Полурешетка корней из формальных языков специального вида

С. Ю. Корабельщикова

***Аннотация*— В данной статье представлен результат изучения структуры множества корней из языков специального вида, а именно содержащих всевозможные слова длины от t_1 до t_2 ($t_1 \le t_2$). Изучен общий случай извлечения корней n-й степени из данного языка, приведены алгоритмы нахождения всех корней из языка, первообразных корней, а также результаты работы программ, реализующих предложенные алгоритмы.**

Рассматриваемая задача сводится к задаче о рюкзаке и решается методом программной реализации предложенных алгоритмов. На множестве корней из языка задается отношение порядка, определяются минимальные и максимальные элементы. В работе показано, что множество корней из языков специального вида либо пусто, либо образует верхнюю полурешетку, минимальными элементами которой являются первообразные корни, а максимальным – тривиальный корень из данного языка.

Приведены наглядные простые примеры, которые иллюстрируют неоднозначность операции извлечения корня из языка, связь между множествами корней из двух языков с равномощными множествами индексов, свойства первообразных корней, строение полурешетки корней. Полученные результаты могут применяться для компактного описания множеств корней и аналогичных им множеств, а также для их количественных оценок.

***Ключевые слова*—формальные языки, полурешетка, корень из языка, первообразный корень.**

I. Введение

Множества текстов, объединённых общим синтаксисом, изучаются теорией формальных языков без связи с семантикой. Мощным стимулом для изучения этого объекта явились приложения математики к естественным языкам, языкам программирования и вычислительной технике. При этом важную роль играют языки некоторого специального вида, обладающие теми или иными свойствами. В теории автоматов это регулярные языки [1,2], в теории кодирования – префиксные коды, или их частный случай – равномерные коды, состоящие из различных слов одинаковой длины. Необходимость подробного изучения максимальных префиксных кодов для рассматриваемых нами задач впервые была отмечена в [3]. Впоследствии применение некоторых доказанных в [3] фактов было рассмотрено в [4]. Среди последних наших работ по этой тематике отметим [5,6].

Статья получена 6 ноября 2019.

С. Ю. Корабельщикова, Северный Арктический федеральный университет имени М. В. Ломоносова, Архангельск, Россия (e-mail: s.korabelsschikova@ narfu.ru).

В данной статье мы рассматриваем языки, содержащие все слова заданной длины, а также объединения языков такого вида. Она расширяет круг вопросов, затронутых в [7, 8] и посвящена изучению строения множества корней из языка, содержащего все слова длины от t_1 до t_2. Для этого на множестве корней из языка задается отношение порядка, определяются минимальные и максимальные элементы. Доказано, что множество корней n-й степени из языка либо пусто, либо образует верхнюю полурешетку.

Приведем необходимые определения и обозначения согласно [9]. Пусть Σ – *алфавит*, некоторое непустое множество букв. *Словом* над заданным алфавитом будем называть конечную последовательность букв этого алфавита, а *длиной слова* - количество букв в нём. *Язык* - любое множество слов некоторого фиксированного алфавита. Конкатенацию нескольких одинаковых слов (языков) будем называть степенью слова (языка). Таким образом, Σ^i – множество всех слов длины i в алфавите Σ, и $\Sigma^*=\bigcup_{i=1}^{\infty}\Sigma^i$ есть множество всевозможных непустых слов в алфавите Σ, причем для любых $u\in\Sigma^*$ и $k, l\in N$ выполнено равенство $u^k\cdot u^l=u^{k+l}$.

В общем виде задача извлечения корня n–й степени формулируется следующим образом: для заданного языка $A\subseteq \Sigma^*$ и заданного $n\in N$ требуется найти все языки B такие, что $A = B^n$. Эта теоретическая задача тесно связана с практической задачей декодирования сообщений, где требуется найти разбиение кодового сообщения на элементарные коды, соответствующие символам алфавита источника. Ранее в работах [5,6] мы рассматривали подходы к решению этой проблемы для языков специального вида, состоящих из всех слов в алфавите Σ длины от t_1 до t_2. Приведем простой пример.

