Математика и её приложения: ЖИМО. 2009. Вып. 1 (6). С. 129 – 138.

УДК 512.54

А. А. Толстопятов[1]

Возможные подходы к разбиению файла на буферы при булевом сжатии

Ключевые слова: булево сжатие, поле кратности, поле порядка.

Обсуждается возможность использования для задачи разбиения файла на буферы методов распознавания образов и квантовых алгоритмов.

We discuss the problem of application methods of pattern recognition and quantum algorithm for the splitting a file on buffers.

Введение

Главным условием, при выполнении которого возможно булево сжатие файла, является существование решений у кодирующего уравнения. Но кодирующее уравнение пишется не для файла, а для его разбиения, причем такого, чтобы параметры разбиения попадали в интервалы, обеспечивающие значение коэффициента сжатия больше единицы. Между тем для одного и того же файла существует экспоненциально большое число разных разбиений, для каждого из которых будет существовать свое кодирующее уравнение. Условия, обеспечивающие сжатие файла, если — и это вовсе не гарантировано, кодирующее уравнение будет иметь решения, несколько ограничивает число разных разбиений [3]; но это число все еще окажется настолько большим, что полный перебор невозможен. С другой стороны, если не делать полного перебора, а взять достаточно случайным образом какое-то разбиение, то из того, что для этого разбиения кодирующее уравнение не имеет решений, вовсе не следует, что их нет у кодирующего уравнения для другого разбиения того же самого файла. Именно такие, вообще говоря, случайные разбиения файла и рассматривались в [3]. Они строились на том основании, что каждый буфер должен содержать такое число кортежей, при котором коэффициент сжатия этого буфера будет максимальным. При этом не рассматривается вопрос о том, будет ли такое разбиение давать кодирующее уравнение с непустым множеством решений. Хотя в [3] и обсуждается возможность передвижения границ между буферами так, чтобы не нарушалось условие сжатия каждого буфера, но ведь само это условие хотя и является достаточным, если, — а это опять не гарантировано, кодирующее уравнение имеет решение, но не является необходимым. Поэтому возникает задача о поиске иных, более эффективных возможностей разбиения файла на буферы при

[1]Ивановский государственный университет; E-mail: khash2@mail.ru. Работа выполнена при финансовой поддержке РФФИ (проект 07-07-00155).

булевом сжатии. Настоящая работа не претендует на решение этой, находящейся на самых первых этапах исследования задачи, а посвящена обсуждению двух, как нам представляется, сейчас наиболее перспективных подходов — квантовым алгоритмам и методам распознавания образов.

1. Постановка задачи о разбиении файла на буферы

Введем следующие понятия и обозначения. Файл — последовательность нулей и единиц. Файл разбит на кортежи равной длины. Кортежи объединяются в буферы, если между последовательно идущими кортежами устанавливается граница. Буферы могут состоять из разного числа кортежей. Вот такое установление границ между последовательно идущими кортежами будем называть разбиением файла на буферы. Введем следующие обозначения:

N_Φ — длина файла в битах;

n — длина кортежей в битах;

N — число кортежей в файле;

L — число буферов при данном разбиении;

m_l — число кортежей в l-м буфере, $l = 1, 2, \ldots, L$;

s_l — число различных кортежей в l-м буфере, $l = 1, 2, \ldots, L$;

$n_k^l,\ l = 1, 2, \ldots, L,\ k = 1, 2, \ldots, s_l,$ — число повторов k-го кортежа в l-м буфере;

$x_k,\ k = 1, 2, \ldots, n,$ — булевы переменные;

$f_l(x_k),\ k = 1, 2, \ldots, n,\ l = 1, 2, \ldots, L,$ — булев полином, кодирующий поле принадлежности l-го буфера;

P — число порождающих булевых полиномов;

$\phi_p(x_k), p = 1, 2, \ldots, P,\ k = 1, 2, \ldots, n,$ — порождающие булевы полиномы;

I — число булевых переменных, от которых зависит кодирующий полином;

$e_i, i = 1, 2, \ldots, I,$ — булевы переменные, от которых зависит кодирующий полином;

$F(e_i)$ — кодирующий полином;

$k_l,\ l = 1, 2, \ldots, L,$ — коэффициент сжатия l-го буфера;

k — коэффициент сжатия всего файла.

