underset{r}{\in} \{0,1\} \\ a \to b : y := z + se \\ b \text{ принимает } y, \text{ если } g^y = xv^e \end{cases}$$

Если a не знает s, но хочет послать такой ответ y, который примет b, то он может смошенничать следующим образом: выбрать $e' \underset{r}{\in} \{0,1\}$, и послать в качестве x значение $g^z v^{-e'}$, а в качестве y – значение z. В этом случае b примет ответ, если будет верно равенство $g^z = g^z v^{-e'} v^e$, т.е. если e' совпадает со значением e, которое ему пришлет b. Таким образом, вероятность успеха в одном раунде при описанном выше мошенничестве равна $1/2$.

Модификации протокола Шаума:

1) Открытые параметры те же, что и в протоколе Шаума. Секретным значением является кортеж $s = (s_1, \ldots, s_l) \underset{r}{\in} \mathbf{Z}_{\varphi(n)}^l$, для проверки знания s используется открытый кортеж $(v_1, \ldots, v_l)$, где $v_i = g^{s_i}$ $(i = 1, \ldots, l)$.

   Протокол состоит из одного раунда, имеющего вид

$$\begin{cases} a \to b : x := g^z, \text{ где } z \underset{r}{\in} \mathbf{Z}_{\varphi(n)} \\ b \to a : (e_1, \ldots, e_l) \underset{r}{\in} \{0,1\}^l \\ a \to b : y := z + s_1 e_1 + \ldots + s_l e_l \\ b \text{ принимает } y, \text{ если } g^y = x v_1^{e_1} \ldots v_l^{e_l} \end{cases}$$

2) Открытые параметры: целое число $n > 0$, и элементы $g_1, \ldots, g_l$ группы $\mathbf{Z}_n^*$, имеющие большой порядок. Секретное значение – кортеж $s = (s_1, \ldots, s_l) \underset{r}{\in} \mathbf{Z}_{\varphi(n)}^l$, для проверки знания s используется открытое значение $v = g_1^{s_1} \ldots g_l^{s_l}$.

Протокол состоит из d раундов, имеющих вид

$$\begin{cases} a \to b : (x_1, \ldots, x_l), \text{где } x_i := g_i^{z_i}, z_i \underset{r}{\in} \mathbf{Z}_{\varphi(n)} \ (i = 1, \ldots, l) \\ b \to a : e \underset{r}{\in} \{0, 1\} \\ a \to b : (y_1, \ldots, y_l), \text{ где } y_i := z_i + s_i e \ (i = 1, \ldots, l) \\ b \text{ принимает } y, \text{ если } g_1^{y_1} \ldots g_l^{y_l} = x_1 \ldots x_l v^e \end{cases}$$

3) Открытые параметры: те же, что и в предыдущем протоколе. Секретным значением является $s \underset{r}{\in} \mathbf{Z}_{\varphi(n)}$, для проверки знания s используется открытый кортеж $(v_1, \ldots, v_l)$, где $v_i = g_i^s \ (i = 1, \ldots, l)$.

Протокол состоит из d раундов, имеющих вид

$$\begin{cases} a \to b : (x_1, \ldots, x_l), \text{ где } x_i := g_i^z \ (i = 1, \ldots, l), z \underset{r}{\in} \mathbf{Z}_{\varphi(n)} \\ b \to a : e \underset{r}{\in} \{0, 1\} \\ a \to b : y := z + se \\ b \text{ принимает } y, \text{ если } \forall\, i = 1, \ldots, l \quad g_i^y = x_i v_i^e \end{cases}$$

4.3.4. Протоколы Фиата-Шамира и Гиллу-Кискате

ПА Фиата-Шамира имеет следующие закрытые параметры: простые числа p, q, где $|p|, |q| \geq 512$. Открытый параметр: $n := pq$. Секретным значением является число $s \underset{r}{\in} \mathbf{Z}_n^*$, для проверки знания s используется число $v := s^2$.

Протокол состоит из d раундов, имеющих вид

$$\begin{cases} a \to b : x := z^2, \text{ где } z \underset{r}{\in} \mathbf{Z}_n, \ z \neq 0 \\ b \to a : e \underset{r}{\in} \{0, 1\} \\ a \to b : y := zs^e \\ b \text{ принимает } y, \text{ если } y^2 = xv^e \end{cases}$$

Если a хочет пройти аутентификацию, не зная s, то для каждого значения $e \in \{0,1\}$, которое ему пришлёт b, a должен уметь вычислить значение y_e, которое он пошлет b в качестве y. Из условия принятия y следует, что $y_1^2 = y_0^2 v$. Нетрудно видеть, что возможность $\forall\, v \in \mathbf{Z}_n^*$ за полиномиальное время вычислить числа y_0, y_1 с описанным выше свойством равносильна возможности за полиномиальное время вычислить $\sqrt{v}$ в $\mathbf{Z}_n$. Однако можно доказать, что при описанных выше условиях на n задача вычисления $\sqrt{v}$ в $\mathbf{Z}_n$ без знания p и q является вычислительно сложной.

Если a не знает s, то для прохождения аутентификации он может попытаться смошенничать, аналогично тому как это делается в протоколе Шаума: a случайно выбирает число $e' \in \{0,1\}$, в качестве x посылает значение $z^2 v^{-e'}$, и в качестве y – значение z. b примет этот ответ, если будет верно равенство $z^2 = z^2 v^{-e'} v^e$, которое равносильно равенству $e' = e$, и которое будет верно с вероятностью $1/2$.

Можно доказать, что данный протокол является Д0Р.

Обобщением ПА Фиата-Шамира является излагаемый ниже ПА **Гиллу-Кискате** получаемый из ПА Фиата-Шамира путем замены возведения в квадрат на возведение в степень l, где $l \geq 2$ – открытое число. В нём для проверки знания секретного значения s используется число $v := s^{-l}$. Каждый раунд имеет вид

$$\begin{cases} a \to b : x := z^l, \text{ где } z \underset{r}{\in} \mathbf{Z}_n^*,\ z \neq 0 \\ b \to a : e \underset{r}{\in} \mathbf{Z}_l \\ a \to b : y := z s^e \\ b \text{ принимает } y, \text{ если } x = y^l v^e \end{cases}$$

Данный ПА тоже является Д0Р.

4.3.5. Протокол Фейге-Фиата-Шамира

Открытый параметр: $n := pq$, где p, q – простые числа, и $(p)_4 = (q)_4 = 3$ (числа n подобного вида называются **числами Блюма**), причем p, q секретны и $|p|, |q| \geq 512$.

Секретным значением s является кортеж $(s_1, \ldots, s_l)$, где $s_i \underset{r}{\in} \mathbf{Z}_n^*$ $(i = 1, \ldots, l)$. Для проверки знания s используется открытый кортеж $(v_1, \ldots, v_l)$, где $v_i = \pm s_i^{-2}$ $(i = 1, \ldots, l)$, причем числа $v_1, \ldots, v_l$ должны быть различны (запись вида $\pm u$ обозначает выражение, значение ко-

торого равно либо значению выражения u, либо значению выражения $-u$).

Протокол состоит из d раундов, имеющих вид

$$\begin{cases} a \to b : x := \pm z^2, \text{ где } z \underset{r}{\in} \mathbf{Z}_n \setminus \{0\} \\ b \to a : (e_1, \ldots, e_l) \underset{r}{\in} \{0,1\}^l \\ a \to b : y := z s_1^{e_1} \ldots s_l^{e_l} \\ b \text{ принимает } y, \text{ если } x = \pm y^2 v_1^{e_1} \ldots v_l^{e_l} \end{cases}$$

Если a не знает s, то для прохождения аутентификации он может попытаться смошенничать: a выбирает кортеж $e' = (e'_1, \ldots, e'_l) \underset{r}{\in} \{0,1\}^l$, в качестве x посылает значение $z^2 v_1^{e'_1} \ldots v_l^{e'_l}$, и в качестве y – значение z. b примет этот ответ, если $e' = e$, что может случиться с вероятностью 2^{-l}.

Можно доказать, что данный протокол является Д0Р.

Рекомендуемые значения для l и d: $l = 5$, $d = 4$.

4.4. Протоколы аутентификации с передачей нового сеансового ключа

В некоторых случаях после аутентификации агентов начинается новый сеанс связи между ними, с использованием для шифрования пересылаемых сообщений нового ключа. Как правило, в этих случаях передача ключа для нового сеанса связи совмещена с аутентификацией в одном протоколе. Этот ключ может быть создан как одним из агентов a, b, так и доверенным посредником I. В этом пункте мы изложим несколько протоколов аутентификации с передачей нового сеансового ключа, который мы будем обозначать символом k.

4.4.1. Односторонняя аутентификация с передачей сеансового ключа

1) Простейшие протоколы (k создается агентом a):

   а) $\{a \to b : b^+\langle b, k, t\rangle_a$

   б) $\{a \to b : \langle b, b^+(a, k), t\rangle_a$

   в) $\{a \to b : b^+(k, t), \langle b, k, t\rangle_a$

2) Вопросно-ответный протокол (k создается агентом a):

$$\begin{cases} b \to a : r \\ a \to b : k_{ab}(b, k, r) \qquad (\text{или } a \to b : k \oplus h(b, k_{ab}, r)) \end{cases}$$

3) Протокол Wide Mouth Frog (k создается агентом a):

$$\begin{cases} a \to I : a, k_{aI}(b, k, t_a) \\ I \to b : k_{bI}(a, k, t_b) \end{cases}$$

4) Протокол Otway–Rees (k создается агентом I):

$$\begin{cases} a \to b : a, b, k_{aI}(a, b, r, r_a), r \\ b \to I : a, b, k_{aI}(a, b, r, r_a), k_{bI}(a, b, r, r_b), r \\ I \to b : k_{aI}(k, r_a), k_{bI}(k, r_b), r \\ b \to a : k_{aI}(k, r_a), r \end{cases}$$

4.4.2. Двусторонняя аутентификация с передачей сеансового ключа

1) Простейшие протоколы (k создается агентом a):

а) $$\begin{cases} a \to b : r_a \\ b \to a : k_{ab}(r_a, r_b) \\ a \to b : k_{ab}(b, k, r_b) \qquad (\text{или } k \oplus h(b, k_{ab}, r_b)) \end{cases}$$

б) $$\begin{cases} a \to b : \langle b^+(k) \rangle_a, k(t_a) \\ b \to a : k(t_b) \end{cases}$$

2) Otway–Rees (k создается агентом I):

$$\begin{cases} a \to b : a, b, k_{aI}(a, b, r, r_a), r \\ b \to I : a, b, k_{aI}(a, b, r, r_a), k_{bI}(a, b, r, r_b), r \\ I \to b : k_{aI}(k, r_a), k_{bI}(k, r_b), r \\ b \to a : k_{aI}(k, r_a), k(r_a, r_b) \\ a \to b : k(r_b) \end{cases}$$

3) Yahalom (k создается агентом I):

$$\begin{cases} a \to b : a, r_a \\ b \to I : b, k_{bI}(a, r_a, r_b) \\ I \to a : k_{aI}(b, k, r_a, r_b), k_{bI}(a, k) \\ a \to b : k_{bI}(a, k), k(r_b) \end{cases}$$

4) Woo–Lam (k создается агентом I):

$$\begin{cases} a \to b : b^+(a, r_a) \\ b \to I : a, b, I^+(r_a) \\ I \to b : b^+\langle a, b, k, r_a\rangle_I \\ b \to a : a^+(\langle a, b, k, r_a\rangle_I, r_b) \\ a \to b : k(r_b) \end{cases}$$

5) Needham–Schroeder (k создается агентом I):

$$\begin{cases} a \to I : a, b, r_a \\ I \to a : k_{aI}(b, k, k_{bI}(k, a, t), r_a) \\ a \to b : k_{bI}(k, a, t) \\ b \to a : k(r_b) \\ a \to b : k(r_b - 1) \end{cases}$$

6) Neuman–Stubblebine (k создается агентом I):

$$\begin{cases} a \to b : \ a, \ r_a \\ b \to I : b, r_b, k_{bI}(a, r_a, t) \\ I \to a : k_{aI}(b, k, r_a, t), k_{bI}(k, a, t), r_b \\ a \to b : k_{bI}(k, a, t), k(r_b) \end{cases}$$

7) Kerberos:

- Прототип (k создается агентом I):

$$\begin{cases} a \to I: \ a, \ b, \ r \\ I \to a : k_{aI}(b, k, r, l), k_{bI}(a, k, l) \\ a \to b : k_{bI}(a, k, l), k(a, r', t) \\ b \to a : k(r', t) \end{cases}$$

где l = время действия ключа k.

