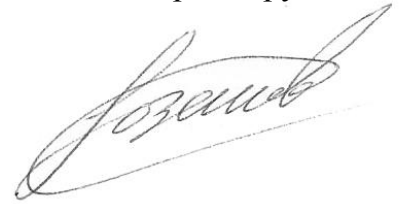


На правах рукописи



РОЗАНОВ Алексей Константинович

МАТЕМАТИЧЕСКОЕ, АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ  
ОБЕСПЕЧЕНИЕ АВТОМАТИЧЕСКОГО ПРЕДСИНТАКСИЧЕСКОГО АНАЛИЗА  
ТЕКСТА В СИСТЕМАХ УПРАВЛЕНИЯ БАЗАМИ ЛИНГВИСТИЧЕСКИХ  
ЗНАНИЙ

Специальность 05.13.11 – Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ  
диссертации на соискание ученой степени  
кандидата технических наук

Рязань 2017

Работа выполнена на кафедре «Вычислительная и прикладная математика» Федерального государственного бюджетного образовательного учреждения высшего образования «Рязанский государственный радиотехнический университет» (ФГБОУ ВО «РГРТУ»).

Научный руководитель:  
Пруцков Александр Викторович,  
доктор технических наук, доцент, профессор кафедры «Вычислительная и прикладная математика» Федерального государственного бюджетного образовательного учреждения высшего образования «Рязанский государственный радиотехнический университет».

Официальные оппоненты:

Ломакина Любовь Сергеевна,  
доктор технических наук, профессор, профессор кафедры «Вычислительные системы и технологии» Федерального государственного бюджетного образовательного учреждения высшего образования «Нижегородский государственный технический университет им. Р.Е. Алексеева», г. Нижний Новгород.

Поляков Дмитрий Вадимович,  
кандидат технических наук, старший преподаватель кафедры «Информационные системы и защита информации» Федерального государственного бюджетного образовательного учреждения высшего образования «Тамбовский государственный технический университет», г. Тамбов.

Ведущая организация: Федеральное государственное бюджетное образовательное учреждение высшего образования «Пензенский государственный технологический университет», г. Пенза.

Защита диссертации состоится «19» апреля 2017 г. в 11 часов 30 минут на заседании диссертационного совета Д 212.211.01 в ФГБОУ ВО «РГРТУ» по адресу: **390005, г. Рязань, ул. Гагарина, д. 59/1.**

С диссертацией можно ознакомиться в библиотеке и на сайте ФГБОУ ВО «Рязанский государственный радиотехнический университет» <http://www.rsreu.ru>.

Автореферат разослан «\_\_» \_\_\_\_\_ 2017 г.

Ученый секретарь  
диссертационного совета,  
канд. техн. наук, доцент



Виктор Николаевич Пржегорлинский

**Актуальность темы исследования.** В настоящее время информация является одним из наиболее ценных ресурсов в мире. Объёмы информации, порождаемой, передаваемой и, как следствие, нуждающейся в оперативной обработке и, что немаловажно, оперативном восприятии, непрерывно растут. Вследствие этого всё большую ценность приобретают методы автоматизации обработки информации.

Текст является одной из важнейших форм представления информации, поэтому в свете постоянного увеличения интенсивности информационных потоков всё более важную роль играет *автоматическая обработка текста*.

Обработка текста на естественном языке – это процесс, включающий в себя несколько стадий, соответствующих уровням обработки текста: морфологический (выделение и анализ отдельных слов текста), синтаксический (определение структур предложений), семантический (выявление смысла) и прагматический (определение целей говорящего).

Предсинтаксический анализ текста на естественном языке, включающий в себя в общем случае этап разбиения текста на слова и этап определения форм слов, является необходимым в любом процессе, включающем обработку текста на естественном языке, поэтому тема диссертации, посвященная повышению скорости определения форм слов в текстах на естественных языках, является *актуальной*.

**Степень разработанности темы.** Существенный вклад в развитие методов автоматической обработки текстов на естественных языках внесли отечественные учёные Г.Г. Белоногов, Э.В. Попов, Д.А. Поспелов, Ю.Д. Апресян, М.Г. Мальковский, И.В. Сегалович, В.М. Брябин, О.С. Кулагина, Ю.Н. Марчук, И.А. Мельчук, А.С. Нариньяни, В.А. Фомичев и другие, а также зарубежные специалисты Т. Виноград (Т. Winograd), В.А. Вудс (W.A. Woods), К. Коскенниemi (K. Koskenniemi), М. Портер (M. Porter), Н. Хомский (N. Chomsky), Д. Джурафски (D. Jurafsky), Дж. Мартин (J.H. Martin) и другие.

Первым этапом обработки текста является этап определения форм слов. Алгоритм определения форм слов – это правило, ставящее в соответствие каждому из слов анализируемого текста специальный маркер, описывающий грамматическую информацию, присущую этому слову (например, «столами» – «неодушевлённое существительное во множественном числе, в творительном падеже»).

Существует целый ряд алгоритмов определения форм слов, однако каждому из них присущи некоторые недостатки (невысокая скорость определения форм слов, ориентированность на конкретный язык, невозможность обратного процесса – генерации форм слов с заданной грамматической информацией).

Устранить перечисленные недостатки позволяет метод генерации и определения форм слов, который является универсальным, и допускает как анализ (определение), так и синтез (генерацию) форм слов. Однако универсальность достигается за счет низкой скорости анализа.

**Соответствие паспорту специальности.** Диссертация соответствует пункту 4 «Системы управления базами данных и знаний» специальности 05.13.11 – «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей», поскольку в работе разработаны программные средства для ЭВМ, включающие систему управления базами знаний о формообразовании естественных языков, и метод решения задачи определения и генерации форм слов на основе этих знаний.