Пример 1. Пусть $A = \bigcup_{i=2}^{10}\Sigma^i$, то есть t_1=2, t_2=10.

Из языка A извлекается *тривиальный* квадратный корень $B_1 = \bigcup_{i=1}^{5}\Sigma^i$.

Однако язык $B_2 = \bigcup_{i\epsilon\{1,2,4,5\}}\Sigma^i$ также является корнем из языка A. Действительно, так как любое число t, $2 \le t \le 10$, можно представить в виде суммы двух (не обязательно различных) слагаемых a и b из множества $\{1, 2, 4, 5\}$, то и любое слово длины t можно получить конкатенацией двух слов длин a и b соответственно.

Полученные ранее результаты приводятся здесь без доказательства. Также в данной статье представлен новый результат (теорема 3) – описание структуры множества корней из языка. Языки, как уже было сказано, являются множествами слов, поэтому к ним применимы любые теоретико-множественные операции, в частности объединение и пересечение, которые определяют точную верхнюю грань и точную нижнюю грань двух множеств.

II. Квадратные корни из языков специального вида

Далее в работе будем рассматривать только формальные языки вида Σ^i, содержащие всевозможные слова заданной длины i над алфавитом Σ, либо объединения языков такого вида. Рассмотрим вопрос извлечения квадратного корня из языка A:

$$A = \bigcup_{i=t_1}^{t_2} \Sigma^i, \text{ где } t_1, t_2 \in N, t_1 \le t_2. \quad (1)$$

Приведем примеры, демонстрирующие соответствие между множествами корней при условии, что $t_2 - t_1 = const$.

Пример 2. Из языка $A = \bigcup_{i=2}^{12} \Sigma^i$ извлекаются три квадратных корня вида $B = \bigcup_{i \in M} \Sigma^i$, где $M = \{1, 2, 3, 4, 5, 6\}$, $M = \{1, 2, 4, 5, 6\}$ и $M = \{1, 2, 3, 5, 6\}$.

Пример 3. Из языка $A = \bigcup_{i=4}^{14} \Sigma^i$ извлекаются также 3 квадратных корня, и они получаются сдвигом ранее найденных в примере 2 множеств индексов корней на 1 вправо. Это корни: вида $B = \bigcup_{i \in L} \Sigma^i$, где $L = \{2, 3, 4, 5, 6, 7\}$, $L = \{2, 3, 5, 6, 7\}$ и $L = \{2, 3, 4, 6, 7\}$.

Из рассмотренных примеров видно, что по крайней мере один (тривиальный) квадратный корень из языка $A = \bigcup_{i=t_1}^{t_2} \Sigma^i$ существует тогда и только тогда, когда t_1 и t_2 чётны. Также мы показали, что задачу нахождения корней из языка $A = \bigcup_{i=t_1}^{t_2} \Sigma^i$ можно свести к задаче нахождения корней из языка $A = \bigcup_{i=2}^{t_2 - t_1 + 2} \Sigma^i$. Справедлива следующая теорема:

Теорема 1. Пусть $t_1 = 2n_1$ и $t_2 = 2n_2$ ($n_1, n_2 \in N$, $n_1 \le n_2$). Тогда из языка $A = \bigcup_{i=t_1}^{t_2} \Sigma^i$существует по крайней мере один квадратный корень $B = \bigcup_{i \in M} \Sigma^i$, где $M = \{n_1, \ldots, n_2\}$. Имеется взаимно однозначное соответствие между квадратными корнями из языка $\bigcup_{i=t_1}^{t_2} \Sigma^i$ и квадратными корнями из языка $\bigcup_{i=2}^{t_2 - t_1 + 2} \Sigma^i$. Если $\bigcup_{i \in M} \Sigma^i$ является квадратным корнем из языка $\bigcup_{i=2}^{t_2 - t_1 + 2} \Sigma^i$, то множество $L = M + (n_1 - 1)$ определяет квадратный корень $\bigcup_{i \in L} \Sigma^i$ из языка $\bigcup_{i=t_1}^{t_2} \Sigma^i$, и наоборот.