Разбиение файла задается последовательностью $m_1, m_2, \ldots, m_L$ чисел кортежей в буферах. Выделенные выше величины связаны очевидными соотношениями:

$$N_\Phi = nN, \tag{1}$$

$$N = \sum_{l=1}^{L} m_l, \tag{2}$$

$$m_l = \sum_{k=1}^{s_l} n_k^l. \tag{3}$$

Код всего файла состоит из одного общего поля и L кодов каждого буфера. Код отдельного буфера состоит из 3-х полей: поля принадлежности, поля кратности и поля порядка. Если поле кратности кодировать одним числом, то для коэффициента сжатия файла k будем иметь (см. [3]):

$$k = \frac{n\sum_{l=1}^{L} m_l}{2^I + 2^n \cdot P + LI\log_2 P + \log_2 \prod_{l=1}^{L} \frac{C_{m_l-1}^{s_l-1} m_l!}{\prod_{k=1}^{s_l} n_k^l!}}; \tag{4}$$

если ввести обозначения

$$\alpha = 2^I + 2^n \cdot P + LI\log_2 P, \tag{5}$$

$$\beta_l = \log_2 \prod_{l=1}^{L} \frac{D_{m_l-s_l}^{s_l} s_l! n_l!}{\prod_{k=1}^{s_l} n_k^l! \prod_{k=1}^{a_l} p_k^l!}, \tag{6}$$

$$\gamma_l = \beta_l/\alpha, \tag{7}$$

то коэффициент сжатия l-го буфера будет равен (см. [3])

$$k_l = \frac{nm_l}{\frac{\alpha}{L} + \beta_l}. \tag{8}$$

Связь между коэффициентами k и k_l дается формулой (см. [3])

$$k = \sum_{l=1}^{L} W_l k_l, \tag{9}$$

где

$$W_l = \frac{1 + L\gamma_l}{L(1 + \sum_{l=1}^{L} \gamma_l)}. \tag{10}$$

Так как из (10) следует, что

$$0 \leqslant W_l \leqslant 1, \tag{11}$$

$$\sum_{l=1}^{L} W_l = 1, \tag{12}$$

то коэффициенты W_l логично интерпретировать как вероятности того, что l-й буфер имеет коэффициент сжатия k_l. Поскольку, согласно (10), W_l зависит только от числа буферов L и величин γ_l, а γ_l, согласно (7),

есть отношение длин кодов полей кратности и порядка β_l к длине кодов общего поля и всех полей принадлежности L буферов α, то распределение вероятностей W_l зависит только от этого отношения.

Каждому буферу в код поля принадлежности ставится информация, позволяющая восстановить входящие в него кортежи. А именно, сначала — это коэффициенты булева полинома $f_l(x_k)$, такого, что уравнение

$$f_l(x_k) = 0 \tag{13}$$

имеет решения, совпадающие с кортежами, входящими в l-й буфер. Далее, пусть $\phi_p(x_k), p = 1, 2, \ldots, P$, — порождающие булевы полиномы, через которые можно выразить $f_l(x_k)$. Чтобы это сделать, рассмотрим булев полином $F(e_i)$ от e_i, $i = 1, 2, \ldots, I$, булевых переменных. Тогда, если вместо e_i подставить порождающие полиномы ϕ_p для l-го буфера, то должно выполняться кодирующее уравнение

$$F(e_k^l) = f_l \ . \tag{14}$$

Для сжатия файла должны выполняться два условия:

$$\begin{array}{ll} 1) & k > 1 \ , \\ 2) & \text{уравнение (14) имеет нетривиальное решение.} \end{array} \tag{15}$$

В результате решения уравнения (14) должны быть найдены полиномы $\phi_p(x_k)$, булевы переменные e_i^l и коэффициенты булева полинома $F(e_i)$.