**Цель и задачи исследования.** Целью диссертационной работы является разработка методов повышения скорости определения форм слов естественных языков и усовершенствование способа представления знаний о формообразовании естественных языков.

Для достижения цели диссертационного исследования необходимо решить следующие задачи:

- разработать универсальную языконезависимую модель представления правил формообразования и алгоритмы на ее основе, применение которых повысит скорость определения форм слов;
- разработать формальное описание структуры словаря системы генерации и определения форм слов, использующей предложенные алгоритмы, что позволило бы сделать процесс заполнения словаря как можно менее трудоёмким;
- разработать информационную систему генерации и определения форм слов, реализующую разработанные алгоритмы.

**Научная новизна.** Научная новизна выполненных исследований состоит в следующем:

1) предложены модель представления правил формообразования естественных языков, метод повышения скорости определения форм слов на её основе, использующий особенности постфиксного базиса элементарных операций и обеспечивающий полуторакратный прирост скорости определения по сравнению с существующими аналогами и метод повышения скорости анализа на основе построения полного банка словоформ (обеспечивающий ещё большую скорость определения за счёт увеличенных затрат памяти), приведены рекомендации по их применению;

2) разработана формальная грамматика, описывающая структуру словаря для хранения знаний о формообразовании, необходимого для повышения скорости определения форм слов;

3) получено представление знаний о формообразовании русского языка в терминах предложенной модели, отличающееся высоким уровнем структуризации знаний и снижающее трудозатраты при пополнении словаря (система требует ввода только словоформ, соответствующих разрешённым комбинациям грамматических значений, исключая заведомо запрещённые, например, падежные формы неизменяемых существительных, причастия совершенного вида настоящего времени, сравнительная степень относительных прилагательных; усреднённая доля запрещённых комбинаций в русском языке для разных частей речи колеблется от 1–5% (доля неизменяемых существительных) до 90–95% (глаголы));

4) разработан метод морфологического представления текста, позволяющий сохранить результаты анализа текста и обеспечивающий сжатие текста до 30–40% от исходного.

**Теоретическая и практическая значимость работы.** Ценность проведенной работы состоит в том, что в её результате была построена универсальная иерархическая модель представления знаний о формообразовании естественных языков, включающая в себя и разбиение хранимых слов языка на их классы, и ассоциации классов с правилами формообразования, и гибкую систему описания парадигм для отдельных классов слов.

Для языков, характеризующихся постфиксным формообразованием, предложены алгоритмы анализа форм слов, обеспечивающие более высокую по сравнению с существующими аналогами скорость определения форм слов.

Результаты, полученные в диссертационном исследовании, являются развитием научного направления разработки и исследования универсальных методов генерации и определения форм слов.

В результате работы был создан программный комплекс, использующий предложенную модель организации знаний о формообразовании языка для решения задачи определения и генерации форм слов, включающий в себя редактор словарей, средство анализа текстов и систему проверки знаний формообразования.

**Объект исследования.** Объектом исследования является система правил формообразования естественного языка, исследуемая с целью построения её формальной модели для системы анализа форм слов естественных языков.

**Предмет исследования.** Предметом исследования являются математические модели правил формообразования естественных языков и алгоритмы, решающие в рамках этих моделей задачи генерации и определения форм слов естественных языков.

**Методология и методы исследования.** Теоретико-методологической основой исследования являются труды отечественных и зарубежных авторов, посвящённые проблемам анализа текстов на естественных языках.

К числу применённых в работе общенаучных методов относятся метод формализации, метод моделирования, системный подход.

При решении задач диссертационного исследования нашли применение теория алгоритмов и структур данных, а также элементы теории алгебр, графов, формальных грамматик, алгоритмов и морфологических категорий в лингвистике.

**Положения, выносимые на защиту:**

- модель представления правил формообразования естественных языков, алгоритм определения форм слов на её основе, использующий встречные префиксные деревья для представления правил формообразования, алгоритм определения форм слов, основанный на подходе «определение через генерацию»;

- формальное описание структуры словаря в системе генерации и определения форм слов (на языке описания формальных грамматик);

- представление знаний о формообразовании русского языка в терминах предложенной модели;

- алгоритм кодирования проанализированных текстов, обеспечивающий их морфологическое сжатие.

**Решение поставленных задач.** Решение задач проведено по следующей схеме. Рассматривается одна из существующих моделей представления правил формообразования (цепочки элементарных преобразований), анализируются возможные подходы к ускорению алгоритма анализа при определённых условиях (в частности, для постфиксного базиса элементарных операций), доказываемое существование верхнего предела числа операций в цепочке, что даёт возможность более компактного представления правил преобразования. Это, в свою очередь, приводит к построению более эффективных алгоритмов анализа (для языков с постфиксным формообразованием).

Поскольку любая система анализа форм слов естественного языка, использующая словарь (слов в начальных формах или основ), обязательно требует структуризации этого словаря, в работе также решается задача создания и формального описания структуры словаря системы, использующей предложенные алгоритмы для анализа форм слов, которая позволила бы сделать процесс заполнения словаря как можно менее трудоёмким.

Для проверки практической применимости разработанных моделей и алгоритмов в рамках работы создан программный комплекс *Salvinia*, предназначенный для решения задач определения и генерации форм слов, и набран тестовый словарь, содержащий более 8000 начальных форм слов (гистограмма, описывающая структуру тестового словаря, приведена в автореферате).

С помощью созданного программного комплекса выполнен ряд контрольных замеров скорости определения форм слов на больших наборах слов с целью сравнения её со скоростью работы существующих средств анализа.

**Личный вклад диссертанта.** Все результаты диссертационной работы получены автором самостоятельно, что отражено в приводимой в конце автореферата библиографии. Программные средства, реализующие предложенные алгоритмы, разработаны автором. Работы, выполненные в соавторстве, подчинены общей постановке проблемы и концепции её решения, предложенной автором.