III. Корни n–й степени из языков специального вида

Теперь рассмотрим вопрос нахождения корней n–й степени из языка вида (1). Отметим вполне очевидный факт: для того, чтобы корень n–й степени из языка вида (1) извлекался, необходимо и достаточно, чтобы t_1 и t_2 делились на n. Обобщим проведённые выше рассуждения. Пусть задан язык A вида:

$$A = \bigcup_{i=n \cdot n_1}^{n \cdot n_2} \Sigma^i, \text{ где } n_1, n_2 \in N, n_1 \le n_2. \quad (2)$$

Теорема 2. Пусть Σ – произвольный алфавит, M – подмножество множества натуральных чисел $\{n_1, n_1 + 1, \ldots, n_2\}$. Язык $B = \bigcup_{i \in M} \Sigma^i$ является корнем n–й степени из языка вида (2) тогда и только тогда, когда выполняется условие:

$$(\forall i)\ (n \cdot n_1 \le i \le n \cdot n_2 \rightarrow i = a_1 + a_2 + \ldots + a_n), \quad (3)$$

где $a_1, a_2, \ldots, a_n$ – некоторые элементы из M (не обязательно различные). При этом между корнями n–й степени из языка $\bigcup_{i=n \cdot n_1}^{n \cdot n_2} \Sigma^i$ и корнями n–й степени из языка $\bigcup_{i=n_1}^{n \cdot n_2 - n \cdot n_1 + n} \Sigma^i$ имеется взаимно однозначное соответствие. Множеству индексов M корня из языка $\bigcup_{i=n_1}^{n \cdot n_2 - n \cdot n_1 + n} \Sigma^i$ соответствует множество индексов $L = M + (n_1 - 1)$ корня из языка $\bigcup_{i=n \cdot n_1}^{n \cdot n_2} \Sigma^i$, и наоборот.

Как уже было отмечено выше, языки вида (1), у которых t_1 и t_2 не делятся на n, не имеют корней. Как следует из теоремы 2, число корней n–й степени зависит от n и от мощности множества $\{n_1, n_1 + 1, \ldots, n_2\}$, которую обозначим k ($k = n_2 - n_1 + 1$). Рассмотренная задача свелась, по сути, к задаче представления натуральных чисел из заданного интервала $[t_1, t_2]$ в виде суммы n натуральных слагаемых из интервала $[t_1 / n, t_2 / n]$, что эквивалентно задаче о рюкзаке.

Алгоритм №1. Приведём алгоритм нахождения корней n–й степени из языка вида (2). Входные данные: t_1, t_2, n, причём $t_1 \le t_2$.

Шаг 1: проверка делимости чисел t_1 и t_2 на n. Если не выполняется – корней нет. Иначе $n_1 = t_1 / n$, $n_2 = t_2 / n$ и переходим к шагу 2.

Шаг 2: генерация всех подмножеств множества $\{n_1, n_1 + 1, \ldots, n_2\}$. Этот шаг можно немного оптимизировать. Если подмножество множества $\{n_1, n_1 + 1, \ldots, n_2\}$ не содержит элементы n_1, или $n_1 + 1$, или $n_2 - 1$, или n_2, то оно заведомо не удовлетворяет условию (3). Поэтому можно сгенерировать только подмножества множества $\{n_1, n_1 + 1, \ldots, n_2\}$, содержащие эти элементы.

Шаг 3: для каждого полученного в шаге 2 множества проводим проверку на выполнение условия (3). Множества, прошедшие проверку, включаем в ответ, подсчитываем их количество.