- Основной протокол Kerberos предполагает работу с несколькими доверенными посредниками: I (authentication server) и $I_1, \ldots, I_n$ (tickets grant servers). k создается одним из агентов $I_1, \ldots, I_n$.

$$\begin{cases} a \to I : a, I_i, r \\ I \to a : k_{aI}(I_i, k_{aI_i}, r, l_1), k_{I_iI}(a, k_{aI_i}, l_1) \\ a \to I_i : k_{I_iI}(a, k_{aI_i}, l_1), k_{aI_i}(a, t_1), b, r' \\ I_i \to a : k_{aI_i}(b, r', k, l_2), k_{bI_i}(a, k, l_2) \\ a \to b : k_{bI_i}(a, k, l_2), k(a, r'', t_2) \\ b \to a : k(r'', t_2) \end{cases}$$

5. Алгоритмы вычисления цифровой подписи

Как было определено в пункте 3.3, алгоритм вычисления цифровой подписи преобразует пару (m, a), где m – подписываемое сообщение, и a – имя агента, подписывающего это сообщение, в строку $\langle m \rangle_a^s$, называемую цифровой подписью сообщения m, созданной агентом a. Ниже мы будем называть **цифровой подписью (ЦП)** не только строку $\langle m \rangle_a^s$, но и алгоритм вычисления этой строки.

В каждой из излагаемых ниже ЦП входными данными являются подписываемое сообщение m, а также дополнительные аргументы, называемые **параметрами**, каждый из которых может быть

- закрытым (т.е. известным только подписывающему агенту, один из таких параметров - закрытый ключ s_a подписывающего агента a), или

- открытым (т.е. известным всем, один из таких параметров – значение v_a, предназначенное для проверки подлинности ЦП, создаваемых агентом a).

Каждый из параметров по умолчанию считается открытым, а если он закрыт, то это специально оговаривается.

Проверка подлинности подписанного сообщения $\langle m\rangle_a$ заключается в вычислении значения некоторого булевозначного выражения e, аргументами которого являются это подписанное сообщение и открытые параметры ЦП. Как правило, выражение e (которое мы будем называть **проверкой ЦП**) имеет вид равенства. Подписанное сообщение считается подлинным, если e истинно.

Ниже используется следующее обозначение: если x – битовая строка, и n – целое положительное число, то запись $Pref(x, n)$ обозначает строку, состоящую из первых n битов строки x (если $|x| \geq n$), или строку, получаемую приписыванием к x справа $n - |x|$ нулей (если $|x| < n$).

5.1. Простейшие цифровые подписи

5.1.1. Цифровая подпись DSA

ЦП DSA (Digital Signature Algorithm) имеет следующие параметры: простые числа p и q, где $q \mid p-1$, $|p| \geq 512$ и делится на 64, $|q|$ – примерно 160, элемент $g \in \mathbf{Z}_p^*$ порядка q, ХФ h со значениями в $\mathbf{Z}_q^*$, $s_a \underset{r}{\in} \mathbf{Z}_q$, $v_a := g^{s_a}$.

$$\langle m\rangle_a^s := (s_1, s_2), \text{ где } \begin{cases} s_1 := (g^z)_q, \text{ где } z \underset{r}{\in} \mathbf{Z}_q^*, \\ s_2 := (h(m) + s_1 s_a) z^{-1}. \end{cases}$$

Проверка ЦП: $s_1 = (g^{h(m)s_2^{-1}} v_a^{s_1 s_2^{-1}})_q$.

5.1.2. Цифровая подпись ГОСТ

Параметры этой ЦП те же, что и у DSA, только $|q| = 256$.

$$\langle m\rangle_a^s := (s_1, s_2), \text{ где } \begin{cases} s_1 := (g^z)_q, \quad \text{где } z \underset{r}{\in} \mathbf{Z}_q^*, \\ s_2 := h(m) z + s_1 s_a. \end{cases}$$

Проверка ЦП: $s_1 = (g^{s_2 h(m)^{-1}} v_a^{-s_1 h(m)^{-1}})_q$.

5.1.3. Цифровая подпись Эль-Гамаля

Параметры: открытое простое число p, примитивный элемент $g \in \mathbf{Z}_p^*$, ХФ h со значениями в $\mathbf{Z}_{p-1}$, $s_a \underset{r}{\in} \mathbf{Z}_{p-1}$, $v_a := g^{s_a}$.

$\langle m\rangle_a^s := (s_1, s_2)$, где $\begin{cases} s_1 := g^z, \quad z \underset{r}{\in} \mathbf{Z}_{p-1}^* \\ s_2 := (h(m) - s_a s_1) z^{-1}. \end{cases}$

Проверка ЦП: $v_a^{s_1} s_1^{s_2} = g^{h(m)}$.

5.1.4. Слепая цифровая подпись

В некоторых случаях требуется, чтобы агент a, не получая никакой информации о сообщении m, создал бы такое значение, из которого можно извлечь $\langle m\rangle_a^s$. ЦП, получаемую при таких условиях, называют **слепой** ЦП.

Один из протоколов слепой ЦП имеет следующий вид. Параметры этой ЦП: секретные большие простые числа p и q, открытое число $n \overset{\text{def}}{=} pq$, ХФ h со значениями в $\mathbf{Z}_n$, $v_a \underset{r}{\in} \mathbf{Z}_{\varphi(n)}^*$, $s_a \in \mathbf{Z}_{\varphi(n)}^*$ удовлетворяет равенству $s_a v_a = 1$. Получение агентом b от агента a слепой ЦП сообщения m происходит по следующему протоколу:

$$\begin{cases} b \to a : u := h(m) x^{v_a}, \text{ где } x \underset{r}{\in} \mathbf{Z}_n^* \\ a \to b : v := u^{s_a} \\ b \text{ вычисляет искомое значение } \langle m\rangle_a^s := v x^{-1} \end{cases}$$

Нетрудно видеть, что

$$\langle m\rangle_a^s = v x^{-1} = (u^{s_a}) x^{-1} = ((h(m) x^{v_a})^{s_a}) x^{-1} =$$
$$= h(m)^{s_a} x^{v_a s_a} x^{-1} = h(m)^{s_a} x x^{-1} = h(m)^{s_a}.$$

Проверка подлинности: утверждение $y = \langle m\rangle_a^s$ считается верным, если $y^{v_a} = h(m)$.

5.2. Цифровая подпись, получаемая из протоколов аутентификации

5.2.1. Цифровая подпись Шнорра

ЦП Шнорра получается из ПА Шнорра, изложенного в пункте 4.3.2, в ней используются следующие параметры:

- простые числа p и q, где $|p| \geq 512$, $|q| \geq 140$, $q \mid p - 1$,
- элемент $g \in \mathbf{Z}_p^*$, такой, что $\operatorname{ord}(g) = q$,

- ХФ h со значениями в $\mathbf{Z}_q$,
- закрытый и открытый ключи: $s_a \underset{r}{\in} \mathbf{Z}_q^*$, $v_a := g^{-s_a}$.

$\langle m \rangle_a^s := (e, y)$, где $e := h(m, g^z)$, $y := z + s_a e$, $z \underset{r}{\in} \mathbf{Z}_q^*$.

Проверка ЦП (e, y): $h(m, v_a^e g^y) = e$.

5.2.2. Цифровая подпись Фиата-Шамира

ЦП Фиата-Шамира получается из ПА Фиата-Шамира, изложенного в пункте 4.3.4, заменой последовательности случайных битов $e_1, \ldots, e_d$, которые b посылает a в каждом из d раундов после получения x_i от a $(i = 1, \ldots, d)$, на префикс длины d строки $h(m, x_1, \ldots, x_d)$.

Параметры: n – открытое число вида pq, где p, q – секретные простые числа, $|p|, |q| \geq 512$, $s_a \underset{r}{\in} \mathbf{Z}_n^*$, $v_a := s_a^2$.

$\langle m \rangle_a^s := (e, y)$, где

- $e = Pref(h(m, z_1^2, \ldots, z_d^2), d)$, где $(z_1, \ldots, z_d) \underset{r}{\in} (\mathbf{Z}_n \setminus \{0\})^d$
- $y = (z_1 s_a^{e_1}, \ldots, z_d s_a^{e_d})$, где $(e_1, \ldots, e_d) = e$.

Проверка ЦП: $Pref(h(m, y_1^2 v_a^{-e_1}, \ldots, y_d^2 v_a^{-e_d}), d) = e$.

5.2.3. Цифровая подпись Гиллу-Кискате

Эта ЦП получается из ПА Гиллу-Кискате, изложенного в пункте 4.3.4, заменой случайного значения $e \underset{r}{\in} \mathbf{Z}_l$, которое генерирует b после получения $x := z^l$ от a, на $h(m, x)$, где h – ХФ с множеством значений $\mathbf{Z}_l$.

Параметры: n – открытое число вида pq, где p, q – секретные простые числа, $|p|, |q| \geq 512$, $l \geq 2$, $s_a \underset{r}{\in} \mathbf{Z}_n^*$, $v_a := s^{-l}$.

$\langle m \rangle_a^s := (e, y)$, где $e := h(m, z^l)$, $z \underset{r}{\in} \mathbf{Z}_n^* \setminus \{0\}$, $y := z s_a^e$.

Проверка подлинности: $e = h(m, y^l v_a^e)$.

5.2.4. Цифровая подпись Фейге-Фиата-Шамира

Эта ЦП получается из ПА Фейге-Фиата-Шамира, её параметры: открытое число $n := pq$, где p, q – секретные простые числа Блюма, $|p|, |q| \geq 512$, открытые числа l и d (рекомендуемые значения: $l = 9, d = 8$), $s_a = (s_1, \ldots, s_l) \underset{r}{\in} (\mathbf{Z}_n^*)^l$, $v_a = (v_1, \ldots, v_l)$, где $v_i := s_i^{-2}$ $(i = 1, \ldots, l)$, причем числа $v_1, \ldots, v_l$ различны.

$\langle m \rangle_a^s := (e, y)$, где

- $e = Pref(h(m, z_1^2, \ldots, z_d^2), dl)$, где $(z_1, \ldots, z_d) \underset{r}{\in} (\mathbf{Z}_n \setminus \{0\})^d$, обозначим строку e записью $(e_{11}, \ldots, e_{1l}, \ldots, e_{d1}, \ldots, e_{dl})$,

- $y = (y_1, \ldots, y_d)$, где $\forall\, i = 1, \ldots, d \quad y_i := z_i s_1^{e_{i1}} \ldots s_l^{e_{il}}$.

Проверка ЦП: $e = Pref(h(m, u_1, \ldots, u_d), d)$, где $\forall\, i = 1, \ldots, d \quad u_i := y_i^2 v_1^{e_{i1}} \ldots v_l^{e_{il}}$.

5.3. Стираемая цифровая подпись

5.3.1. Понятие стираемой цифровой подписи

Иногда требуется создать такую ЦП $\langle m \rangle_a^s$, чтобы

- подлинность этой ЦП могла быть доказана (или опровергнута) только с участием **уполномоченных агентов** (**designated confimers**), и

- если какой-либо уполномоченный агент доказал (или опроверг) подлинность этой ЦП какому-либо неуполномоченному агенту b, то b не мог бы использовать запись этого доказательства (или опровержения) для того, чтобы доказать (или опровергнуть) подлинность $\langle m \rangle_a^s$ другим агентам.

ЦП такого вида называется **стираемыми** (**undeniable**).

Доказательство или опровержение подлинности определяемых в этом параграфе стираемых ЦП производится при помощи интерактивных протоколов, где под **интерактивным протоколом (ИП)** доказательства справедливости какого-либо утверждения A понимается протокол, удовлетворящий следующим условиям:

- если A верно, то это будет установлено в результате работы этого протокола с большой вероятностью, и

- если A неверно, то установить в результате работы этого протокола обратное (т.е. то, что A верно) можно с небольшой вероятностью.

В пунктах 5.3.2 и 5.3.3 рассматриваются случаи, когда уполномоченным является только a, а в 5.3.4 – случай когда уполномоченными являются a и ещё один агент c.

5.3.2. Простейшая стираемая цифровая подпись

Излагаемая в этом пункте стираемая ЦП принадлежит Шауму и Антверпену. Её параметры: простые числа p и q, где $q \mid p-1$, элемент $g \in \mathbf{Z}_p^*$ порядка q, $s_a \underset{r}{\in} \mathbf{Z}_q^*$, $v_a := g^{s_a}$.

Будем предполагать, что подписываемое сообщение m является элементом $\mathbf{Z}_p^*$ (если это не выполняется, то заменим m на $h(m)$, где h – ХФ со значениями в $\mathbf{Z}_p^*$).

$\langle m \rangle_a^s := m^{s_a}$.