Таким образом, параметрами, задающими разбиение файла на буферы являются n, I, P, L и последовательность чисел кортежей в буферах $m_1, m_2, \ldots, m_L$. Зная эти параметры из (1)–(3) можно найти N_Φ и N. Перечисленные выше параметры не полностью произвольны, а связаны определенными неравенствами, выражающимися из 1-го условия (15) (см. [3]):

$$P < L - 2^{-n}, \tag{16}$$

$$I < \frac{2^n(L-P)}{L \log_2 P}, \tag{17}$$

$$I < n + \log_2 P. \tag{18}$$

Наконец, можно найти неравенство, ограничивающее число буферов L как сверху так и снизу (см. [3]),

$$L_{\min} < L < L_{\max}, \tag{19}$$

где

$$L_{\min} = \frac{(2^{I-n} + P)n}{n^2 - (I + I \cdot 2^{-n} \log_2 P)n + \log_2 n}, \tag{20}$$

$$L_{\max} = \frac{(N_\Phi 2^{-n} - I \cdot 2^{-n} \log_2 P - P)n}{(1 + I \cdot 2^{-n} \log_2 P)n - \log_2 n}. \tag{21}$$

Неравенства (16)–(19) являются необходимым условием для выполнения 1-го условия (15), но они не определяют разбиение файла на буферы,

такое, чтобы выполнялось 2-е условие (15), а лишь несколько ограничивают полный перебор всех возможных разбиений, которых экспоненциально много.

Действительно, если забыть про (19), то можно разбить файл на один буфер, на два, и т. д. до N буферов. Это значит, что полное число разбиений $\mathcal{N}$ будет равно

$$\mathcal{N} = \sum_{L=0}^{N} C_N^L = 2^N. \tag{22}$$

Учет (19) несколько уменьшит (22), а именно:

$$\mathcal{N} = \sum_{L=L_{\min}}^{L_{\max}} C_N^L < 2^N, \tag{23}$$

где $L_{\min}$ дается формулой (20), а $L_{\max}$ — (21).

Обозначим через a_l^k ($l = 1, 2, \ldots, L;\ k = 1, 2, \ldots, m_l$) k-й кортеж, входящий в l-й буфер. Если, при фиксированном l вычеркнуть все одинаковые a_l^k, то мы получим файл (не исходный), разбитый на буферы, такие, что в каждый буфер входят разные кортежи. Обозначим их через $\tilde{a}_l^k$, где $l = 1, 2, \ldots, L,\ k = 1, 2, \ldots, s_l$. Задача заключается в том, чтобы найти такие разбиения файла на буферы, чтобы после факторизации по вхождению одинаковых кортежей в один буфер, т. е. после перехода $a_l^k \to \tilde{a}_l^k$, получающееся разбиение удовлетворяло условию (15), а значит неравенствам (16)–(19), или можно было определить, что такого разбиения нет. При этом задача не должна решаться путем полного перебора всех разбиений, которых, согласно (23), почти экспоненциально много.

2. Возможность использования квантовых алгоритмов

Одним из подходов к задачам с экспоненциально большим перебором, позволяющим сделать этот перебор полиномиальным, является построение квантовых алгоритмов [1, 2]. Пусть регистр квантового компьютера содержит K кубитов и описывается волновой функцией Ψ. Если i-й кубит описывается волновой функцией Ψ_i, где

$$\Psi_i = a_i|0> + b_i|1>, \tag{24}$$

причем a_i, b_i — комплексные числа, удовлетворяющие условию

$$|a_i|^2 + |b_i|^2 = 1, \tag{25}$$

то

$$\Psi = \prod_{i=1}^{K} \Psi_i\ . \tag{26}$$

Подставляя (24) в (26) найдем, что

$$\begin{array}{ll} \Psi = & a_1a_2\ldots a_{k-1}a_k|00\ldots00> + \\ + & a_1a_2\ldots a_{k-1}b_k|00\ldots01> + \\ + & \ldots \\ + & b_1b_2\ldots b_{k-1}a_k|11\ldots10> + \\ + & b_1b_2\ldots b_{k-1}b_k|11\ldots11> . \end{array} \tag{27}$$

Из (27) видно, что волновая функция ψ описывает все 2^K файлов длиной в K кубитов каждый. При этом вероятности W_0 того, что в результате вычислений получится файл $00\ldots00$, W_1 — файл $00\ldots01$, W_{2^K-2}— файл $11\ldots10$, W_{2^K-1}— файл $11\ldots11$ равны

$$\begin{array}{ll} W_0 & = |a_1a_2\ldots a_{k-1}a_k|^2, \\ W_1 & = |a_1a_2\ldots a_{k-1}b_k|^2, \\ & \ldots \\ W_{2^K-2} & = |b_1b_2\ldots b_{k-1}a_k|^2, \\ W_{2^K-1} & = |b_1b_2\ldots b_{k-1}b_k|^2. \end{array} \tag{28}$$