Асимптотическая сложность вычислений полученной реализации равна $O(2^{(k-4)} \cdot check\_time(n, k))$, где $check\_time(n, k)$ – асимптотическая сложность проверки конкретного подмножества.

Так как задача проверки эквивалентна задаче о рюкзаке, то опишем классическое решение этой задачи, находящее точное решение за полиномиальное время, а именно, решение с помощью динамического программирования.

Алгоритм №2. Введем бинарную величину $D_{i,j}$, где $D_{i,j} = 1$, если можно набрать вес i, использовав при этом j любых предметов. Если же нельзя набрать, то $D_{i,j} = 0$. Допустим, что сейчас мы находимся в допустимом состоянии $D_{i,j} = 1$. Попытаемся добавить предмет с номером k в рюкзак из текущего состояния $D_{i,j}$. Обозначим за $q = w_k$ – вес k предмета. Так как текущее состояние $D_{i,j} = 1$, то, при добавлении предмета с номером k, мы перейдем в состояние $D_{i+q,j+1}$ и оно тоже будет равно 1. Начальное состояние бинарной величины D будет $D_{0,0} = 1$, так как всегда можно набрать нулевой вес, не использовав ни одного предмета (грубо говоря, ничего не взять мы можем всегда). Ответом же на задачу будет $R = max\{\ i \mid i \le W \wedge D_{i,M} = 1\ \}$ – эта величина показывает, какой максимальный вес мы сможем набрать, использовав ровно M предметов.

Чтобы посчитать такую бинарную величину, нужно использовать следующий алгоритм:

1. Перебирать предметы в порядке увеличения количества взятых предметов в рюкзак. Обозначим за

j ($0 \le j \le M-1$) – текущее количество взятых предметов.

2. Перебирать в любом порядке номера предметов, добавляемых в рюкзак.

3. Ввести обозначения.

Обозначим за k ($1 \le k \le N$) – текущий номер предмета, который мы будем пытаться добавить в рюкзак.

Обозначим за $q = w_k$ – вес предмета под номером k.

Обозначим за i ($0 \le i \le W$) – текущий рассматриваемый вес.

4. Если $D_{i,j} = 1$, то помечать $D_{i+q,j+1}$ состояние тоже единицей.

Асимптотическая сложность данного алгоритма $O(M \cdot N \cdot W)$.

Замечание (о программной реализации данного алгоритма). Хранить веса предметов можно в виде бинарной строки, используя встроенные числовые типы размера 32 или 64 бита. Это позволит ускорить скорость работы алгоритма в 32^2 или 64^2 раза соответственно. Асимптотическая сложность такой реализации $O(M \cdot (N / 64) \cdot (W / 64))$.

Используя алгоритм 2 для проверки множества в алгоритме 1, получим итоговую асимптотическую сложность алгоритма $O(2^{(k-4)} \cdot n^2 \cdot k^2)$, что позволяет считать количество корни n–й степени вида $\bigcup_{i\epsilon M} \Sigma^i$, если мощность множества M не превосходит 30.

В таблице 1 приведены результаты работы программы, то есть количество корней n–й степени из языков вида (2) для некоторых n и k ($k = n_2 - n_1 + 1$). При k от 1 до 3 значения равны 1.

Таблица 1 – Количество корней n–й степени

$n \backslash k$	4	5	6	7	8	9	10	11	12	13	14
2	1	2	3	5	9	15	28	50	95	174	337
3	1	2	4	7	13	25	49	95	185	365	721
4	1	2	4	8	15	29	57	113	225	447	889
5	1	2	4	8	16	31	61	121	241	481	961
6	1	2	4	8	16	32	63	125	249	497	993
7	1	2	4	8	16	32	64	127	253	505	1009

IV. Первообразные корни из языков специального вида

Определение 1. Пусть Σ – произвольный алфавит, M – подмножество множества натуральных чисел $\{n_1, n_1 + 1, \ldots, n_2\}$ и язык $B = \bigcup_{i\epsilon M} \Sigma^i$ является корнем n–й степени из языка вида (2). Пусть S – семейство всех таких множеств индексов M корней n–й степени из языка вида (2). Корень $B = \bigcup_{i\epsilon M} \Sigma^i$ называется первообразным, если M – минимальное по включению множество из S.