**Степень достоверности и апробация результатов.** Достоверность научных результатов, вынесенных на защиту, подтверждена экспериментальной проверкой скорости предложенных методов, свидетельством о регистрации программы для ЭВМ, наличием актов внедрения исследований в организациях и компаниях.

Полученные результаты докладывались на Всероссийской научно-технической конференции «Новые информационные технологии в научных исследованиях» (г. Рязань, 2011, 2013 гг.), научно-практической конференции «Традиции и инновации в лингвистике и лингвообразовании» (г. Арзамас, 2012 г.), на конференции «Математические методы в технике и технологиях» (г. Рязань, 2015 г.), 6th Seminar on Industrial Control Systems: Analysis, Modeling and Computation (г. Москва, 2016 г.), а также на научном семинаре в Рязанском государственном радиотехническом университете под руководством д.ф.-м.н., профессора Миронова В.В.

**Внедрение результатов работы.** Результаты исследований, подтвержденные соответствующими актами, внедрены:

- в компании «Консалт Недвижимость» (г. Москва) для первоначальной классификации объявлений по их наиболее вероятным целям;
- во внутренней системе генерации документов компании «ДизайнЕвроСтрой» (г. Москва) для согласования падежей;
- в учебном процессе в ФГБОУ ВПО «Рязанский государственный радиотехнический университет».

**Публикации.** Основные результаты диссертации отражены в 16 работах, 7 из которых опубликованы в изданиях из перечня ВАК, 1 публикация в каталоге Web of Science. Получено 2 свидетельства о регистрации программы для ЭВМ.

**Структура и объем работы.** Диссертация состоит из введения, четырех глав, разделенных на параграфы (15 параграфов), заключения, списка литературы, включающего 107 наименований, и 2 приложений. Работа изложена на 117 страницах стандартного машинописного текста, содержит 6 таблиц.

#### ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

**Во введении** дается обоснование актуальности темы диссертации, ставится цель и определяются основные задачи диссертации, определяется место решаемой задачи в контексте более общей задачи анализа текста на естественном языке, кратко описываются другие этапы этой более общей задачи, упоминаются отечественные и зарубежные учёные и их труды, посвящённые этой задаче, и приводится краткое содержание диссертации.

**Первая глава** представляет собой обзор существующих методов решения задачи определения и генерации форм слов естественных языков (в хронологическом порядке), с указанием их достоинств и недостатков. В заключительной части первой главы на основании сравнения методов и оценки их достоинств и недостатков делается вывод об актуальности задачи повышения скорости определения форм слов и построения универсальной модели представления знаний о формообразовании языков.

**Вторая глава** посвящена разработке модели представления правил формообразования в виде цепочек преобразования строк, позволяющей повысить скорость определения форм слов по сравнению с имеющимися аналогами.

Предлагается модель представления правил формообразования, использующая понятие цепочки элементарных преобразований строк.

Рассматривается множество  $Z$  всех конечных строк алфавита, преобразование строки определяется как функция  $P: Z \rightarrow Z$  с областью определения  $D[P]$  и областью значений  $E[P]$ . Применение этой функции к строке  $S$  обозначается как  $P(S)$ .

Вводится термин «элементарная операция» – любая такая операция  $P$ , для которой выполняются условие *невыполнимости или однозначности*

$$(\forall P \in R) (\forall S_1, S_2 \in D[P]) ((P(S_1) = P(S_2)) \Leftrightarrow (S_1 = S_2)), \quad (1.1)$$

и условие *обратимости*

$$(\forall P \in R) (\exists! P^{-1} \in R) (\forall S \in D[P]) P^{-1}(P(S)) = S. \quad (1.2)$$

В качестве примеров элементарных операций можно рассматривать:

- пустая операция, результатом которой является исходная строка;
- операция добавления или удаления цепочки символов слева  $((a +), (a -))$  или справа  $((+a), (-a))$  от исходной строки;
- дубликация, или повторение строки (например, удвоение основы слова для образования множественного числа в малайском языке: *orang* — «человек», *orangorang* — «люди»).

Приводятся свойства данных операций, следующие из их определений:

- фиктивная операция является обратной по отношению к себе самой, и ни одна другая операция не обладает таким свойством;
- операция  $(-a)$  имеет обратную операцию  $(+a)$  и обратно;
- фиктивная операция и любые операции присоединения подстрок могут быть применены к любым строкам;
- операция  $(a -)$  удаления подстроки  $a$  слева (или справа,  $(-a))$  применима только к строкам, начинающимся (или заканчивающимся) на  $a$ ;
- операция, обратная удвоению, применима только к строкам чётной длины, левая половина которых совпадает с правой.

Показывается нетривиальность свойства обратимости (1.2) путём приведения примера операции, не обладающей таким свойством: операция замены первого слева вхождения подстроки  $b$  на подстроку  $a$  не обладает свойством обратимости, поскольку из результата её применения к строке  $aab$  (то есть, из строки  $aaa$ ) невозможно установить, имела ли изначальная строка вид  $aab$  (либо же  $aba$  или  $baa$ , поскольку все три варианта вполне возможны).