Протокол подтверждения подлинности $\langle m \rangle_a^s$ (т.е. доказательства утверждения $z = m^{s_a}$):

$$\begin{cases} b \to a : y := z^u v_a^v, \text{ где } u, v \underset{r}{\in} \mathbf{Z}_q \\ a \to b : w := y^{(s_a^{-1})} \\ b \text{ принимает ЦП, если } w = m^u g^v \end{cases}$$

(отметим, что этот протокол не обеспечивает нулевого разглашения s_a).

Стираемость этой ЦП обосновывается тем, что b может самостоятельно вычислить значение $m^u g^v$ и предъявить его в качестве того значения w, которое ему якобы переслал a.

5.3.3. Стираемая цифровая подпись с протоколом опровержения

Излагаемая в этом пункте стираемая ЦП с протоколом опровержения принадлежит Шауму. Её параметры: простое число p, примитивный элемент $g \in \mathbf{Z}_p^*$, $s_a \underset{r}{\in} \mathbf{Z}_p^*$, $v_a := g^{s_a}$.

Будем предполагать, что подписываемое сообщение m является элементом $\mathbf{Z}_p^*$ (если это не выполняется, то заменяем m на $h(m)$, где h – ХФ со значениями в $\mathbf{Z}_p^*$).

$\langle m \rangle_a^s := m^{s_a}$.

1) Протокол подтверждения подлинности $\langle m \rangle_a^s$ (т.е. доказательство утверждения $z = m^{s_a}$, где $z \in \mathbf{Z}_p^*$):

$$\begin{cases} b \to a : y := m^u g^v, \text{ где } u, v \underset{r}{\in} \mathbf{Z}_{p-1} \\ a \to b : (w_1, w_2), \text{ где } w_1 := yg^w, w_2 := w_1^{s_a}, \ w \underset{r}{\in} \mathbf{Z}_{p-1} \\ b \to a : (u, v) \\ a \to b : [\![y = m^u g^v]\!] \ w \\ b \text{ принимает ЦП, если } w_1 = yg^w \text{ и } w_2 = z^u v_a^{v+w} \end{cases}$$

Отметим, что данный протокол не обеспечивает нулевого разглашения s_a.

Стираемость этой ЦП обосновывается тем, что b может сам выбрать $w \underset{r}{\in} \mathbf{Z}_{p-1}$, и предъявить $(yg^w, z^u v_a^{v+w})$ как пару (w_1, w_2), которую ему якобы переслал a.

2) Протокол опровержения подлинности $\langle m \rangle_a^s$ (т.е. доказательство утверждения $z \neq m^{s_a}$, где $z \in \mathbf{Z}_p^*$): выбирается небольшое число $k \geq 2$, и

$$\begin{cases} b \to a : & (y_1 := m^u g^v, \ y_2 := z^u v_a^v), \text{ где } u \underset{r}{\in} \mathbf{Z}_k, v \underset{r}{\in} \mathbf{Z}_{p-1} \\ a \to b : & ru', \text{ где } r \underset{r}{\in} \mathbf{Z}_p^*, \text{ и } u' \in \mathbf{Z}_k \text{ – решение уравнения} \\ & \frac{y_1^{s_a}}{y_2} = (\frac{m^{s_a}}{z})^{u'}, \text{ которое ищется перебором} \\ & (u \text{ – одно из решений этого уравнения}) \\ b \to a : & v \\ a \to b : & [\![(y_1 = m^{u'} g^v) \wedge (y_2 = z^{u'} v_a^v)]\!] \ r \\ b : & \text{принимает опровержение, если } u' = u \end{cases}$$

Для обоснования корректности данного протокола заметим, что если $z = m^{s_a}$, то $\forall\, u' \in \mathbf{Z}_k \ \ u'$ является решением уравнения в описанном выше протоколе, поэтому вероятность того что $u' = u$, равна $1/k$.

5.3.4. Стираемая цифровая подпись, подтверждаемая двумя уполномоченными агентами

В этом пункте мы рассмотрим пример такой ЦП, в которой уполномоченными агентами являются подписывающий агент a и ещё один агент c. Её параметры: большое простое число p, элемент $g \in \mathbf{Z}_p^*$ порядка $p-1$, $s_a = (p_1, p_2)$, где p_1, p_2 – большие простые числа, $s_c \underset{r}{\in} \mathbf{Z}_{p-1}$, $v_a = (p, g, \eta, n)$, где $\eta := g^{s_c}$, $n := p_1 p_2$, ХФ h со значениями в $\mathbf{Z}_n$.

$\langle m \rangle_a^s := (s_1, s_2, s_3)$, где

$$s_1 = g^y \ \ (y \underset{r}{\in} \mathbf{Z}_{p-1}), \quad s_2 = \eta^y, \quad s_3 = \sqrt[3]{h(m) \oplus h(s_1, s_2)},$$

где $\sqrt[3]{\ }$ – функция вида $\mathbf{Z}_n \to \mathbf{Z}_n$ (т.к. a знает разложение n на простые множители, то он может вычислять её быстро).

Доказательство подлинности ЦП $\langle m \rangle_a^s$ агентом a:

$$\begin{cases} b \to a : z := g^u \eta^v, \text{ где } u, v \underset{r}{\in} \mathbf{Z}_{p-1} \\ a \to b : (d := g^w, e := (zd)^y), \text{ где } w \underset{r}{\in} \mathbf{Z}_{p-1} \\ b \to a : (u, v) \\ a \to b : [\![g^u \eta^v = z]\!] \ w \\ b \text{ принимает ЦП, если } g^w = d,\ e = s_1^{u+w} {s_2}^v, \text{ и} \\ \qquad h(m) \oplus h(s_1, s_2) = {s_3}^3 \end{cases}$$

Доказательство подлинности ЦП $\langle m \rangle_a^s$ агентом c получается из предыдущего протокола заменой a на c, y на s_c, η на s_1, и s_1 на η.

На базе этой ЦП и схемы разделения секрета можно построить такую ЦП, в которой доказывать подлинность ЦП a могут любые m агентов из заданной совокупности.

5.4. Совместная цифровая подпись

5.4.1. Понятие совместной цифровой подписи

В некоторых случаях требуется, чтобы сообщение m было одновременно подписано несколькими агентами $a_1, \ldots, a_k$. Соответствующая ЦП называется **совместной ЦП**, обозначается записью $\langle m \rangle_{a_1 \ldots a_k}^s$.

5.4.2. Примеры совместных цифровых подписей

Совместная ЦП может иметь например следующий вид.

1) $\langle m\rangle^s_{a_1\dots a_k} := (\langle m\rangle^s_{a_1}, \dots, \langle m\rangle^s_{a_k})$.

2) $\langle m\rangle^s_{a_1\dots a_k} := a_k^-(\dots(a_1^-(h(m)))\dots)$ (в данном случае агенты a_1, …, a_k используют общую АСШ).

3) Ещё одна совместная ЦП имеет следующие параметры: открытые натуральные числа n и u, ХФ h со значениями в $\mathbf{Z}_u$, $\forall\, i = 1, \dots, k \quad s_{a_i} \underset{r}{\in} \mathbf{Z}_n^*,\ v_{a_i} := s_{a_i}^{-u}$.

   $\langle m\rangle^s_{a_1\dots a_k} := (c, d)$, где $c := h(m, r_1^u \dots r_k^u)$,
   $d := r_1 s_{a_1}^c \dots r_k s_{a_k}^c$, $\forall\, i = 1, \dots, k \quad r_i \underset{r}{\in} \mathbf{Z}_n \setminus \{0\}$.

   Проверка подлинности: $c = h(m, d^u v_{a_1}^c \dots v_{a_k}^c)$.

5.4.3. Совместная цифровая подпись Брикелла-Ли-Якоби

Эта ЦП имеет следующие параметры: простое число p, элемент $g \in \mathbf{Z}_p^*$ порядка $p-1$, большое натуральное число l, ХФ h, $\forall\, i = 1, \dots, k \ \ s_{a_i} = \{s_{i1}, \dots, s_{il}\},\ \ v_{a_i} = \{v_{i1}, \dots, v_{il}\}$, где $\forall\, j = 1, \dots, l \ \ s_{ij} \underset{r}{\in} \mathbf{Z}_{p-1}$, $v_{ij} := g^{-s_{ij}}$.

В вычислении ЦП $\langle m\rangle^s_{a_1\dots a_k}$ принимает участие доверенный посредник I, с которым агенты $a_1, \dots, a_k$ могут обмениваться сообщениями с использованием общей АСШ. Протокол вычисления $\langle m\rangle^s_{a_1\dots a_k}$ имеет следующий вид:

$$\begin{cases} \forall\, i = 1, \dots, k \quad a_i \to I : I^+(y_i), \text{ где } y_i := g^{x_i}, x_i \underset{r}{\in} \mathbf{Z}_{p-1} \\ I \to \{a_1, \dots, a_k\} : y := y_1 \dots y_k \\ a_i \to I : z_i := x_i + \sum\limits_{j\in\{1,\dots,l\},\ b_j=1} s_{ij}, \quad b_j = \ j\text{-й бит } g^{h(m,y,a)} \\ \qquad\qquad \text{где } a \text{ – конкатенация имён агентов } a_1, \dots, a_k \\ I \text{ вычисляет } \langle m\rangle^s_{a_1\dots a_k} = z := \sum_{i=1}^k z_i \end{cases}$$

Проверка подлинности: $g^z \prod_{i=1}^k \prod\limits_{j\in\{1,\dots,l\},\ b_j=1} v_{ij} = y$.

Можно доказать, что данный протокол обеспечивает нулевое разглашение $s_{a_1}, \dots, s_{a_k}$, и если совместная ЦП не может быть создана по причине того, что некоторые агенты отказались вносить свой вклад в её

создание, в то время как другие агенты свой вклад уже внесли, то по результату работы агентов, внесших свой вклад в создание общей ЦП, невозможно идентифицировать этих агентов.

Список литературы

[1] Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. - М.: Триумф, 2002. - 816 с.

[2] Черемушкин А.В. Криптографические протоколы. Основные свойства и уязвимости. - М.: Академия, 2009. - 269 с.

[3] Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии. - М.: Гелиос АРВ, 2002. - 480 с.

[4] Запечников С.В. Криптографические протоколы и их применение в финансовой и коммерческой деятельности. - М.: Горячая Линия - Телеком, 2007. - 320 с.

[5] Анохин М.И., Варновский Н.П., Сидельников В.М., Ященко В.В. Криптография в банковском деле. - М.: МИФИ, 1997.

Security Protocols, Part 1

Mironov A.M.

The main cryptographic primitives used in security protocols are described (symmetric and asymmetric encryption systems, hash functions, secret sharing schemes), authentication protocols, and digital signature algorithms.

Keywords: protocols, security, cryptography, hash functions, authentication, digital signature

Часть 3.
Математические модели

О методах построения LDPC-кодов с заданными характеристиками.

Ананьев К.Ю.

В работе представлены алгоритмы построения проверочных матриц для LDPC - кодов на основе графа Таннера с обхватом 8. Также, в качестве параметров графа выступают разбиение степеней символьных вершин: отношение вершин степени 3 и степени 4 к общему числу символьных вершин, и скорость полученного кода. Код строится для произвольной скорости и произвольного разбиения за линейное, относительно колличество элементов матрицы, время.

Ключевые слова: LDPC - коды, граф Таннера, двудольные графы, распределение степеней вершин.

При передаче, информации разбивается на блоки определенной длины. Блоки преобразуются кодером или кодируются. Полученные блоки, которые называются кодовыми словами, передаются по каналу, возможно с ошибками. На обратной стороне декодер преобразовывает кодовые слова в исходную последовательность, исправляя, насколько возможно, ошибки.

Обобщение LDPC-кодов - коды на графах, были предложены Таннером. Проверочной матрице можно однозначно сопоставить двудольный граф следующим образом: пусть проверочная матрица имеет размер $(l \times m)$. Сопоставим ей граф $G = (V, W)$, причем $V = V_c \sqcup V_s$. $V_c = \{v_0^c, \dots v_{l-1}^c\}$ - множество проверочных вершин, $V_s = \{v_0^s, \dots v_{m-1}^s\}$ - множество символьных вершин. Тогда W множество ребер типа $W \subseteq V_c \times V_s$. Причем ребро $(v_i^c, v_j^s) \in W$, если на пересечении соответствующих строк и столбцов в проверочной матрице стоит 1. Число ребер, связывающих данный символьный/проверочный узел с проверочными/символьными узлами называется степенью этого узла.

Одной из характеристик кода является «скорость». Величина скорости показывает степень «избыточности» кода: чем больше скорость, тем эффективнее алгоритм кодирования.

Также важной характеристикой матрицы LDPC кода является отсутствие «циклов» определенной длины. Под «циклами» матрицы понимаются циклы в соответствующем Графе Таннера. Практика показала, что наличие циклов малой длины существенно усложняет процесс декодирования и увеличивает вероятность ошибки.