Определение 2. Мощность множества индексов называется весом корня.

Пример 4. Из языка $A = \bigcup_{i=2}^{14} \Sigma^i$ извлекаются 5 квадратных корней с множествами индексов $\{1, 2, 3, 4, 5, 6, 7\}$, $\{1, 2, 3, 4, 6, 7\}$, $\{1, 2, 4, 5, 6, 7\}$, $\{1, 2, 4, 6, 7\}$ и $\{1, 2, 3, 5, 6, 7\}$. Из них 2 последних множества индексов – минимальные, им соответствуют 2 первообразных корня веса 5 и 6 соответственно.

Отметим вполне очевидные свойства первообразных корней:

1. Если $M \in S$ и M – множество индексов минимальной мощности, то соответствующий M корень – первообразный.

2. Обратное утверждение в общем случае неверно. Первообразный корень не обязан иметь минимальный вес (см. пример 4).

3. Если вес первообразного корня $w = k$ ($k = n_2 - n_1 + 1$ и обозначает максимально возможный вес), то он единственен, и никаких других корней из данного языка нет.

4. Пусть $\bigcup_{i\epsilon M} \Sigma^i$ – первообразный корень n–й степени из языка $\bigcup_{i=n\cdot n_1}^{n\cdot n_2} \Sigma^i$ с множеством индексов M, и $(\alpha_1, \alpha_2, \ldots, \alpha_k)$ – двоичный характеристический вектор подмножества M множества $\{n_1, n_1 + 1, \ldots, n_2\}$. Тогда симметричный вектор $(\alpha_k, \alpha_{k-1}, \ldots, \alpha_1)$ также задаёт множество индексов первообразного корня.

5. Если первообразный корень из языка единственен, то характеристический вектор его множества индексов симметричен.

6. Пусть $\bigcup_{i\epsilon M} \Sigma^i$ – первообразный корень n–й степени из языка $\bigcup_{i=n\cdot n_1}^{n\cdot n_2} \Sigma^i$ с множеством индексов M. Тогда для любого множества M' такого, что $M \subseteq M' \subseteq \{n_1, n_1 + 1, \ldots, n_2\}$, $\bigcup_{i\epsilon M'} \Sigma^i$ – корень n–й степени из этого же языка.

7. Обобщением свойства 3 является следующее свойство: если вес первообразного корня $w < k$, и он единственен, то общее число корней вида $\bigcup_{i\epsilon M} \Sigma^i$ равно 2^{k-w}.

Алгоритм №3. Приведём алгоритм нахождения первообразных корней n–й степени из языка вида (2). Входные данные: t_1, t_2, n, причём $t_1 \le t_2$.

Шаг 1: проверка делимости чисел t_1 и t_2 на n. Если не выполняется – первообразных корней нет. Иначе $n_1 = t_1 / n$, $n_2 = t_2 / n$ и переходим к шагу 2.

В алгоритме предполагается, что мы сохраняем «потенциальные» первообразные корни, то есть минимальные по включению корни из уже рассмотренных подмножеств. Обозначим множество «потенциальных» первообразных корней за E.

Шаг 2: берем любое не рассмотренное подмножество Y множества $\{n_1, n_1 + 1, \ldots, n_2\}$ и переходим к шагу 3.

Шаг 3: проверяем, является ли Y корнем (данную проверку можно делать так же, как и в алгоритме 1, шаг 3). Если да, то переходим к шагу 4, иначе к шагу 6.