Для формального описания правил формообразования вводится понятие **базиса элементарных операций**. Для построения строгого определения вводятся дополнительные термины и понятия:

- операции называются **особыми**, если они не имеют параметров (например, удвоение строки или обратная ей операция);
- операции называются **регулярными**, если таковые параметры есть – например, присоединение и отделение постфиксов (в этом случае параметром операции будет сам постфикс); регулярные операции могут иметь и более одного параметра (например, замена постфикса);
- **функция-конструктор** регулярных элементарных операций (или просто **конструктор**) – преобразование  $b: Z^K \rightarrow R$  (из множества  $K$ -элементных кортежей строк во множество элементарных операций, то есть функций  $P: Z \rightarrow Z$ ), позволяющее строить конкретные элементарные операции по некоторому правилу и имеющее  $K > 0$  аргументов-строк; число  $K$  аргументов конструктора  $b$  будем обозначать  $N[b]$ ;
- **однородные операции** – операции, построенные одним и тем же конструктором;
- **базис конструкторов** элементарных операций – конечный набор конструкторов,  $B = \{b: Z^{N[b]} \rightarrow R\}$  (множество всех регулярных операций, соответствующее базису, есть  $R_B = \bigcup_{b \in B} \bigcup_{x \in Z^{N[b]}} \{b(x)\}$ );

Тогда **базис элементарных операций** есть множество, полученное объединением  $E = R_0 \cup R_B$  множества регулярных операций, полученных с помощью базиса конструкторов, и некоторого конечного множества особых операций  $R_0 = \{P | P \text{ – особая операция}\}$ ,  $|R_0| \in \mathbb{N}$ .

Примерами базисов элементарных операций могут служить:

- **постфиксный**  $E_{\text{постф}}$ :  
 $B_{\text{постф}} = \{b_+, b_-\}$ ,  $R_B = R_+ \cup R_-$ ,  $R_0 = \emptyset$ ,  
 $E_{\text{постф}} = R_0 \cup R_B = R_B = \bigcup_{S \in Z} \{b_+(S), b_-(S)\}$ ,  
 примеры операций:  $P_1 = (+a)$ ,  $P_2 = (-a)$ ;
- **префиксный**  $E_{\text{преф}}$ , по аналогии с постфиксным,  
 примеры операций:  $P_1 = (a+)$ ,  $P_2 = (a-)$ ;
- **префиксно-постфиксный**,  $E_{\text{пп}} = E_{\text{постф}} \cup E_{\text{преф}}$ ;
- **постфиксный с редупликацией**  $E_{\text{пост+рд}}$ :  
 $P_{01} = (\times 2)$ ,  $P_{02} = (P_{01})^{-1} = (\div 2)$ ,  $R_0 = \{P_{01}, P_{02}\}$ ,  $E_{\text{постф+рд}} = R_0 \cup E_{\text{постф}}$ .

В работе изучается постфиксный базис, поскольку формообразование рассматривается на примере русского языка – языка с постфиксным формообразованием. Преимущества данного базиса заключаются в его простоте (и, как следствие, наличии высокоскоростных алгоритмов анализа, один из которых предлагается в настоящей диссертации) и универсальности (такой базис не использует особенности конкретных языков, такие как, к примеру, чередующиеся и беглые гласные).

Цепочка преобразований по определению является конечной последовательностью элементарных преобразований строк, длина цепочки (количество операций в ней) обозначается  $|C|$ , а сами операции либо указываются (если заданы ранее), например,  $C_1 = (P_1, \dots, P_n)$ , либо указываются непосредственно, например,  $C_2 = (-o+a)$ . Применение цепочки к строке будем обозначать непосредственно, например, для приведённой цепочки  $C_2$  возможно применение  $C_2(\text{дело}) = \text{дела}$ .

Поскольку любую цепочку можно рассматривать как сложную функцию (суперпозицию элементарных операций), область определения цепочки  $C$  (подмножество множества всех конечных строк  $Z$ ) будем обозначать  $D[C]$ , область значений –  $E[C]$ , то есть  $C = f: D[C] \rightarrow E[C]$ .



**Тождественными** будем считать цепочки, представляющие из себя одинаковые наборы преобразований. Если,  $C_1 = (P_{1C_1} \dots P_{nC_1})$  и  $C_2 = (P_{1C_2} \dots P_{nC_2})$ , то определение тождественности имеет вид

$$C_1 \equiv C_2 \stackrel{def}{\iff} |C_1| = |C_2| \& (P_{iC_1} \dots P_{iC_2}, i = \overline{1, |C_1|}). \quad (1.3)$$

**Эквивалентными** будем считать цепочки, если и множества строк, допускающих применение этих цепочек, и результаты применения этих цепочек к любой из строк, совпадают. Определение эквивалентности можно кратко записать в виде

$$C_1 = C_2 \stackrel{def}{\iff} D[C_1] = D[C_2] \& (\forall S \in D[C_1] C_1(S) = C_2(S)). \quad (1.4)$$

Так, цепочки  $(-a+b-\bar{b})$  и  $(-a)$  эквивалентны, а  $(-a+a)$  и  $(-a\bar{b}+a\bar{b})$  – нет, поскольку множества строк, допускающих их применение, отличаются.

Для заданного базиса операций  $E$  множество всех цепочек можно описать конструктивно:  $G_E = \{C \in 2^E: |C| < \aleph_0\}$ .

**Взаимно обратными** будем считать цепочки  $C$  и  $C^{-1}$ , если для любого слова  $S_1$ , к которому  $C$  применима с результатом  $S_1'$ , применение  $C^{-1}$  к  $S_1'$  возможно и даёт  $S_1$  в качестве результата, и, наоборот, для любого слова  $S_2'$ , к которому  $C^{-1}$  применима с результатом  $S_2$ , применение  $C$  к  $S_2$  возможно, и даёт  $S_2'$ . Формально  $C$  и  $C^{-1}$  взаимно обратны, если

$$(\forall S_1 \in D[C](C^{-1}(C(S_1)) = S_1)) \& (\forall S_2 \in D[C^{-1}](C(C^{-1}(S_2)) = S_2)). \quad (1.5)$$

Показывается существование обратной цепочки для всякой цепочки  $C$  при выполнении условий  $|C| > 0$  и  $D[C] \neq \emptyset$ : для цепочки  $C = (P_1 \dots P_n)$  обратной будет являться цепочка  $C^{-1} = (P_n^{-1} \dots P_1^{-1})$ .