В данной работе приведен пример построения семейства LDPC-кодов для произвольной скорости без циклов длины 4 и 6 для некоторого распределения степеней символьных узлов за линейное относительно размеров матрицы время.

1. Постановка задачи и формулировка основных результатов

Пусть $\mathbf{G}$ некоторое семейство проверочных матриц (графов), которые будут получаться при реализации алгоритма. Тогда, опишем параметры графы, на которые мы будем обращать внимание при построении $\mathbf{G}$.

Скорость кода определяется соотношением $v = \frac{k}{m}$, где k - длина кодируемых данных, m - длина кодовых данных. Тогда проверочная матрица H имеет размер $((n-k) \times m) = (l \times m)$.

Для любой скорости v должна существовать проверочная матрица $g \in \mathbf{G}$ со скоростью $v_1 \geq v$.

Будем строить семейство матриц, с обхватом 8. Для любой матрицы $g \in \mathbf{G}$ обхват соответствующего графа Таннера больше 6.

Теперь опишем требования для степеней вершин. Среди проверочных вершин не должно быть «висячих» вершин: степень всех проверочных вершин больше единицы. Степени всех символьных вершин должны быть больше 3-х. Семейство $\mathbf{G}$ будет допускать построение графов, у которых степени символьных вершин 3 или 4. Причем для любой доли символьных вершин со степенью четыре - p_4 будет существовать граф $g \in \mathbf{G}$ такой, что доля вершин степени четыре в нем p_4^g в графе g приблизительно равна p_4.

В ходе работы будут построены два семейства графов $\mathbf{G}_1$ и $\mathbf{G}_2$. Разделение $\mathbf{G}_1$ и $\mathbf{G}_2$ обусловлена тем, что длина кода в $\mathbf{G}_1$ существенно ниже.

Первое семейство $\mathbf{G}_1$. Будет доказано, что обхват всех матриц из $\mathbf{G}_1$ будет больше 6-и, а все символьные вершины имеют степень 3 и будет справедлива следующая теорема:

Теорема 1. *Для любой скорости кода v существует такая $g \in \boldsymbol{G}_1$, у которой все символьные вершины имеют степень 3, и скорость полученного кода $v_1 > v$.*

Опишем $\mathbf{G}_2$. Помимо свойств $\mathbf{G}_1$, $\mathbf{G}_2$ будет допускать построение матриц, с заданным распределением символьных вершин степеней 3 и 4. Поэтому для $\mathbf{G}_2$ будет доказана следующая теорема.

Теорема 2. *Для любой скорости кода v и доли символьных вершин степени p_4 существуют матрица $g \in \boldsymbol{G}_2$, у которой все символьные вершины имеют степень 3 или 4, причем доля вершин степени 4 приблизительно равна p_4, а скорость полученного кода $v_1 > v$.*

2. Построение графа для произвольной скорости кода со степенью символьных узлов равной 2-м

Строение графа будет зависеть он нескольких параметров. Пусть $n > 1$ - произвольное натуральное число.

Выберем одну проверочную вершину и назовем ее «корневой». Из нее выходят ребра в n символьных узлов, а каждую символьную с новой проверочной. Пронумеруем последние проверочные вершины числами $1 \dots n$. Из каждой проверочной вершины крайнего слоя проведем i ребер в новые символьные вершины, где i номер вершины. Таким образом на последнем слое у нас $\frac{n^2+n}{2}$ вершин. Каждую из символьных вершин, выходивших из вершины под номером n соединяем с новыми n различными вершинами. Выбираем из них $n-1$ и соединяем со всеми вершинами выходящими из вершины с номером $n-1$. Затем выбираем из оставшихся $n-2$ вершины и т.д. Затем соединяем n вершин из последнего рассматриваемого слоя с новыми n символьными, а те в свою очередь сводим в одну проверочную. На рисунке 1 приведен граф при $n = 3$

Лемма 1. *Обхват получившегося графа равен 8.*

Пронумерует слои графа числами от 1 до 7. Будем рассматривать только простые циклы в графе.

Если цикл проходит через вершину первого(седьмого) слоя, то в силу построения он проходят через вершину пятого(третьего) слоя, и в цикле будет минимум 8 вершин. В этом случае цикл может проходит ровно через одну вершину в пятом(третьем) слое.

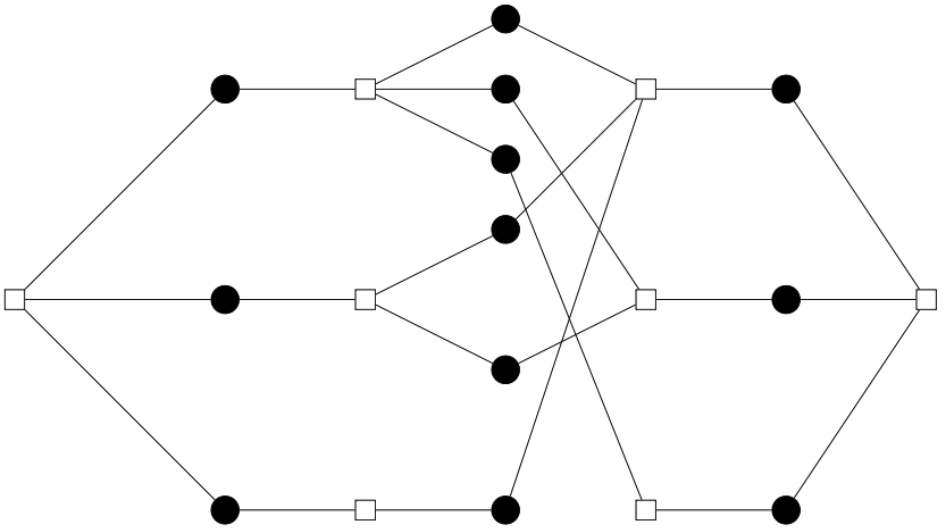


Рис.1

Если цикл проходит через вершину второго (шестого) слоя, то он проходит и через первый (седьмой) слой. Следовательно, переходим к предыдущему случаю.

Если цикл не проходит через вершины первых двух слоев, то в нем присутствует, как минимум, две вершины из третьего слоя v_1 и v_2. Т.к. вершин второго слоя в цикле нет, то, в силу построения, в цикле будут, минимум, четыре вершины из четвертого слоя: две из которых соединены с v_1 и две с v_2. В силу построения, вершины соединенные с v_i в четвертом слое соединяются с разными вершинами в пятом слое. Следовательно, в цикле есть минимум две вершины из пятого слоя. Таким образом, длина цикла не меньше 8.

Следовательно, обхват графа равен 8. Лемма доказана.

Скорость кода, соответственно, у таких графов равна:

$$v = 1 - \frac{2(1+n)}{2n + \frac{n^2+n}{2}} = 1 - \frac{4(1+n)}{n^2+5n}$$

и можно доказать следующую лемму:

Лемма 2. *Для любой скорости v_1 существует код построенный на графах данного типа со скоростью $v > v_1$.*

Несмотря на хорошие оценки скорости в этом графе есть существенный недостаток: наличие символьных вершин степени два может вы-

звать ошибки при декодировании. Для этого необходимо модифицировать исходные графы, чтобы степени символьных вершин были не меньше трех.

3. Первая модификация: построение семейства графов G_1 со степенями символьных вершин равными 3-м

Возьмем граф состоящий из первых пяти слоев графа, полученного в первой части работы, и назовем его «базисным». Сделаем $2 \cdot q$ копий базисного графа, пронумеруем их и начнем соединять следующим образом: вершины второго слоя первого графа соединяем с вершинами пятого слоя следующего и т.д. Это можно сделать, т.к. во втором и пятом слоях ровно по n вершин. Вершины последнего q-ого графа второго слоя с вершинами пятого слоя первого графа. Вершины второго слоя - символьные, а вершины пятого слоя проверочные, поэтому получившийся граф также двудольный.

Теперь добавим два дополнительного набора по $\frac{n^2+n}{2}$ проверочных вершин. Вершины из первого набора соединим с вершинами из 4 слоя базисных графов с четными номерами, а вершины второго набора соответственно с нечетными. Это можно сделать, т.к. в четвертом слое базисного графа также ровно $\frac{n^2+n}{2}$ вершин. Для того, чтобы не было висячих вершин в графе сделаем $q \geq 2$. Пример графа при $n = 3$ и $k = 2$ приведен на рисунке 2.

Лемма 3. *Обхват полученного графа равен 8.*

Случаи когда цикл расположен в одном базисном графе рассмотрены ранее.

Пусть цикл не содержит вершин из дополнительного множества и не лежит полностью в одном базисном графе. Тогда он лежит, минимум в двух базисных графах. Следовательно, ему принадлежат вершины из четвертых слоев, минимум, двух базисных графов. Граф двудольный и, следовательно, ему принадлежат минимум 4 символьные вершины. Таким образом длины цикла не меньше 8.

Пусть цикл содержит одну вершину v_1 из дополнительного множества. Вершина v_1 смежна с вершинами четвертого слоя не соседних базисных графов. Но путь между четвертыми слоями не соседних базисных

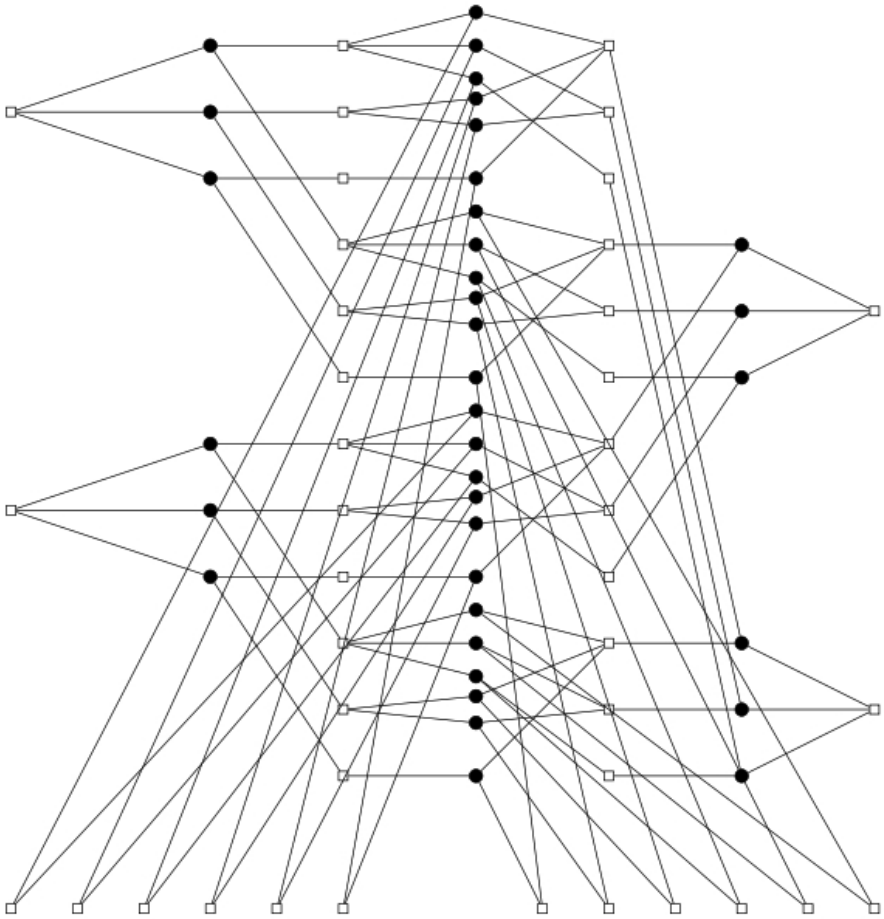


Рис.2

графов без использования допонительных вершин имеет длину больше 8. Следовательно, цикл имеет длину больше 10.

Пусть циклу принадлежат, минимум, 2 вершины из дополнительно множества. Тогда в силу построения циклу принадлежит 4 вершины из четвертых базисных слоев. Следовательно, длина цикла будет больше 8.

Таким образом, в графе нет циклов длины 4 и 6 и лемма доказана.

Оценим скорость кода для полученного графа.

$$v = 1 - \frac{2q(1+2n)+n^2+n}{2q(\frac{n^2+n}{2}+n)} = 1 - \frac{2q(1+2n)+n^2+n}{q(n^2+3n)}$$

Можно заметить, что при увеличении q доля дополнительных вершин, среди всех проверочных, стремится к 0:

$$\lim_{q->\infty} v = \lim_{q->\infty} (1 - \frac{2q(1+2n)+n^2+n}{q(n^2+3n)}) = 1 - \frac{2+4n}{n^2+3n}$$

Таким образом доказана следующая теорема:

Теорема 3. *Для любой скорости кода v существуют такие n, q и двудольный граф задающий проверочную матрицу LDPC кода, у которой все символьные вершины имеют степень 3, и скорость полученного кода $v_1 > v$.*

4. Вторая модификация: построение семейства графов G_2 со степенями символьных вершин равными 4-м

Будем использовать конструкция подобную той, что была описана в предыдущем разделе. Только теперь вместо двух дополнительных наборов проверочных вершин возьмем p наборов по $\frac{n^2+n}{2}$, где число p зависит только от q - числа копий базисного графа.