Шаг 4: проверяем, нельзя ли получить Y из какого-нибудь корня из E. Если можно, то Y точно не является первообразным корнем, пропускаем его и переходим к шагу 6, иначе считаем его новым «потенциальным» первообразным корнем и переходим к шагу 5.

Шаг 5: проверяем, нельзя ли получить какие-то корни в E из корня Y. Если такие корни находятся, то удаляем их из множества E. После этого добавляем Y в E и переходим к шагу 6.

Шаг 6: если еще имеются не рассмотренные подмножества, то переходим к шагу 2, иначе конец алгоритма.

Асимптотическая сложность вычислений полученной реализации равна $O(2^{(k-4)} \cdot n^2 \cdot k^2)$, что позволяет считать количество первообразных корней n–й степени вида $\bigcup_{i\epsilon M} \Sigma^i$, если мощность множества M не превосходит 30.

В таблице 2 приведены результаты работы

программы, то есть количество первообразных корней n–й степени из языков вида (2) для некоторых n и k ($k = n_2 - n_1 + 1$). При k от 1 до 5 значения равны 1.

Таблица 2 – Количество первообразных корней n–й степени

n \ k	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
2	2	2	4	6	9	15	20	41	50	87	126
3	1	3	3	5	9	11	22	33	45	82	120
4	1	1	4	4	6	10	16	18	29	48	77
5	1	1	1	5	5	7	11	17	25	27	38
6	1	1	1	1	6	6	8	12	18	26	36
7	1	1	1	1	1	7	7	9	13	19	27

V. Полурешетка корней n–й степени из языков специального вида

Все необходимые определения можно найти в [10]. Пусть задан язык $A = \bigcup_{i=n\cdot n_1}^{n\cdot n_2} \Sigma^i$. Пусть $B=\{B_1, B_2, \ldots, B_r\}$ – множество всех корней n–й степени из языка A, а $S=\{M_1, M_2, \ldots, M_r\}$ – семейство соответствующих корням множеств индексов. Очевидно, что множеству S принадлежит множество $M = \{n_1, n_1 + 1, \ldots, n_2\}$, которое является наибольшим элементом в S и соответствует тривиальному корню из языка A. Как следует из определения первообразного корня, множества индексов первообразных корней являются минимальными элементами в S. Кроме того, объединение любых двух множеств из S принадлежит S, что нельзя сказать о пересечении. Справедлива следующая теорема.

Теорема 3. Пусть $B=\{B_1, B_2, \ldots, B_r\}$ – множество всех корней n–й степени из языка A, а $S=\{M_1, M_2, \ldots, M_r\}$ – семейство соответствующих корням множеств индексов. Множество S с отношением включения является верхней полурешеткой с одним максимальным и несколькими минимальными элементами, число которых равно числу первообразных корней. Полурешетке S соответствует такая же полурешетка B корней n-й степени из языка A.

Покажем полурешетку, соответствующую множеству корней из примера 4. $S=\{M_1, M_2, M_3, M_4, M_5\}$, где
$M_1=\{1, 2, 3, 4, 5, 6, 7\}$ - максимальный элемент;
$M_2=\{1, 2, 3, 4, 6, 7\}$, $M_3=\{1, 2, 4, 5, 6, 7\}$;
$M_4=\{1, 2, 4, 6, 7\}$, $M_5=\{1, 2, 3, 5, 6, 7\}$ - минимальные элементы.

Распределим их на уровни по убыванию веса и установим отношения включения.

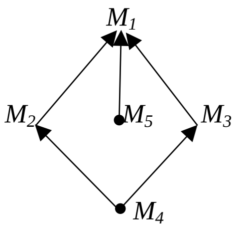

Рисунок 1. Полурешетка квадратных корней из языка $A = \bigcup_{i=2}^{14} \Sigma^i$ (пример 4).