Во множестве цепочек имеется целый класс неэквивалентных друг другу цепочек, каждая из которых обратна самой себе. Примерами могут служить цепочки вида  $(-a+a)$ , которые **не являются** эквивалентными нулевой цепочке, поскольку не могут применяться к строкам, не оканчивающимся на  $a$ . Эти цепочки, далее именуемые **фиктивными**, не меняют строки, однако ценны тем, что дают определение подмножества строк, к которому они применимы.

Цепочка  $C$  называется **избыточной**, если существует цепочка  $C^*$ , эквивалентная ей и состоящая из меньшего числа операций.

Например, цепочка  $(-a-\bar{b})$  может быть переписана в виде  $(-\bar{b}a)$ ; цепочка  $(-a+b\bar{v}-\bar{v})$  избыточна по сравнению с эквивалентной ей  $(-a+\bar{b})$ .

Формально, цепочка  $C$  избыточна, если

$$\exists C^* \in G_B: (C = C^*) \& (|C| > |C^*|). \quad (1.6)$$

**Редукция** – это процесс получения цепочки, эквивалентной данной, но не обладающей избыточностью.

**Редуцированные цепочки** – цепочки, редукция которых невозможна.

При таком определении редуцированных цепочек естественным образом возникает вопрос о возможности устранения избыточности произвольных цепочек алгоритмическим путём и вопрос о максимальной длине редуцированной цепочки.

Ответы на оба этих вопроса даёт следующее утверждение.

**Утверждение.** Существует алгоритм, позволяющий для любой цепочки в постфиксном базисе с непустой областью применимости за полиномиальное время получить эквивалентную ей редуцированную цепочку, причём эта цепочка не будет содержать в себе более двух элементарных операций.

Для доказательства этого утверждения приводится алгоритм последовательного упрощения цепочек преобразований строк в постфиксном базисе и даётся анализ его работы.

Предложенный алгоритм для любой цепочки из  $N$  операций выполнит не более чем  $5(N^2 + N)/2$  сравнений с таблицей замен, и не более  $N - 1$  замен внутри редуцируемой цепочки, то есть предложенный алгоритм является полиномиальным от количества операций в цепочке.

Показывается, что всякая редуцированная цепочка соответствует одной из следующих схем: пустая цепочка, отделение по

стфикса ( $-a$ ), присоединение постфикса ( $+a$ ), присоединение постфикса при условии наличия постфикса ( $-a+ab$ ), удаление части постфикса, ( $-ab+a$ ), замена постфикса ( $-a+b$ ).

Все перечисленные схемы соответствуют общему случаю операции замены окончания (в случае присоединения заменяется пустое окончание, в случае отделения пустое окончание заменяет собой имевшееся).

В заключительной части главы 2 приводятся подходы к повышению скорости определения форм слов.

Для количественной оценки скорости определения форм слов была набрана база знаний о формообразовании более 8000 слов русского языка: более 4100 существительных, более 2200 глаголов, более 1600 прилагательных, более 300 наречий, 50 числительных, 27 местоимений и 104 неизменяемых служебных слова. Сама база знаний, загруженная в оперативную память в виде иерархии объектов, занимала около 20 мегабайт. Изначальный алгоритм определения форм слов на тестовом оборудовании обеспечивал скорость определения около 4000 слов в секунду, не требуя при этом дополнительных затрат оперативной памяти.

Первый предложенный подход заключается в представлении правил формообразования языка (цепочек) в виде дуг, соединяющих вершины двух встречных префиксных деревьев. Подход иллюстрирует рисунок 1.

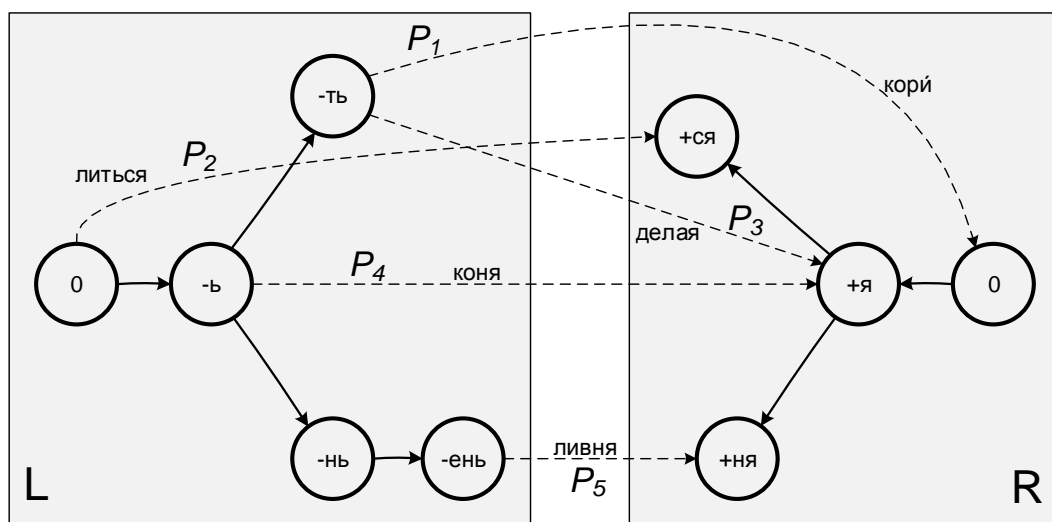


Рисунок 1 – представление цепочек в виде дуг, соединяющих вершины двух графов

При таком представлении правил формообразования увеличение скорости определения форм слов достигается за счёт исключения из рассмотрения тех цепочек, которые включают отсоединение постфикса, не присущего анализируемому слову. Предлагается алгоритм анализа форм слов, использующий такое представление правил формообразования языка и пример его применения.