Суть построения заключается в том, чтобы соединить вершины из дополнительного набора с вершинами четвертых слоев базисных графов так, чтобы каждая вершина из четвертого слоя была соединена с двумя дополнительными вершинами, вместо одной, как это было при прошлом построении. При этом не должно остаться висячих вершин и обхват графа должен по-прежнему быть равен 8-ми.

Введем следующее обозначение: Пронумеруем базисные графы числами $\{1 \dots q\}$ и дополнительные наборы числами $\{1 \dots p\}$. Тогда запись $\{(a_{11} \dots a_{1w_1})(a_{21} \dots a_{2w_2}) \dots \ \dots (a_{p1} \dots a_{pw_p})\}$ обозначает, что вершины первого дополнительного набора соединены с вершинами четвертого слоя базисных графов под номерами $(a_{1,1} \dots a_{1,w_1})$ и так далее до p-ого дополнительного слоя.

Таким образом, в предыдущем построении было соединение типа: $\{(0\ 2\ \dots 2q-2)(1\ 3\ \dots 2q-1)\}$.

Перейдем непосредственно к построению. Для этого рассмотрим несколько случаев, которые могут привести к появлению циклов длины меньше 8-ми.

А) Для любых различных базисных графов с номерами $i, j, k \in \overline{1 \dots q}$ не существует трех дополнительных наборов типа: $(\dots i \dots j \dots)$, $(\dots i \dots k \dots)$

и $(\dots j \dots k \dots)$. Иначе опять на выходе получится цикл длины 6(Рисунок 3).

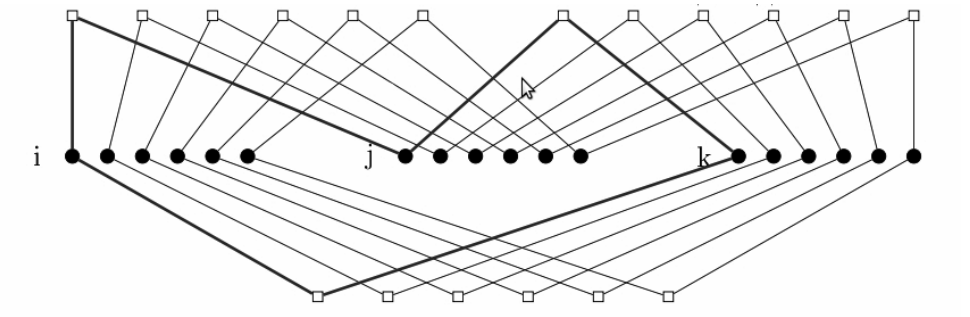


Рис.3

B) Из одного дополнительного набора, не должны выходить ребра в соседние базисные графы. Т.е. не должно быть соединения типа: $(\dots i\ i+1\dots)$. Иначе на выходе получаются циклы длины 6 (Рисунок 4).

C) Для любых различных базисных графов с номерами $i, j \in \overline{1\dots q}$ существует не более одного дополнительного набора, который соединен с ними. Иначе на выходе получатся циклы длины 4 (Рисунок 4).

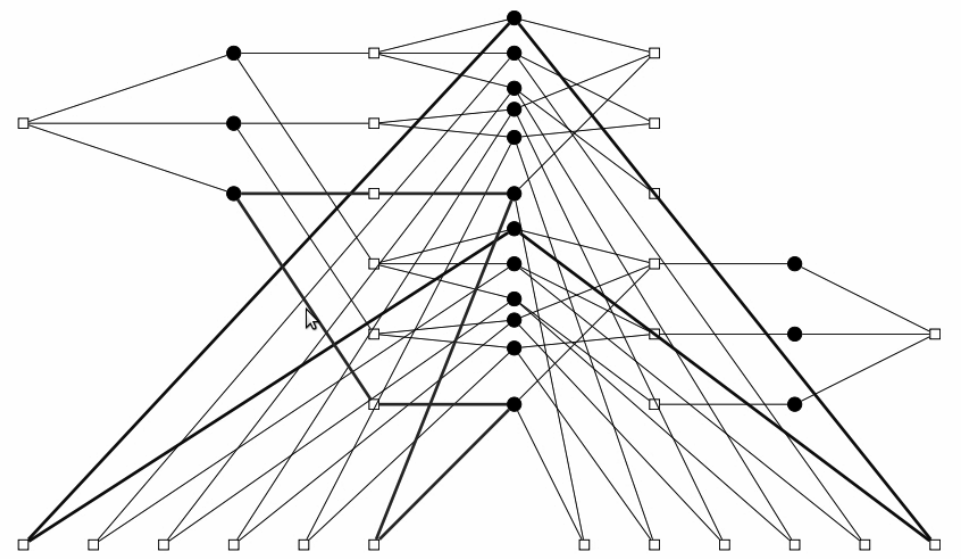


Рис.4

Таким образом, можно сформулировать следующую лемму:

Лемма 4. *Для некоторых q, n, существует p такое, что граф построенный с помощью данного алгоритма с соединениями типа $\{(a_{11} \dots a_{1w_1})(a_{21} \dots a_{2w_2}) \dots \dots (a_{p1} \dots a_{pw_p})\}$, и эти соединения удовлетворяют условиям **A, B, C**, имеет обхват 8.*

Будем рассматривать только простые циклы.

Если цикл не проходит через вершины дополнительного набора, то его длина больше 6. Это доказывалось раньше.

Если цикл проходит через одну вершины дополнительного набора, то единственный случай, когда может появиться цикл длины 6, это когда в одном соединении лежат два соседних базисных графов. Но это противоречит пункту **B**. Если в одном соединении нет соседних базисных графов, то длина подобных циклов будет больше 6-ти, т.к. расстояние между четвертыми слоями не соседних базисных графов больше 8-ми.

Если цикл содержит только две вершины из одного дополнительного набора, то длина цикла будет больше 8-ми, т.к. эти 2 вершины соединены с 4-мя разными вершинами из четвертых слоев базисных графов. Следовательно, в силу двудольности графа длина цикла будет не меньше 8-ми.

Если цикл содержит две вершины $v_1,\ v_2$ из разных дополнительных наборов. Если их соединения не имеют общих базисных графов, то цикл, очевидно, будет больше 8. Если они имеют единственное общее соединение с графом под номером i. Тогда $v_1,\ v_2$ могут иметь общую смежную вершину из графа i (иначе можно свести к предыдущему пункту). Также $v_1,\ v_2$ соединены с другими различными графами $j,\ k$ соответственно. Но путь соединяющий четвертые слои базисных графов без использования вершин дополнительного множества содержит больше четырех вершин. Следовательно, длина всех таких циклов будет больше 6.

Если цикл содержит 3 вершины из разных дополнительных наборов. Тогда нет соединений типа $(\dots i \dots j \dots)$, $(\dots i \dots k \dots)$ и $(\dots j \dots k \dots)$. Следовательно, цикл содержит вершины, по крайней мере, четырех разных базисных графов и длина цикла будет не меньше 8.

Если цикл содержит больше трех вершины из дополнительных наборов, то его длина будет больше 8-ми, в силу двудольности графа.

Следовательно, в графе нет циклов длины 4 и 6.

Пусть q- четное. Возьмем $p = q$ и разбиение типа $\{(0,2)(1,3)...(2q-2,0)(2q-1,1)\}$. Соединение удовлетворяет **A, B, C**. Следовательно, существуют графы удовлетворяющие заданным условиям и имеющие обхват 8. Лемма доказана.

5. Подсчет скорости и оптимизация числа p

Скорость кода на заданном графе равна:

$$v = 1 - \frac{q(1+2n) + p(\frac{n^2+n}{2})}{q(\frac{n^2+n}{2} + n)}$$

Выбор параметра p играет важную роль для определения скорости, поэтому необходимо сделать его как можно меньше.

Предлагается следующий подход: пусть задано некоторое $q > 8$, тогда возьмем начальное соединение такого типа $\{(0)(1\ 3)(2\ 4\ 6)...\}$ и покрываем все множество $\overline{0 \dots q-1}$. Чередуем блоки с четными и нечетными числами увеличивая длину блока до тех пор, пока не дойдем до q. Последние два блока могут быть неполными. Посчитаем количество блоков, которые необходимо для покрытия множества. Данный набор удовлетворяет условиям **A, B, C**.

Рассмотрим последовательность a_i $i \geq 0$ самых больших чисел в каждом блоке: $0, 3, 6, 11, 16, \dots$. Тогда последовательность $a_{i+1} - a_i$ имеет вид: $3, 3, 5, 5, 7, \dots$. Найдем общий вид этих чисел в четных и нечетных подпоследовательностях. В четном блоке имеет вид $a_{2N} = 2(\frac{3+(2N+1)}{2}N) = 2N^2 + 4N$. В нечетном блоке $a_{2N+1} = a_{2N} + (3 + 2N) = 2N^2 + 6N + 3$. Последовательность a_i - возрастающая. Следовательно, $a_{i-1} < q \leq a_i$ и нам необходимо $i + 1$ набор. Отметим, что если последний неполный блок имеет 1 элемент, то предпоследний блок тоже будет неполным. Тогда вместо q возьмем $q + 2$ при этом колличество блоком неувеличится, а в последнем блоке будет 2 элемента.

На данном этапе мы имеем только одно соединение базисных графов с дополнительными наборами. Построим второе соединение. Мы имеем $i+1$ блоков $\{(0)(1\ 3)(2\ 4\ 6)\dots\}$. Добавим еще i блоков следующим образом: первый новый блок имеет вид: $(0\ 3\ 6\dots)$ возьмем первый элемент из первого блока, второй элемент из второго блока и так далее, пока это возможно. Получаем, что в первом блоке нет элемента из последнего блока предыдущего соединения. Второй новый блок $(1\ 4\dots)$ т.е. первый элемент второго блока, второй элемент третьего блока и так далее. Таким образом заполняем все $i+1$ блоков. В последнем будет один элемент, который был первым в последнем блоке в прошлом соединении. Получаем второе начально соединение типа $\{(0\ 3\ 6\dots)(1\ 4\dots)(2\dots)\dots\}$. Второе соединение также удовлетворяет **A, B, C**.

Теперь объединяем первое и второе начальное соединения, при этом, отождествляя первый единичный блок из первого и последний единич-

ный блок из второго. Таким образом получим соединение с $p = 2i + 1$ блоком типа $\{(1\ 3)(2\ 4\ 6)\dots(0\ 3\ 6\dots)(1\ 4\dots)(2\dots)\dots(0\ j)\}$, где j единственный элемент последнего блока во втором соединении.

Такое построение обусловлено тем, чтобы полученное соединение удовлетворяло условиям **A, B, C**. Очевидно, что в каждом блоке нет соседних элементов, поэтому **B** выполнено, если $q \neq 2N^2 + 4N$ и $q \neq 2N^2+6N+3$. Это необходимо, что первый и последний элемент не попали в один блок. Так как при построении второго начального соединения мы объединяли в блоки элементы из разных блоков в первом соединении, то в совокупности получим соединение удовлетворяющее условию **C**. Рассмотрим соединение полученное из последнего удалением блока $(0\ j)$. Предположим, существует три блока, которые не удовлетворяют условию **A**. Т.е. существует элементы с номерами $i, j, k \in \overline{0\dots q-1}$ и блоки: $(\dots i \dots j \dots)$, $(\dots i \dots k \dots)$ и $(\dots j \dots k \dots)$. Так как первый и второй блоки имеют общий элемент i, они изначально принадлежали разными начальным соединениям. Пусть первый блок принадлежал первому соединению, второй блок второму. Третий блок имеет элемент j, следовательно, он не принадлежит первому соединению. Но третий блок также содержит элемент k, следовательно, второму соединению он также не принадлежит. Следовательно, получаем противоречие и полученное соединение удовлетворяет **A**. Таким образом, если полное соединение не удовлетворяет **A**, то «неправильными» блоками являются $(0\ j)$, последний блок первого начального соединения и первый блок второго начально соединения. Но так как первый блок из второго начально соединения не содержит элементов из последнего блока первого начально разбиения, получаем противоречие. Следовательно, полное разбиение удовлетворяет **A** и полученный граф имеет обхват 8.