Задача квантового алгоритма состоит в следующем. Если в результате вычислений должен получиться файл с номером j, где $0 \leqslant j \leqslant 2^K - 1$, то, изменяя коэффициенты a_i и b_i путем вращения комплексной плоскости с осями $|0>$ и $|1>$, необходимо получить неравенство

$$W_j >> W_k,\ k = 0, \ldots, j-1, j+1, \ldots, 2^k - 1. \tag{29}$$

При этом, разумеется, должно выполняться равенство

$$\sum_{i=0}^{2^k-1} W_i = 1. \tag{30}$$

Таким образом, квантовые вычисления производятся не с одним файлом длиной K, а сразу со всеми возможными 2^K файлами такой же длины.

Если идею квантовых компьютеров приспособить к задаче поиска разбиениея файла на буферы, то мы сталкиваемся со следующей проблемой. Если кортежи имеют длину n, то существует 2^n различных кортежей, 2^{2^n} различных булевых полиномов и $2^{2^{2^n}}$ различных подмножеств всех булевых полиномов от n переменных. Поскольку поиск разбиения файла на буферы нужно вести именно на множестве всех подмножеств булевых полиномов, то это трижды экспоненциально сложная задача. Эту сложность можно уменьшить, если учесть, что согласно (22), есть 2^N различных разбиений файла, содержащего N кортежей.

Пусть выполнено неравенство

$$N_\Phi << n \cdot 2^{2^n}. \tag{31}$$

Так как $N = N_\Phi/n$, то, заменяя множество всех подмножеств булевых полиномов от n переменных на множество всех разбиений файла, содержащего N кортежей, получим существенное снижение сложности этой задачи.

Реально n может равняться 8 или 16; тогда (31) соответственно превратится в следующие неравенства:

$$\begin{array}{ll} N_\Phi << 2^{259}, & n = 8, \\ N_\Phi << 2^{65540}, & n = 16. \end{array} \tag{32}$$

Ясно, что неравенства (32) выполняются всегда.

Перенумеруем все возможные разбиения файла, содержащего N буферов. В результате получится последовательность нулей и единиц. Всего таких последовательностей будет 2^N. Заменим каждое двоичное число в этой последовательности на кубит, т. е. на волновую функцию Ψ_i из (24). Заметим, что каждое число разбиений $\mathcal{N}$ из (22), равное 2^N, естественно раскладывается на $N+1$ чисел, каждое из которых равно C_N^L, $L = 0, 1, \ldots, N$. Это значит, что мы будем иметь не один регистр, а $N+1$ регистров квантовых компьютеров, каждый из которых содержит L кубитов. Если отдельный кубит обозначить через $\mathcal{X}_1$, а регистр с L кубитами через $\mathcal{X}_L$, то требуемая конструкция для квантового алгоритма, разбивающего файлы на буферы, будет следующая:

$$\mathcal{X} = \mathcal{X}_0 \oplus \mathcal{X}_1 \oplus \cdots \oplus \mathcal{X}_N. \tag{33}$$

В квантовой теории поля конструкция (33) называется пространством Фока и используется для описания квантовых систем с переменным числом частиц. При этом переход $\mathcal{X}_L \to \mathcal{X}_{L+1}$ описывается оператором рождения, а переход $\mathcal{X}_L \to \mathcal{X}_{L-1}$ — оператором уничтожения. В (33) отдельные слагаемые прямой суммы есть регистры квантовых компьютеров длины L:

$$\mathcal{X} = \underbrace{\mathcal{X}_1 \otimes \mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_1}_{L \text{ раз}}. \tag{34}$$

Чтобы связать коэффициенты a_l и b_l волновой функции ψ_l — l-го кубита в регистре длиной L, можно воспользоваться (8), т. к. коэффициенты W_l из (8) удовлетворяют (11) и (12). С другой стороны, a_l и b_l из (24) удовлетворяют (25). Это значит, что можно положить

$$\begin{array}{ll} |a_l| & = \sqrt{W_l}\,, \\ |b_l| & = \sqrt{1 - W_l}\,. \end{array} \tag{35}$$