В качестве исходных данных для алгоритма используются:

- само анализируемое слово  $S$ ;
- левый и правый («встречные») графы;
- дуги  $P_k$  (пунктирные линии на рисунке 1), представляющие цепочки преобразований, допустимые в рамках данного языка;
- области применимости (в виде множеств начальных форм) каждой из цепочек преобразований.

Алгоритм анализатора слов в этом случае содержит два этапа: предварительный, выполняемый один раз при запуске анализатора, и основной, выполняемый для каждого анализируемого слова.

В рамках данного алгоритма используется понятие цепи (такого связного графа или подграфа, который не имеет циклов и вершин с более чем двумя инцидентными дугами), поэтому, чтобы избежать путаницы, вместо термина «цепочка преобразований строк» в описании данного алгоритма будет использоваться термин «правило».

Предварительный этап заключается в построении быстрых (в смысле поиска) структур данных для хранения списков начальных форм слов, доступных для каждой из цепочек.

Основной этап алгоритма, ответственный за анализ конкретного слова  $X$ , описывается следующей последовательностью шагов:

1. выделить цепь  $Z$  вершин  $R_0 \dots R_n$  правого дерева, соответствующую слову, которое необходимо проанализировать (перечисление справа налево;  $R_0$  соответствует пустому окончанию); эта цепь позволит исключить из множества всех возможных вариантов разбиения этого слова на основу и окончание те варианты, которые не предусмотрены деревом окончаний, соответствующим конкретному языку;
2. выделить очередной ( $i = \overline{0, n}$ ) допустимый вариант разбиения слова на основу и окончание  $S = B_i + R_i$  путём перебора вершин выделенной цепи правого графа;
3. осуществить идентификацию словоформы, имеющей окончание  $R_i$ , соответствующее текущей вершине цепи:
  - а) отыскать все правила  $P_1 \dots P_m$ , присоединяющие окончание, соответствующее данному узлу цепи (на рисунке 1 – все пунктирные дуги, входящие в узел  $R_i$ ) – все эти правила будут иметь вид  $P_j = (-L_j + R_i)$ , где  $j = \overline{0, m}$ ;
  - б) для каждого из найденных правил  $P_j$  получить подписание всех начальных форм  $W_{C_j}$ , к которым это правило применимо, и проверить наличие в этом списке начальной формы  $W_j = B + L_j$ ;
  - в) если начальная форма  $W_j$  найдена в списке  $W_{C_j}$ , значит, считать одним из вариантов трактовки вариант  $x = (P_j, W_j, G_j)$ , где  $G_j$  – грамматическая информация, соответствующая правилу  $P_j$ ;
4. полученное множество  $X = \{x_s\}, s = \overline{1, l}$  трактовок словоформы и есть результат её анализа; длина  $l$  этого списка определяется количеством найденных на этапе (3.в) вариантов, и если список пуст, то слово системе неизвестно.

Схема основного этапа алгоритма приведена на рисунке 2.

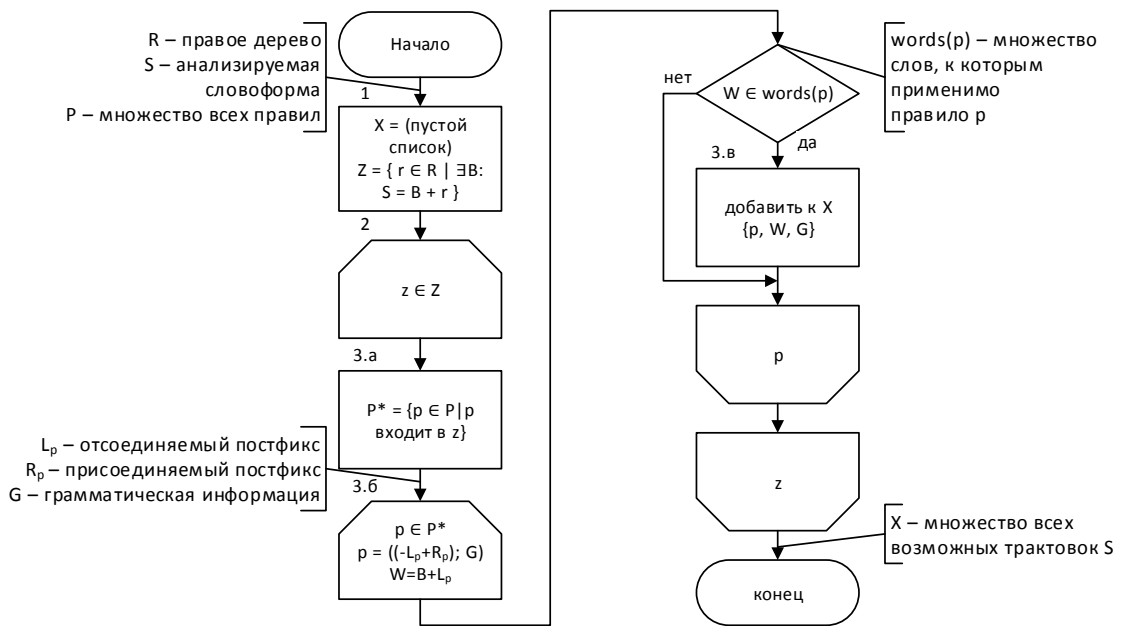


Рисунок 2 – основной этап алгоритма анализа формы слова

Скорость определения форм слов при применении такого алгоритма увеличивается как минимум десятикратно по сравнению с исходным алгоритмом, и для обычных текстов (с повторяющимися словами) растёт пропорционально числу проанализированных слов благодаря применению кэширующих контейнеров. Дополнительные затраты оперативной памяти при этом также были на порядок меньше (около 2 Мб) затрат на представление самого словаря.