Перейдем к подсчету скорости. Пусть $i = 2N$ четное и $a_{2N-1} < q < a_{2N}$. Тогда:

$$2(N-1)^2 + 6(N-1) + 3 < q \leq 2N^2 + 4N$$

$$2N^2 + 2N - 1 < q < 2N^2 + 4N$$

Так как $N, q \in \mathbb{N}$, получаем, что $\lfloor\sqrt{3+2q} - 1\rfloor \geq N \geq \lceil\sqrt{4+2q} - 2\rceil$. Соответственно $p = 2i+1 = 4N+1 \leq 4\lfloor\sqrt{3+2q}-1\rfloor+1 = 4\lfloor\sqrt{3+2q}\rfloor-3$. С другой стороны $p > 4\lceil\sqrt{4+2q}\rceil - 5$ В результате мы имеем:

$$v = 1 - \frac{q(1+2n) + p(\frac{n^2+n}{2})}{q(\frac{n^2+n}{2} + n)} \geq 1 - \frac{q(1+2n) + (4\lfloor\sqrt{3+2q}\rfloor - 3)(\frac{n^2+n}{2})}{q(\frac{n^2+n}{2} + n)}$$

Видно, что при увеличении q скорость стремится к отношению скорости на одном базисном графе:

$$\lim_{q->\infty} v = 1 - \frac{2+4n}{n^2+3n}$$

Обозначим через p_3 и p_4 доли символьных вершин, степени 3 и 4 соответственно, в получившемся графе. Тогда, в такой конструкции максимальная доля символьных вершин имеющих степень 4 равна

$$p_4^m = \frac{\frac{n^2+n}{2}}{\frac{n^2+n}{2}+n} = \frac{n^2+n}{n^2+3n} = 1 - \frac{2n}{n^2+3n}$$

Для построения графа с произвольной долей p_4 вершин степени 4, необходимо последовательно заполнять второе начальное соединение пока доля вершин степени не достигнет заданного значения, с той лишь оговоркой, что блок $(1, j)$ необходим, чтобы в конструкции не было висячих вершин. Поэтому будет 1 блок вершин степени четыре необходим. $v_4^0 = \frac{\frac{n^2+n}{2}}{q(\frac{n^2+n}{2}+n)} = \frac{n^2+n}{qn^2+3qn} = \frac{1}{q} - \frac{2n}{qn^2+3qn}$ Следовательно:

$$\frac{1}{q} - \frac{2n}{qn^2+3qn} = v_4^0 \le v_4 \le v_4^m = 1 - \frac{2n}{n^2+3n}$$

Можно заметить, что с увеличением q и n растет интервал выбора скорости и отрезок выбора доля вершин степени четыре. Таким образом можно сформулировать следующую теорему:

Теорема 4. *Для любой скорости кода v и доли символьных вершин степени p_4 существуют такие n, q и двудольный граф задающий проверочную матрицу LDPC кода, у которой все символьные вершины имеют степень 3 или 4, причем доля вершин степени 4 приблизительно равна p_4, а скорость полученного кода $v_1 > v$.*

Список литературы

[1] Shannon C.E. A Mathematical Theory of Communication // Bell System Technical Journal. — 1948. — Т. 27. — С. 379-423, 623–656.

[2] Gallager, R. G. Low Density Parity Check Codes. — Cambridge: M.I.T. Press, 1963. — P. 90.

[3] http://www.inference.phy.cam.ac.uk/mackay/otherECC.html

[4] Gallager R.G. Low-Density Parity-Check Codes. Cambridge, MA, MIT Press, 1963.

[5] Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. — М.: Техносфера, 2006. — 320с.

About methods of constructing LDPC-codes with preset characteristics

Ananiev K.Y.

The work presents algoritms for building test matrices for LDPC, which are codes based on a Tanner graph with a girth of 8. Other parameters of the graph, apart from the girth, include the division of degrees of character vertices: the ratio of portion of vertices with degrees 3 and 4 to their total number, as well as the speed of the code generated. The code is built for random speed and random division in linear time depending on the number of elements of the matrix.

Keywords: LDPC-codes, bipartite graph, division of degrees of vertices.

О длине минимальной алфавитной склейки для класса линейных регулярных языков

Дергач П.С., Раджабов Ж.И.

В кандидатской диссертации [1] была поставлена и решена задача о нахождении верхней оценки на минимальную длину слов из регулярного языка, склеивающихся (то есть имеющих совпадающий образ) при алфавитном кодировании (если такая склейка вообще существует). В данной статье исследуется задача о нахождении соответствующих нижних оценок на длину склейки для случая, когда регулярные языки имеют линейную функцию роста, а схема кодирования преобразует все буквы входного алфавита в один и тот же символ. Для такого кодирования образ слова однозначно определяется по его длине. Приводятся нижние оценки, совпадающие по порядку с верхними оценками из [1] для таких языков и такого кодирования. Кроме того, для этого подслучая приводится более точная верхняя оценка.

Ключевые слова: алфавитное кодирование, регулярный язык, склейка.

Введение

В работе [1] решается проблема проверки однозначности алфавитного декодирования в классе регулярных языков с некоторыми ограничениями на функцию роста. Вполне естественно, что для этого при условии существования склейки строятся верхние оценки на ее длину. Однако, вопрос о соответствующих нижних оценках тоже представляет отдельный научный интерес. Поскольку в общем случае решить эту задачу сложно, то для первого приближения было решено рассмотреть класс регулярных языков с линейной функцией роста со схемой кодирования, преобразующей все буквы входного алфавита в один и тот же символ.О решении похожих задач можно прочитать в статьях [2-8]. О других интересных аспектах исследований авторов и других ученых в смежных областях к тематике данной работы можно прочитать в [9-20].

Основные определения

Пусть $A = B = \{0, 1\}$. Пусть P_1, P_2 — непустые множества слов в алфавите A. Определим следующие операции над P_1 и P_2 :

1) *Объединение* множеств P_1 и P_2 (обозначение $P_1 \bigcup P_2$) есть множество всех слов вида α, где $\alpha \in P_1$ или $\alpha \in P_2$.

2) *Конкатенация* множеств P_1 и P_2 (обозначение $P_1 \cdot P_2$) есть множество всех слов вида $\alpha_1\alpha_2$, где $\alpha_1 \in P_1,\ \alpha_2 \in P_2$.

3) *Итерация* множества P_1 (обозначение $(P_1)^*$) есть множество всех слов вида $\alpha_1 \dots \alpha_k$, где $\alpha_1, \alpha_2, \dots, \alpha_k \in P_1,\ k \geqslant 0$. При $k = 0$ здесь имеется ввиду пустое слово λ.

Введем понятие регулярного языка в алфавите A. Называем множество $P,\ P \subseteq A^*$ *регулярным языком в алфавите* A, если его можно получить из пустого множества и одноэлементных однобуквенных множеств $\{a\},\ a \in A$ применением конечного числа конкатенаций, объединений и итераций. Более подробно, определение регулярных языков таково:

1) $\{a\}$, где a — произвольная буква алфавита A, — регулярные языки в алфавите A;

2) Если P_1, P_2 — регулярные языки в алфавите A, то и множества $P_1 \cup P_2$, $P_1 \cdot P_2$, $(P_1)^*$ — регулярные языки в алфавите A;

3) Регулярность произвольного языка в алфавите A устанавливается в соответствиями с пунктами (1)-(3) за конечное число шагов.

Множество регулярных языков в алфавите A обозначаем через $R(A)$.

Рассмотрим *схему алфавитного кодирования* $f : A \to B$, для которой $f(0) = f(1) = 0$. Далее эта схема кодирования доопределяется на произвольном языке $P \subset A^*$ следующим образом:

$$\tilde{f}(a_{i_1}a_{i_2} \dots a_{i_n}) = f(a_{i_1})f(a_{i_2}) \dots f(a_{i_n}) = 00 \dots 0.$$

Полученную функцию $\tilde{f} : A^* \to B^*$ называем *функцией алфавитного кодирования.*

Пусть $P \in R(A)$ и $\beta \in \tilde{f}(P)$. Тогда $\alpha \in P$ называется *расшифровкой* β *при алфавитном кодировании* $\tilde{f}$ *на регулярном языке* P или просто *расшифровкой* β, если $\tilde{f}(\alpha) = \beta$. Также говорим, что β — *код слова* α. Если для любых различных слов $\alpha_1, \alpha_2 \in P$ выполняется $\tilde{f}(\alpha_1) \neq \tilde{f}(\alpha_2)$,

то *декодирование однозначно на P по $\tilde{f}$*. Также говорим, что $\tilde{f}$ *биективно на P*. В противном случае говорим, что *в регулярном языке P есть склейка* (α_1, α_2). Под склейкой здесь понимается произвольная неупорядоченная пара различных слов языка P с одинаковым кодом. *Минимальной склейкой для языка P* называем склейку, доставляющую среди всех склеек языка P минимальное значение на максимум длин слов из склейки. А само это значение называем *размером минимальной склейки* и обозначаем его через $m(P)$. Для произвольной склейки ее *размером* также называем максимальную длину слов из этой склейки. Впрочем, очевидно, что для функции $\tilde{f}$ длины слов, образующих склейку, совпадают.

Пусть $\mathbf{E}$ — произвольное множество языков в алфавите A, каждый из которых имеет склейку. Через $m(\mathbf{E})$ обозначаем величину

$$m(\mathbf{E}) := \max_{P \in \mathbf{E}} m(P),$$

если, конечно, такой максимум существует.

Пусть $P \subseteq A^*$. Через $\mathbb{N}_0$ обозначаем множество $\mathbb{N} \cup \{0\}$. Для произвольного $n \in \mathbb{N}$ через $P_{\leqslant}(n)$ обозначаем множество слов из P, длина которых не превосходит n. Через $T_n(P)$ обозначаем мощность множества $P_{\leqslant}(n)$:

$$T_n(P) := |P_{\leqslant}(n)|.$$

Через T_P обозначаем функцию $T_P : \mathbb{N} \to \mathbb{N}_0$, где

$$T_P(n) := T_n(P)$$

для всех $n \in \mathbb{N}$. Называем T_P *функцией роста* для P. Говорим, что бесконечный язык P *имеет линейную функцию роста* и пишем $T_P \in Lin$, если функция T_P ограничена сверху каким-нибудь полиномом первой степени. Обозначаем класс бесконечных регулярных языков в алфавите A с линейной функцией роста через $LR(A)$. Из [1] известно, что всякий язык из класса $LR(A)$ представим в виде

$$\bigvee_{i=1}^{s} \alpha_i \beta_i^* \gamma_i, \tag{1}$$

где $\alpha_i, \beta_i, \gamma_i$ — слова в алфавите A и $\beta \neq \lambda$. Число s в этом представлении называется *его высотой.* Класс всех языков $P \in LR(A)$, представимых

выражением (1) с высотой s обозначим через $LR(A, s)$. *Сложностью представления (1)* называется число

$$L\left(\bigvee_{i=1}^{s} \alpha_i \beta_i^* \gamma_i\right) := \max_{i=1,\ldots,s} (|\tilde{f}(\alpha_i)| + |\tilde{f}(\beta_i)| + |\tilde{f}(\gamma_i)|).$$

Здесь под $|\alpha|$ имеется ввиду длина слова α, то есть количество букв в этом слове.

Сложностью языка $P \in LR(A, s)$ с линейной функцией роста называется минимальная сложность среди всех его представлений вида (1), имеющих высоту s. Для произвольного $k \in \mathbb{N}$ через $LR(A, s, k)$ обозначаем множество всех языков $P \in LR(A, s)$, имеющих сложность не выше k.

Утверждение 1. *Пусть* $k \in \mathbb{N}$, $k \geqslant 2$. *Тогда*

$$m(LR(A, 2, k)) \geqslant k(k-1).$$

Утверждение 2. *Пусть* $k > s > 2$. *Тогда*

$$m(LR(A, s, k)) \geqslant \left]\frac{(k+2-s)(k+1-s)}{s-1}\right[.$$

Утверждение 3. *Пусть* $k, s \in \mathbb{N}$, $k, s \geqslant 2$. *Тогда*

$$m(LR(A, s, k)) \leqslant 2k(k-1).$$

Доказательство утверждений

Лемма 1. *Пусть* $k \in \mathbb{N}$, $k \geqslant 2$, $P = \alpha_1 \beta_1^* \gamma_1 \vee \alpha_2 \beta_2^* \gamma_2$ *для некоторых* $\alpha_i, \beta_i, \gamma_i \in A^*, \beta_i \neq \lambda$, $|\alpha_i| + |\beta_i| + |\gamma_i| \leqslant k$ *и в* P *есть склейка. Тогда*

$$m(P) \leqslant 2k(k-1).$$

▷ **Доказательство леммы 1:**

Рассмотрим множество длин слов из языка P. Оно состоит из двух арифметических прогрессий l_1, l_2 с началом и шагом не выше k. Так как в P есть склейка, то эти прогрессии пересекаются. Отсюда, очевидно, следует, что они пересекаются по бесконечной арифметической прогрессии. Обозначим эту прогрессию через (a, b), где a — начало прогрессии, а b — ее шаг.