Пример 5. Найдем все корни третьей степени из языка $A = \bigcup_{i=27}^{42} \Sigma^i$.

Множество индексов корня является подмножеством в $M_1=\{9, 10, 11, 12, 13, 14\}$, то есть $k = 6$. При $n = 3$, $k = 6$ первообразный корень один (см. таблицу 2), и он имеет характеристический вектор (110011), то есть минимальное множество индексов $M_2 = \{9, 10, 13, 14\}$. Поэтому по свойству 6 первообразных корней получим еще 2 корня с множествами индексов: $M_3=\{9, 10, 11, 13, 14\}$ и $M_4=\{9, 10, 12, 13, 14\}$. Это совпадает с результатами, приведенными в таблице 1.

Изобразим полурешетку $S=\{M_1, M_2, M_3, M_4\}$.

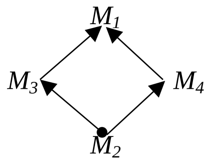

Рисунок 2. Полурешетка корней третьей степени из языка $A = \bigcup_{i=27}^{42} \Sigma^i$.

Учитывая установленное в теореме 2 соответствие, аналогичную структуру будут иметь корни третьей степени из языков $A = \bigcup_{i=3}^{18} \Sigma^i$, $A = \bigcup_{i=6}^{21} \Sigma^i$ и т. д.

VI. Заключение

В данной работе приведено решение задачи извлечения корней n-й степени из языка, содержащего всевозможные слова длины от t_1 до t_2. Эта задача сводится к задаче о рюкзаке и может быть решена для допустимых значений методом программной реализации предложенных автором алгоритмов. Кроме того, нами подробно изучена структура множества решений данной задачи, введено понятие первообразного корня из языка, выяснены свойства первообразных корней.

Предложенный автором алгоритм 3 эффективно находит первообразные корни при условии $k \leq 30$. Количественные оценки приведены в таблицах 1 и 2, они подтверждаются рассмотренными примерами.

В работе доказано, что множество корней из языка специального вида образует верхнюю полурешетку, для которой первообразные корни являются минимальными элементами, а тривиальный корень – максимальным элементом. Также множество первообразных корней n–й степени может использоваться для компактного описания всех корней, так как, благодаря свойству 6, позволяет найти все корни из языка рассматриваемого вида.

Библиография

[1] Мельников Б.Ф., Вылиток А.А., Мельникова Е.А. Итерации языков и конечные автоматы. International Journal of Open Information Technologies, 2017, Т. 5, № 12, С. 1–7.

[2] Melnikov B. Extended nondeterministic finite automata Fundamenta Informaticae, 2010, T. 104, № 3, C. 255–265.

[3] Melnikov B. The equality condition for infinite catenations of two sets of finite words. International Journal of Foundations of Computer Science, 1993, T. 4, № 3, C. 267–273.

[4] Melnikov B., Kashlakova E. Some grammatical structures of programming languages as simple bracketed languages. Informatica, 2000, T. 11, № 4, C. 441–446.

[5] Корабельщикова С.Ю., Мельников Б.Ф. Итерации языков и максимальные префиксные коды. Вестник Воронежского государственного университета. Серия: Физика. Математика, 2015, № 2, С. 106–120.

[6] Корабельщикова С.Ю., Мельников Б.Ф. Максимальные префиксные коды и подклассы класса контекстно-свободных

языков. Вестник Северного (Арктического) федерального университета. Серия: Естественные науки, 2015, № 1, С. 121–129.
[7] Melnikov B.F., Korabelshchikova S.Yu., Dolgov V.N. On the task of extracting the root from the language. International Journal of Open Information Technologies, 2019, Т. 7, № 3, pp. 1–6.
[8] Корабельщикова С.Ю., Чесноков А.И., Тутыгин А.Г. О первообразных корнях из языков специального вида. Сборник трудов IX международной конференции «Дискретные модели в теории управляющих систем», 2015, С. 116–118.
[9] Кревский И. Г., Селивёрстов М. Н., Григорьева К. В. Формальные языки, грамматики и основы построения трансляторов: Учебное пособие / Под ред. А. М. Бершадского. – Пенза: Изд-во Пенз. гос. ун-та, 2002, 124 с.
[10] Биркгоф Г. Теория решеток. – М.: Наука, 1984, 566 с.