Второй подход к ускорению анализа заключается в предварительном построении полного банка всех форм всех известных (хранимых в словаре) слов (получение такого банка возможно при наличии полного набора правил формообразования языка). Недостатком является размер такого банка, уже для тестовой базы знаний составившего около 250 мегабайт.

Приводится алгоритм анализа, включающий два этапа – построение такого полного банка словоформ и поиск анализируемого слова в нём. Такой подход целесообразен для серверного программного обеспечения, так как построение полного банка словоформ выполняется лишь один раз.

Таблица 1 иллюстрирует прирост скорости определения форм слов по сравнению с существующей системой определения форм слов Yandex Mystem.

Таблица 1. Скорость определения форм слов (слов/с)

Эксперимент	Система анализа форм слов Mystem	Реализация алгоритма на префиксных деревьях	Определение через генерацию
250000 уникальных словоформ	80000	55000	500000
Обычный текст, 250000 слов	120000	180000	800000 – 1000000

В третьей главе предлагается способ представления знаний о формообразовании, обладающий преимуществами над существующими подходами с точки зрения процедуры определения форм слов.

Во вводной части главы описываются существующие способы организации знаний о формообразовании естественных языков (различные способы организации

словарей формообразования) и обосновывается практическая полезность структуризации грамматической информации.

Затем даются определения и приводятся примеры основных компонентов грамматической информации (*грамматических категорий*, *GType* и соответствующих им грамматических значений – *граммем*, *GValue*).

После этого вводится понятие карты грамматических значений (набора пар «грамматическая категория – значение») и, окончательно, *правила получения словоформ* в том виде, в котором оно используется при построении словаря.

С использованием введённых понятий определяется представление *парадигмы* и, следовательно, способа построения множества всех возможных словоформ для некоторого слова (или группы слов) в начальной форме.

Наиболее простым способом построения такого множества является перечисление всех изменяемых грамматических категорий и рассмотрение всех возможных комбинаций их значений; такая схема даёт приемлемые результаты для русских существительных, однако её использование затруднено уже для прилагательных (приводится таблица, демонстрирующая большое количество запрещённых комбинаций), и совершенно непрактично для глаголов (запрещённых комбинаций гораздо больше разрешённых).

Рассматриваются подходы к решению этой проблемы (фиктивные грамматические категории, например, «род и число прилагательного», ручная правка отдельных цепочек, дополнительные наборы правил-ограничений) и показывается необходимость более сложной структуры множества правил формообразования, чем структура, опирающаяся на прямое произведение множеств граммем, соответствующих грамматическим категориям.

Вводится понятие *карты формообразования* (*InflectionMap*) – структуры данных, описывающей совокупность карт грамматических значений через перечисление независимых грамматических категорий:

$$InflectionMap = \{GTypes, Restrictions, Constants\}, \quad (1.7)$$

где *GTypes* – множество всех независимых грамматических категорий, *Restrictions* – множество ограничений, *Constants* – карта (возможно, пустая) постоянных граммем.

Множество ограничений представляет собой ассоциативный массив пар вида «{ *грамматическая категория*: { *недопустимые граммемы* } }», например, для возвратного местоимения – «{ *падеж*: { *именительный* } }» (возвратное местоимение *себя* в русском языке не имеет именительного падежа).

Далее предлагается иерархия типов слов в словаре, основанная на разделении всех слов по системе *супертипов*, *типов* и *семейств*.

Супертипы (*Supertypes*, классы верхнего уровня в структуре словаря формообразования языка) соответствуют частям речи (для тех частей речи, которые не характеризуются словоизменением, естественным образом выделяется общий супертип «неизменяемые слова»).

Типы (*Kinds*) слов в предлагаемой схеме организации словаря объединяют слова, парадигмы которых имеют одинаковую структуру (описываемую одним и тем же набором карт формообразования).

Семейства (*Families*) описывают наборы слов, подчиняющихся одним и тем же правилам формообразования (то есть, имеющим идентичную структуру парадигм и одинаковые наборы соответствующих правил получения словоформ из начальных форм слов).

Цепочка преобразований (*Chain*) в этом случае описывает правило получения словоформы в конкретном семействе и, помимо самой последовательности преобразований строки, содержит ссылку на породившую её карту формообразования и карту грамматических значений, которыми эта цепочка наделяет слово. Эта карта значений содержит только граммы из числа переменных грамматических категорий в карте формообразования; значения прочих грамматических категорий берутся из карт родительских сущностей.

Далее приводится формальная грамматика языка описания словаря формообразования. Часть грамматики, описывающая представление супертипов, имеет вид:

```
SUPERTYPES → SUPERTYPE
SUPERTYPES → SUPERTYPES SUPERTYPE
SUPERTYPE → NAME { CONST_GTYPES INFLECTION_MAPS KINDS }
CONST_GTYPES → const gtype REFS;
```

Часть словаря, описанная с применением этого языка, описывающая глагол (как части речи, имеющую самое сложное формообразование в русском языке), имеет вид:

```
supertype 'Глагол' verb
{
  const gtype 'Вид глагола', 'Переходность';
  inflection map
    'Причастие в полной форме в единственном числе'
    sing_part_full_imap
  {
    'Число' = 'Единственное', 'Форма глагола' = 'Причастие',
    'Форма причастия или прилагательного' = 'Полная',
    'Возвратность' = 'Нет';
    exclude 'Винительный' from 'Падеж';
    gtype 'Род', 'Падеж',
      'Время деепричастия или причастия',
      'Залог причастия';
  }
  (...другие карты формообразования русских глаголов...)
  kind 'Непереходные совершенные'
  { (...описание типа глаголов и входящих в него семейств...) }
  (...другие типы глаголов...)
}
```

Достоинства предложенного способа организации словаря заключаются в снижении избыточности (что проявляется при добавлении в словарь новых семейств или типов слов, и при применении алгоритмов предсказания окончаний неизвестных слов задача решается ещё быстрее), повышении точности и качества описания формообразования классов слов, для которых структура парадигмы сильно отличается от простого произведения множеств граммем независимых грамматических категорий, наличии системы типов, детально описывающих формообразование, что облегчает использование словаря в системах, выполняющих синтаксический анализ текста.