Покажем, что $a \leqslant k(k-1)$. Здесь возможны 2 случая. В первом из них оба шага прогрессий l_1, l_2 равны k и тогда $|\beta_1| = |\beta_2| = k$, а значит $\alpha_1 = \alpha_2 = \gamma_1 = \gamma_2 = \lambda$. Тогда прогрессии l_1, l_2 равны (k, k) и их пересечение (a, b) тоже равно (k, k). Очевидно, что $a = k \leqslant k(k-1)$. Во втором случае хотя бы один из шагов прогрессий l_1, l_2 меньше k и тогда утверждение следует из китайской теоремы об остатках, ведь НОК прыжков прогрессий l_1, l_2 в этом случае не превосходит $k(k-1)$.

Покажем, что $b \leqslant k(k-1)$. В первом случае (смотри выше) прогрессия (a, b) равна (k, k) и $b = k \leqslant k(k-1)$ — утверждение очевидно. Во втором случае утверждение, опять же, следует из китайской теоремы об остатках и того факта, что НОК прыжков прогрессий l_1, l_2 не превосходит $k(k-1)$.

Рассмотрим теперь два слова ρ_1, ρ_2 из языков $\alpha_1\beta_1^*\gamma_1$, $\alpha_2\beta_2^*\gamma_2$ соответственно, которые имеют длину a. Либо они образуют склейку (и в этом случае утверждение леммы очевидно), либо они совпадают. Если они, все-таки совпадают, то рассмотрим два других слова ρ_3, ρ_4 из языков $\alpha_1\beta_1^*\gamma_1$, $\alpha_2\beta_2^*\gamma_2$ соответственно, которые уже имеют длину $a + b$. Покажем от противного, что они образуют склейку. Для этого нам потребуется более внимательно посмотреть на структуру слов ρ_i. Пусть

$$\rho_1 = \alpha_1\beta_1^x\gamma_1,\ \rho_2 = \alpha_2\beta_1^y\gamma_2,\ \rho_3 = \alpha_1\beta_1^{x+z}\gamma_1,\ \rho_4 = \alpha_2\beta_2^{y+w}\gamma_2.$$

Введем ряд дополнительных обозначений

$$\gamma_1' := \alpha_1\beta_1^x,\ \gamma_2' := \alpha_2\beta_2^y,\ \delta_1 := \beta_1^z,\ \delta_2 := \beta_2^w.$$

Тогда получаем

$$\gamma_1'\gamma_1 = \rho_1 = \rho_2 = \gamma_2'\gamma_2,\ \gamma_1'\delta_1\gamma_1 = \rho_3 = \rho_4 = \gamma_2'\delta_2\gamma_2. \tag{2}$$

Без ограничения общности, $|\gamma_1| \leqslant |\gamma_2|$. Тогда

$$\gamma_2 = \gamma_3\gamma_1 \tag{3}$$

для некоторого (возможно, пустого) γ_3. Условия (2) с учетом (3) можно переписать в виде

$$\gamma_1' = \gamma_2'\gamma_3, \tag{4.1}$$

$$\gamma_1'\delta_1 = \gamma_2'\delta_2\gamma_3. \tag{4.2}$$

Подставив (4.1) в (4.2), получаем

$$\gamma_3\delta_1 = \delta_2\gamma_3. \tag{4.3}$$

Мы знаем, что у P есть какая-то склейка (ρ_5, ρ_6). Очевидно, что тогда она состоит из двух слов вида

$$\gamma_1'\delta_1^c\gamma_1 = \rho_5,\ \gamma_2'\delta_2^c\gamma_2 = \rho_6,$$

где $c \geqslant 2$. Но тогда из (3), (4.1-4.3) выводим

$$\rho_5 = \gamma_1'\delta_1^c\gamma_1 = \gamma_2'\gamma_3\delta_1^c\gamma_1 = \gamma_2'\gamma_3\delta_1\delta_1^{c-1}\gamma_1 = \gamma_2'\delta_2\gamma_3\delta_1^{c-1}\gamma_1 = \ldots =$$

$$= \gamma_2'\delta_2^{c-1}\gamma_3\delta_1\gamma_1 = \gamma_2'\delta_2^c\gamma_3\gamma_1 = \gamma_2'\delta_2^c\gamma_2 = \rho_6.$$

Полученное противоречие доказывает, что (ρ_3, ρ_4) — склейка. Осталось вспомнить, что размер этой склейки совпадает с длиной слов (ρ_3, ρ_4) и значит равен $a + b$. Но мы уже доказали, что $a, b \leqslant k(k-1)$. Поэтому $a + b \leqslant 2k(k-1)$. ■

Утверждение 1. *Пусть $k \in \mathbb{N}$, $k \geqslant 2$. Тогда*

$$m(LR(A, 2, k)) \geqslant k(k-1).$$

▷ **Доказательство утверждения 1:**

Для доказательства утверждения достаточно привести пример такого множества $P \in LR(A, 2, k)$, размер минимальной склейки в котором не меньше $k(k-1)$. Это верно, в частности для

$$P := (0^k)^* \cup (1^{k-1})^*,$$

так как $\text{НОК}(k, k-1) = k(k-1)$. ■

Утверждение 2. *Пусть $k > s > 2$. Тогда*

$$m(LR(A, s, k)) \geqslant \left]\frac{(k+2-s)(k+1-s)}{s-1}\right[.$$

▷ **Доказательство утверждения 2:**

Обозначим через $t(s,k)$ число $\left]\frac{(k+2-s)(k+1-s)}{s-1}\right[$. Для доказательства утверждения достаточно привести пример такого множества P из класса $LR(A,s,k)$, размер минимальной склейки в котором не меньше $t(s,k)$. Рассмотрим язык

$$P := (01^{k-1-s}0)^* \cup (01^{k-s}0)^* \cup \ldots (01^{k-3}0)^* \cup (01^{k-2}0)^*. \qquad (5)$$

Очевидно, что он принадлежит $LR(A,s,k)$. И у него есть склейки. Рассмотрим произвольную склейку (α_1, α_2) в языке (5). Тогда для некоторых $1 \leqslant i < j \leqslant s$ верно, что

$$\alpha_1 \in (01^{k+1-i}0)^*,\ \alpha_2 \in (01^{k+1-j}0)^*.$$

Из определения размера склейки следует, что размер склейки (α_1, α_2) равен длине слов α_1 и α_2. Но длина слова α_1 делится нацело на $k+1-i$, а длина слова α_2 делится нацело на $k+1-j$. Поэтому размер склейки делится на НОК$(k+1-i, k+1-j)$. Осталось заметить, что

$$\text{НОК}(k+1-i, k+1-j) = \frac{(k+1-i)(k+1-j)}{\text{НОД}(k+1-i, k+1-j)} \geqslant$$

$$\geqslant \frac{(k+2-s)(k+1-s)}{j-i} \geqslant \frac{(k+2-s)(k+1-s)}{s-1}.$$

А значит верно и что

$$\text{НОК}(k+1-i, k+1-j) \geqslant \left]\frac{(k+2-s)(k+1-s)}{s-1}\right[= t(s,k).$$

Поэтому размер минимальной склейки тоже не меньше $t(s,k)$. ■

Утверждение 3. *Пусть* $k, s \in \mathbb{N}$, $k, s \geqslant 2$. *Тогда*

$$m(LR(A,s,k)) \leqslant 2k(k-1).$$

▷ **Доказательство утверждения 3:**

Для доказательства утверждения достаточно показать, что размер минимальной склейки (если она есть) для произвольных языков из

класса $LR(A, s, k)$ не превосходит $2k(k-1)$. Рассмотрим любой язык $P \in LR(A, s, k)$. Значит

$$P = \bigvee_{i=1}^{s} \alpha_i \beta_i^* \gamma_i,$$

где $|\alpha_i| + |\beta_i| + |\gamma_i| \leqslant k$ при $i = 1, \ldots, s$. Пусть (δ_1, δ_2) — минимальная склейка языка P. Без ограничения общности, можно считать, что

$$\delta_1 \in \alpha_1 \beta_1^* \gamma_1,\ \delta_2 \in \alpha_2 \beta_2^* \gamma_2.$$

Тогда (δ_1, δ_2) будет минимальной склейкой и в множестве

$$P_1 := \alpha_1 \beta_1^* \gamma_1 \bigvee \alpha_2 \beta_2^* \gamma_2.$$

Для доказательства утверждения осталось применить лемму 1. ■

Список литературы

[1] П. С. Дергач. *Алфавитное кодирование регулярных языков с полиномиальной функцией роста.* Кандидатская диссертация, Москва, 2016.

[2] П. С. Дергач, Э. С. Айрапетов. *О прогрессивном разбиении некоторых подмножеств натурального ряда.* Интеллектуальные системы, 2015. Т.19, вып. 3, М., Сс. 79-86.

[3] П. С. Дергач. *О каноническом регулярном представлении S-тонких языков.* Интеллектуальные системы, 2014. Т.18, вып. 1, М., Сс. 211-242.системы, 2014. Т.18, вып. 1, М., Сс. 211-242.

[4] П. С. Дергач. *О проблеме вложения допустимых классов.* Интеллектуальные системы, 2015. Т.19, вып. 2, М., Сс. 143-174.

[5] П. С. Дергач. *О двух размерностях спектров тонких языков.* Интеллектуальные системы, 2015. Т.19, вып. 3, М., Сс. 155-174.

[6] П. С. Дергач, Э. С. Айрапетов. *О прогрессивном разбиении последовательности натуральных чисел, имеющей пропуск длины 2.* Интеллектуальные системы, 2016. Т.20, вып. 2, М., Сс. 67-86.

[7] П. С. Дергач, Е. Д. Данилевская. *О покрытиях и разбиениях натуральных чисел, имеющих два последовательных пропуска длины 1.* Интеллектуальные системы, 2017. Т.21, вып. 1, М., Сс.192-237.

[8] П. С. Дергач. *О структуре вложения прогрессивных множеств сложности два.* Интеллектуальные системы, 2017. Т.21, вып. 2, М., Сс.117-162.

[9] Д. Е. Александров. *Эффективные методы реализации проверки содержания сетевых пакетов регулярными выражениями.* Интеллектуальные системы, 2014. Т.18, вып. 1, М., Сс. 37-60.

[10] Д. Н. Бабин. *Частотные регулярные языки.* Интеллектуальные системы, 2014. Т.18, вып. 1, М., Сс. 205-210.

[11] Д. Е. Александров. *Об оценках автоматной сложности распознавания классов регулярных языков.* Интеллектуальные системы, 2014. Т.18, вып. 4, М., Сс. 161-190.

[12] В. М. Дементьев. *О звездной высоте регулярного языка и циклической сложности минимального автомата.* Интеллектуальные системы, 2014. Т.18, вып. 4, М., Сс. 215-222.

[13] И. Е. Иванов. *О сохранении периодических последовательностей автоматами с магазинной памятью с однобуквенным магазином.* Интеллектуальные системы, 2015. Т.19, вып. 1, М., Сс. 145-160.

[14] А. А. Петюшко. *О контекстно-свободных биграмных языках.* Интеллектуальные системы, 2015. Т.19, вып. 2, М., Сс. 187-208.

[15] И. Е. Иванов. *Нижняя оценка на максимальную длину периода выходной последовательности автономного автомата с магазинной памятью.* Интеллектуальные системы, 2015. Т.19, вып. 3, М., Сс. 175-194.

[16] В. А. Орлов. *О конечных автоматах с максимальной степенью различимости состояний.* Интеллектуальные системы, 2016. Т.20, вып. 1, М., Сс. 213-222.

[17] П. С. Дергач. *О проблеме проверки однозначности алфавитного декодирования в классе регулярных языков с полиномиальной функцией роста.* Интеллектуальные системы, 2016. Т.20, вып. 2, М., Сс. 147-202.

[18] А. М. Миронов. *Основные понятия теории вероятностных автоматов*. Интеллектуальные системы, 2016. Т.20, вып. 2, М., Сс. 283-330.

[19] А. А. Петюшко, Д. Н. Бабин. *Классификация Хомского для матриц биграммных языков*. Интеллектуальные системы, 2016. Т.20, вып. 2, М., Сс. 331-336.

[20] С. Б. Родин. *О связи линейно реализуемых автоматов и автоматов с максимальной вариативностью относительно кодирования состояний*. Интеллектуальные системы, 2016. Т.20, вып. 2, М., Сс. 337-348.