Из (35) следует, что

$$\begin{array}{ll} a_l & = \sqrt{W_l} e^{i\phi_l}\,, \\ b_l & = \sqrt{1 - W_l} e^{i\phi_l}\,. \end{array} \tag{36}$$

Тогда гейт — эрмитов оператор, поворачивающий вектор ψ_l, который задает состояние l-го кубита, — это просто изменение фазы ϕ_l волновой функции ϕ_l,

$$\phi_l \to \phi_l + \Phi_l, \tag{37}$$

и одновременное измерение вероятностией W_l, которые зависят от разбиения файла на неизменное число буферов L:

$$W_l = W_l(\Phi_l). \tag{38}$$

Таким образом, общая схема возможного применения квантовых алгоритмов для задачи разбиения файла на буферы выглядит следующим образом.

1. Пространство, на котором действуют эти алгоримты — это пространство Фока.

2. Гейты, действующие на отдельном прямом слагаемом в пространстве Фока — это вращения кубитов (36)–(38).

3. Гейты, действующие между отдельными прямыми компонентами в пространстве Фока — это последовательность операторов рождения и уничтожения, можно сказать, буферов при разбиении файла.

Описанная схема — это не более, чем схема, которая еще нуждается в детальной проработке.

3. Возможность использования методов распознавания образов для задачи разбиения файла на буферы

Для использования методов распознавания образов необходимо решение следующих задач.

1. Описание множества объектов, на котором нужно распознать данный.

2. Формальное описание признаков объектов, по которым строится их классификация.

3. Признаки из п. 2 разбивают множество из п. 1 на классы.

4. На множестве из п. 1 нужно задать функционал, т. е. отображение этого множества в линейно упорядоченное множество чисел.

5. Построение алгоритма, позволяющего распознать элемент из множества п. 1, удовлетворяющий признакам из п. 2, на котором функционал из п. 4 достигает экстремального (в нашем случае максимального) значения.

Пусть имеется файл длиной в N_Φ битов, который разбит на N кортежей длиной по n битов. Разбиение файла на буферы — это разложение N в сумму L членов $m_1, m_2, \ldots, m_L$ — чисел кортежей в L буферах. Последовательность $m_l,\ l = 1, 2, \ldots, L$, порождает последовательность $s_1, s_2, \ldots, s_L$ чисел разных кортежей в L буферах. А эта последовательность, с учетом того, что нам известны не только $s_l,\ l = 1, \ldots, L$, но и то, какие это кортежи, порождает последовательность $f_1, f_2, \ldots, f_L$ булевых полиномов, кодирующих l-е поля (поля принадлежности) L буферов. Разбиение файла на буферы в зависимости от L и файла будет выделять один из элементов множества всех подмножеств булевых полиномов от n переменных:

$$\begin{array}{l}\{\{0\},\ \{f_1, f_2, \ldots, f_N\},\ \{f_1, f_2, \ldots, f_{N-1}\},\ \ldots,\\ \{f_1\},\ \{f_2\},\ \ldots, \{f_N\},\ \{1\}\}.\end{array} \tag{39}$$

С точки зрения существа булева сжатия, основанного на поиске алгебраических зависимостей между полиномами $f_j,\ j = 1, 2, \ldots, 2^n$, именно

множество (39) и является тем пространством, на котором нужно найти один элемент — разбиение файла на буферы. Именно на пространстве (39) должен быть задан функционал, максимизация которого и служит признаком выбора этого элемента. Однако, для построения этого функционала надо иметь еще дополнительные данные помимо (39). А именно, для каждого элемента подмножества $\{f_1, f_2, \ldots, f_L\}$ из (39) для кодируемого файла известно:

1) $m_l,\ l = 1, 2, \ldots, L,$ — число кортежей в l-ом буфере,

2) $s_l,\ l = 1, 2, \ldots, L,$ — число разных кортежей в l-ом буфере,

3) $n_k^l,\ l = 1, 2, \ldots, L,\ k = 1, 2, \ldots, s_l,$ — число повторов k-го кортежа в l-м буфере.

Данные (39) и (40) однозначно определяются конкретным разбиением конкретного файла. Таким образом, ответ на 1-ю задачу дается (39).

Что касается ответа на 2-ю и 3-ю задачу, то, согласно (15), существуют два признака, по которым все разбиения файла делятся на 4 класса, которые удобно представить в виде следующей таблицы.