Svetlana Yurievna KORABELSHCHIKOVA,
Associated Professor (Docent) of Northern (Arctic) Federal University named after M. V. Lomonosov, Russia
https://narfu.ru/en/university/,
email: s.korabelsschikova@narfu.ru,
elibrary.ru: authorid=711810.

Semilattice of roots from formal languages of a special kind

S. Y. Korabelshchikova

***Abstract*—This article presents the result of studying the structure of a set of roots from languages of a special type, namely containing all possible words of length from t_1 to t_2 ($t_1 \leq t_2$). The General case of extraction of n-th degree roots from a given language is studied; algorithms for finding all roots from the language, primitive roots, as well as the results of the programs that implement the proposed algorithms are given.**

The problem under consideration is reduced to the problem of the backpack and is solved by the method of software implementation of the proposed algorithms. On the set of roots from the language, the order relation is set, the minimum and maximum elements are defined. The paper shows that the set of roots from languages of a special type is either empty or forms an upper semilattice, the minimum elements of which are the primitive roots, and the maximum element is the trivial root from the given language.

There are clear simple examples that illustrate the ambiguity of the operation of extracting a root from a language, the relationship between sets of roots from two languages with equally cardinal number of index sets, the properties of primitive roots, and the structure of the semilattice of roots. The results obtained can be used for a compact description of sets of roots and similar sets, as well as for their quantitative estimates.

***Key words*-formal language, semilattice, root from language, primitive root.**

REFERENCES

[1] Melnikov B., Vylitok A., Melnikova E. Iterations of languages and finite automata. *International Journal of Open Information Technologies*, 2017, vol. 5, № 12, pp. 1–7.

[2] Melnikov B. Extended nondeterministic finite automata. *Fundamental Informatics*, 2010, vol. 104, № 3, pp. 255–265.

[3] Melnikov B. The equality condition for infinite catenations of two sets of finite words. *International Journal of Foundations of Computer Science*, 1993, vol. 4, № 3, pp. 267–273.

[4] Melnikov B., Kashlakova E. Some grammatical structures of programming languages as simple bracketed languages. *Informatics*, 2000, vol. 11, № 4, pp. 441–446.

[5] Korabel'shchikova S., Melnikov B. Iterations of languages and maximal prefix codes. *Bulletin of the Voronezh State University. Series: Physics. Mathematics*, 2015, № 2, pp. 106–120.

[6] Korabel'shchikova S., Melnikov B. Maximal prefix codes and subclasses of the context-free languages class. *Bulletin of the Northern (Arctic) Federal University. Series: Natural sciences*, 2015, № 1, pp. 121–129.

[7] Melnikov B.F., Korabelshchikova S.Yu., Dolgov V.N. On the task of extracting the root from the language. *International Journal of Open Information Technologies*, 2019, vol. 7, № 3, pp. 1–6.

[8] Korabelschikova S.Yu., Chesnokov A.I., Tutygin A.G. About primitive roots from languages of a special kind. *Proceedings of the IX international conference* "Discrete models in the theory of control systems", 2015, pp. 116–118.

[9] Krevsky I. G., Seliverstov M. N., Grigorieva K. V. Formal languages, grammars and bases of construction of translators: a Textbook / ed. by A. M. Bershadsky. *Penza: Publishing house of Penza State University*, 2002, 124 p.

[10] Birkhoff G. Theory of lattices. *Moscow: Publishing house Science*, 1984, 566 p.