Сведения об авторах

Дергач Петр Сергеевич, Dergach Pyotr Sergeevich

Младший научный сотрудник МГУ имени М. В. Ломоносова в городе Москве;

адрес: Россия, г. Москва, 125565, Ленинградское ш., 88-19;

тел. моб.: +79037189288;

e-mail: dergachpes@mail.ru.

Раджабов Жахонгир Ихтиер угли, Radjabov Jakhongir Ikhtiyor ogli

Студент факультета ПМиИ филиала МГУ имени М. В. Ломоносова в городе Ташкенте;

адрес: Узбекистан, г. Ташкент, 100000, ул. Лашкарбеги, 1;

тел. моб.: +998977335030;

e-mail: karkidon@icloud.com.

On the length of a minimal alphabetical bonding in linear regular languages

Dergach P.S., Radjabov J.I.

In the Ph.D. thesis [1] it has been found the upper bound on the minimal length of two words in regular language with similar image under alphabetic coding (if such pair of words exists at all). In this paper, we investigate the problem of finding corresponding lower bounds in subcase when regular languages have a linear growth function, and the coding scheme transforms all the letters of the input alphabet into the same symbol. For such encoding, the image of any word is uniquely determined by its length. Below we give lower bounds that coincide in order with the upper bounds from [1] for such languages and such coding. In addition, a more accurate upper estimate is given for this subcase.

Keywords: alphabetic coding, regular language, bonding.

Об одном критерии полиномиальной полноты квазигрупп

Югай В.Л.

В работе формулируется и доказывается критерий полиномиальной полноты квазигрупп в терминах предполных классов k-значной логики.

Ключевые слова: квазигруппа, полиномиальная полнота, квазилинейность.

1. Введение

В последние годы наблюдается рост интереса к криптосистемам на основе квазигрупп (или, что эквивалентно, на основе латинских квадратов). В качестве примера можно привести работы [1], [2], [3], [4].

С криптографической точки зрения одним из самых важных свойств квазигрупп является полиномиальная полнота. Это обусловлено тем, что в функционально полной алгебре задача распознавания разрешимости системы уравнений является NP-полной ([5]). Известно, что квазигруппа полиномиально полна тогда и только тогда, когда она простая и неаффинная ([6]). Для ряда частных случаев построены более эффективные критерии: в работе [7] рассмотрен случай квазигрупп порядка 4, в работах [8], [9] предложен кубический алгоритм проверки полиномиальной полноты для случая квазигрупп простого порядка. В работе [10] проведено исследование связи свойств простоты и аффинности.

В настоящей работе предлагается критерий полиномиальной полноты, аналогичный критерию из работы [6], но сформулированный в терминах предполных классов k-значной логики.

2. Основные определения

Определение 1. *Квазигруппой* $(Q, *)$ *называется множество элементов* Q *с заданной на нем операцией* $* : Q \times Q \to Q$*, такой что для любых*

$a, b \in Q$ *уравнения*

$$a * x = b$$
$$y * a = b$$

имеют единственное решение.

В дальнейшем мы будем предполагать, что множество Q конечно.

Квазигрупповая операция может быть задана "таблицей умножения" — матрицей M порядка $|Q| \times |Q|$. Занумеруем элементы Q числами от 1 до $|Q|$: $Q = \{q_1, \ldots, q_{|Q|}\}$. Элемент матрицы, стоящий на пересечении строки номер i и столбца номер j, равен $q_i * q_j$. Матрица M называется латинским квадратом, связанным с квазигруппой $(Q, *)$. Несложно увидеть, что каждая строка и каждый столбец M является перестановкой на множестве Q.

Пусть $n \in \mathbb{N} \cup \{0\}$. Обозначим множество всех n-арных операций на множестве Q через P^n. В частности, P^0 — это множество всех констант, являющихся элементами Q. Положим $P = \bigcup\limits_{n=0}^{\infty} P^n$. Заметим, что функции из P можно рассматривать как функции логики значности $|Q|$ и естественным образом ввести операцию замыкания, обозначаемую квадратными скобками: если $F \subseteq P$, то $[F]$ — замыкание F ([11]).

Определение 2. *Квазигруппа $(Q, *)$ называется полиномиально (или функционально) полной, если*

$$\left[\{*\} \cup P^0\right] = P.$$

Определение 3. *Квазигруппа $(Q, *)$ называется простой, если операция $*$ не сохраняет никакое нетривиальтное отношение эквивалентности на множестве Q.*

Несложно увидеть, что все квазигруппы простого порядка являются простыми.

Определение 4. *Квазигруппа $(Q, *)$ называется аффинной, если на множестве Q можно ввести структуру абелевой группы $(Q, +)$, такую что*

$$x * y = \alpha(x) + \beta(y) + c,$$

где α, β — некоторые автоморфизмы группы $(Q, +)$, $c \in Q$.

Известно ([6]), что квазигруппа полиномиально полна тогда и только тогда, когда она проста и неаффинна.

Определение 5. *Функция от n переменных f^n на множестве Q из p^m элементов, где p — простое число, $m \in \mathbb{N}$, называется квазилинейной, если на Q можно ввести структуру конечного поля, относительно которой f^n представима в виде*

$$f^n(x_1, \ldots, x_n) = a_0 + \sum_{i=1}^{n} \sum_{j=0}^{m-1} a_{ij} \cdot x_i^{p^j},$$

В частности, если $m = 1$, такая функция называется линейной.

3. Критерий полиномиальной полноты квазигрупп

Теорема 1. *Квазигруппа $(Q, *)$ полиномиально полна тогда и только тогда, когда операция $*$ не сохраняет никакое нетривиальное отношение эквивалентности на Q и не является квазилинейной.*

Доказательство. Необходимость вытекает из замкнутости и неполноты классов сохранения отношений и классов квазилинейных функций, а также принадлежности всех констант всем таким классам.

Докажем достаточность. Пусть квазигруппа не является полиномиально полной. Тогда она либо не простая, либо аффинная. Если она не простая, то сохраняется нетривиальное отношение эквивалентности.

Пусть квазигруппа простая. Известно ([Предложение 3.2][10]), что простая квазигруппа может быть аффинной, только если порядок является степенью простого числа, а соответствующая абелева группа $(Q, +)$ является примарной. Таким образом, для доказательства достаточности осталось показать, что если квазигруппа аффинна над примарной группой, то квазигрупповая операция квазилинейна.

Пусть $x * y = \alpha(x) + \beta(y) + c$ для некоторой примарной абелевой группы $(Q, +)$, $\alpha, \beta \in Aut((Q, +))$, $c \in Q$. В [Лемма 5.2.4.2][11] показано, что функция $f^n \in P^n$ квазилинейна тогда и только тогда, когда для любых $(a_1, \ldots, a_n), (b_1, \ldots, b_n) \in Q^n$ выполнено равенство

$$f^n(a_1 + b_1, \ldots, a_n + b_n) + f(0, \ldots, 0) = f(a_1, \ldots, a_n) + f(b_1, \ldots, b_n),$$

где 0 — нейтральный элемент абелевой группы. Из этого факта вытекает квазилинейность автоморфизмов α и β. Следовательно, и функция $x * y$ квазилинейна. ■

В рамках обозначений предполных классов, принятых в работе [11], теорема может быть переформулирована следующим образом.

Следствие 1. *Квазигруппа* $(Q, *)$ *полиномиальнo полна тогда и только тогда, когда операция* $*$ *не лежит ни в одном из классов типа* $\mathfrak{U}$ *(сохранения нетривиального отношения эквивалентности) и* $\mathfrak{L}$ *(квазилинейных функций).*

В случае, когда порядок квазигруппы простой, в работах [8], [9] показано, что линейность двухместной функции $x * y$ эквивалентна одновременной линейности функций вида $x * a$ и $b * y$ для всевозможных $a, b \in Q$. В случае, когда порядок является степенью простого числа, это вообще говоря неверно. В качестве примера можно рассмотреть случай $p = m = 2$. В работе [7] показано, что полиномиально полные квазигруппы порядка 4 существуют. В силу доказанного критерия, квазигрупповые операции при этом не квазилинейны. Однако, как несложно увидеть, все 24 перестановки порядка 4 квазилинейны.

Автор выражает глубокую благодарность своему научному руководителю, к.ф.-м.н., с.н.с, А.В. Галатенко за постановку задачи и внимание к работе.

Список литературы

[1] М.М. Глухов. *О применениях квазигрупп в криптографии* // ПДМ, 2008, №2(2), 28–32.

[2] V. Shcherbacov. *Quasigroup based crypto-algorithms* // arXiv:1201.3016.

[3] S. Markovski, D. Gligoroski, V. Bakeva. *Quasigroup String Processing: Part 1* // Proc. of Maked. Academ. of Sci. and Arts for Math. And Tech. Sci. 1999. Vol. XX, 1–2. P. 13—28.

[4] S. Markovski, V. Kusacatov. *Quasigroup String Processing: Part 2* // Proc. of Maked. Academ. of Sci. and Arts for Math. and Tech. Sci. 2000. Vol. XXI, 1–2. P. 15–32.

[5] G. Horváth, C. L. Nehaniv, Cs. Szabó. *An assertion concerning functionally complete algebras and NP-completeness* // Theoretical Computer Science. 2008. Vol. 407, 1–3. P. 591—595.

[6] J. Hagemann , C. Herrmann. *Arithmetical locally equational classes and representation of partial functions* // Universal Algebra, Esztergom (Hungary), 1982. Vol. 29, Colloq. Math. Soc. Janos Bolyai, P. 345—360.

[7] V.A. Artamonov, S. Chakrabarti, S. Gangopadhyay, S.K. Pal. *On Latin squares of polynomially complete quasigroups and quasigroups generated by shifts* // Quasigroups and Related Systems. 2013. Vol. 21, 2. P. 117—130.

[8] А.В. Галатенко, А.Е. Панкратьев, С.Б. Родин. *О полиномиально полных квазигруппах простого порядка* // Интеллектуальные системы. Теория и приложения. 2016. Т. 20, Вып 3. С. 194—198.

[9] А.В. Галатенко, А.Е. Панкратьев, С.Б. Родин. *О полиномиально полных квазигруппах простого порядка* // Алгебра и логика. Принято к печати.

[10] V.A. Artamonov, S. Chakrabarti, S.K. Pal. *Characterizations of highly non-associative quasigroups and associative triples* // Quasigroups and Related Systems. 2017. Vol. 25, 1. P. 1—19.

[11] D. Lau. *Function algebras on finite sets: a basic course on many-valued logic and clone theory.* Springer, 2006.

A criterion for polynomial completeness of quasigroups

Yugay V.L.

We formulate and prove a criterion for the polynomial completeness of quasigroups in terms of precomplete classes of k-valued logic.

Keywords: quasigroup, polynomial completeness, quasilinearity.

К сведению авторов публикаций в журнале «Интеллектуальные системы. Теория и приложения»

В соответствии с требованиями ВАК РФ к изданиям, входящим в перечень ведущих рецензируемых научных журналов и изданий, в которых могут быть опубликованы основные научные результаты диссертаций на соискание ученой степени доктора и кандидата наук, статьи в журнал «Интеллектуальные системы. Теория и приложения» предоставляются авторами в следующей форме:

1. Статьи, набранные в пакете LaTeX, предоставляются к загрузке через WEB-форму http://intsysjournal.org/generator_form .
2. К статье прилагаются файлы, содержащие название статьи на русском и английском языках, аннотацию на русском и английском языках (не более 50 слов), список ключевых слов на русском и английском языках (не более 20 слов), информация об авторах: Ф.И.О. полностью, место работы, должность, ученая степень и/или звание (если имеется), контактные телефоны (с кодом города и страны), e-mail, почтовый адрес с индексом города (домашний или служебный).
3. Список литературы оформляется в едином формате, установленном системой Российского индекса научного цитирования.
4. За публикацию статей в журнале «Интеллектуальные системы. Теория и приложения» с авторов (в том числе аспирантов высших учебных заведений) статей, рекомендованных к публикации, плата не взимается. Оттиски статей авторам не предоставляются. Журнал распространяется по подписке, экземпляры журнала рассылаются подписчикам наложенным платежом. Условия подписки публикуются в каталоге НТИ «Роспечать», индекс журнала 64559.
5. Доступ к электронной версии последнего вышедшего номера осуществляется через НЭБ «Российский индекс научного цитирования». Номера, вышедшие ранее, размещаются на сайте http://intsysjournal.org, и доступ к ним бесплатный. Там же будут размещены аннотации всех публикуемых статей.

Подписано в печать: 20.09.2017
Дата выхода: 30.09.2017
Тираж: 200 зкз.
Цена свободная
Свидетельство о регистрации СМИ: ПИ № ФС77–58444 от 25 июня 2014 г., выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций(Роскомнадзор).