Таблица 1. Признаки, делящие разбиения файла на буферы, на классы

Решения кодирующего уравнения	$k > 0$	$k < 0$
есть	I	II
нет	III	IV

(40)

Только разбиения, попадающие в I-й класс, являются приемлемыми. Что касается 1-го признака ($k > 0$ или $k < 0$), то задавшись параметрами, характеризующими не разбиения файла, а условия его сжатия — P и I, и используя данные (39) и (40), как раз характеризующие разбиение файла, будем иметь формальное описание этого признака в виде формулы для коэффициента сжатия k,

$$k = \frac{n \sum\limits_{l=1}^{L} m_l}{2^I + P \cdot 2^n + LI \cdot \log_2 P \prod\limits_{l=1}^{L} \dfrac{C_{m_{l-1}}^{s_l - 1} m!}{\prod\limits_{k=1}^{s_l} n_l^s!}}, \tag{41}$$

которая позволяет для любого разбиения отнести его либо к 1-му столбцу табл. 1, либо ко 2-му. Что касается 2-го признака, то нетрудно получить его формальное описание, для любого разбиения конкретного файла. Булевы полиномы f_j и порождающие булевы полиномы

$$\phi_p,\ j = 0, \ldots, 2^n - 1;\ p = 1, 2, \ldots, P,$$

могут быть разложены по полиномам Лагранжа $L_j(x_k)$ или по полиномам Лагранжа $L_j(\phi_p)$ от порождающих полиномов ϕ_p:

$$\phi_p(x_i) = \sum_{j=0}^{2^n - 1} C_{pj} L_j(\phi_p), \tag{42}$$

$$f_l = \sum_{j=0}^{2^n-1} C_j^l L_j(\phi_p) = \sum_{j=0}^{2^P-1} a_i^l L_j(\phi_p), \tag{43}$$

где

$$L_i(\phi_p) = \sum_{j=0}^{2^P-1} L_{i_p}(\phi_p), \tag{44}$$

i_p — коэффициенты в двоичном представлении числа i:

$$i = \sum_{p=0}^{P} i_p 2^{p-1}. \tag{45}$$

Используя (42)–(45) нетрудно показать, что кодирующее уравнение (14) примет вид

$$\bigvee_{j=0}^{2^n-1} \bigvee_{l=1}^{L} \left\{ \sum_{i=0}^{2^P-1} \left[a_i^l \prod_{p=1}^{P} (1 + i_p + c_{pj}) + c_j^l \right] \right\} = 0 \ . \tag{46}$$

Конкретное разбиение конкретного файла одозначно определяет коэффициенты C_j^i из (43). Тогда в кодирующем уравнении (46) a_i^l и c_{pj} — это неизвестные булевы переменные. Но на уравнение (46) можно посмотреть и иначе. А именно, можно в число булевых переменных включать и C_j^l. Тогда (46) определит все возможные разбиения файлов, которые выделяют первую строку в табл. 1. Таким образом, формулы (41) и (46) дают формальное описание признаков распознаваемого объекта, т. е. решения 2-й задачи, а объединение этих признаков дает решение 3-й задачи.

Что касается 4-й задачи, то коэффициент сжатия k из (41) и служит тем самым функционалом, определенным на (39), максимум которого и надо найти. Если перенумеровать все подмножества из (39), которые согласно признакам (41) и (46) попадают в I-й класс, то их можно упорядочить в зависимости от k:

$$k_1 < k_2 < \cdots < k_\sigma. \tag{47}$$

В таком случае, как раз подмножество с номером σ и даст решение 5-й задачи.

Таким образом, все условия использования методов распознавания образов для решения задачи о разбиении файла на буферы выполнены.

Список литературы

1. *Валиев К. А., Канин А. А.* Квантовые компьютеры: надежды и реальность. – Москва–Ижевск: НТЦ “Регулярная и хаотическая динамика”, 2001. – 352 с.
2. *Китаев А., Шень А., Вялый М.* Классические и квантовые вычисления. – М.: МЦНМО, ЧеРо, 1999. – 192 с.
3. *Толстопятов А. А.* Алгоритм разбиения файла на буферы при булевом сжатии // Математика и ее приложения: Журн. Иванов. матем. об-ва. – 2008. – Вып. 1(5). – С. 77–88.

Поступила в редакцию 27.12.2009