Предлагается способ представления результатов анализа текста, приводятся формат файла и алгоритм кодирования результатов анализа. Основным преимуществом морфологического представления текста является тот факт, что повторный его анализ проводить уже не требуется.

Суть заключается в том, что в файл записываются коды начальных форм и закодированная грамматическая информация в двоичном виде, при этом наиболее часто встречающиеся слова получают самые короткие коды.

Несмотря на то, что данное представление добавляет к тексту дополнительную информацию, оно сокращает его объём по сравнению с исходным файлом, поскольку код основы чаще всего занимает меньше байт, чем её строковое представление.

Приводятся алгоритмы декодирования и кодирования текста, оценивается коэффициент сжатия (таблица 2).

Таблица 2. Количественная оценка эффекта морфологического сжатия текста

Характер текста	Объём текста (слов / байт)	Сжатие архиватором (байт)	Морфологическое сжатие (байт)	Сжатие архиватором после морфологического сжатия (байт)
Текст типового договора 2 сторон	513 / 7168	2259 (31%)	2727 (38%)	1911 (27%)
Научная статья (А. В. Пруцков)	3283 / 50442	8647 (17.1%)	15233 (30%)	8378 (16.6%)
Роман «Сами боги» (А. Азимов)	82797 / 950571	188251 (20%)	327730 (35%)	167869 (18%)
Введение в философию: учебное пособие для вузов	294149 / 4161770	691893 (17%)	1535125 (37%)	656488 (16%)

**Четвёртая глава** описывает разработку комплекса программ анализа и генерации форм слов естественных языков, включающего редактор словаря, анализатор форм слов и систему проверки знаний формообразования.

В главе решены следующие задачи:

- реализованы алгоритмы для работы с цепочками преобразований строк и получено экспериментальное подтверждение того, что практическая реализация предлагаемых алгоритмов обеспечивает приемлемую скорость определения форм слов и позволяет обрабатывать все виды цепочек для выбранного базиса элементарных операций;

- спроектирована система классов, соответствующая предложенной в главе 3 структуре словаря формообразования, в рамках этой системы описано формообразование русского языка и, тем самым, доказано, что предложенная автором в главе 3 модель решает задачу описания формообразования естественного языка;

- разработаны алгоритмы, решающие задачу автоматизации процесса наполнения словаря системы, опирающиеся на спроектированную систему классов, осуществлена их реализация и оценена их эффективность, благодаря чему, определены преимущества предлагаемой модели описания формообразования естественных языков;

- сформулированы выводы о границах применимости предложенной модели описания формообразования, определены направления дальнейшего развития созданной системы.

Получено свидетельство Роспатента об официальной регистрации программы для ЭВМ «Информационная система проверки знаний по формообразованию естественных языков» (SALVINIA) № 2011611621 от 17.02.2011.

В качестве примера практического применения созданного комплекса рассматривается использование предложенных алгоритмов в электронных словарях (автор

принял участие в разработке информационной системы. «Русско-английский словарь математических терминов» (комплекс программ), рег. № 18 951 в Объединённом фонде электронных ресурсов «Наука и образование» РАО).

Приводятся статистические данные, показывающие выигрыш, обеспечиваемый благодаря применению предложенных алгоритмов, приводятся некоторые статистические данные использованного в экспериментах тестового словаря русского формообразования.

С помощью разработанного программного комплекса были также получены статистические данные, обеспечивающие представление о количественных соотношениях сущностей базы знаний о словоизменении (супертипов, типов, семейств, правил формообразования и пр.). Так, для обработки словаря из более 8000 начальных форм слов (487 семейств в составе 49 типов, 8 супертипов) в накопленной базе знаний хранится чуть менее 27000 различных цепочек (правил), и полный банк известных словоформ включает более 250000 слов. Сравнительно низкое число начальных форм слов обусловлено в первую очередь тем, что предложенная модель представления знаний о формообразовании позволяет получать особые формы глаголов из их начальных форм без необходимости введения дополнительных начальных форм. Это, помимо прочего, позволяют сделать выводы об адекватности модели представления правил формообразования реальной структуре языка.

Помимо этого, приводятся данные, показывающие коэффициент морфологического сжатия (отношение размера файла с анализируемым текстом к размеру файла с результатами анализа) для разных тестовых файлов и приводятся объёмы памяти, необходимые как для хранения словаря на диске, так и для представления структур данных в оперативной памяти ЭВМ, как для алгоритма с применением встречных префиксных деревьев, так и для алгоритма, использующего банк всех известных словоформ.

Полученные результаты свидетельствуют, во-первых, о достижении поставленной цели – повышения скорости определения форм слов естественного языка, во-вторых, о практической применимости и полезности предлагаемых алгоритмов анализа, а равно и предлагаемой модели представления данных, и её преимуществах перед существующими аналогами.

#### ЗАКЛЮЧЕНИЕ

Настоящая диссертация является научно-квалификационной работой, в которой содержится решение научной задачи построения универсальной модели формообразования флективных языков и увеличения скорости определения форм слов благодаря применению алгоритмов, построенных на основе этой модели, что имеет важное значение для развития научной отрасли автоматической обработки текстов.

Основные результаты диссертационного исследования состоят в следующем:

1. Проведён анализ существующих методов определения и генерации форм слов естественных языков, выявлены их достоинства и недостатки;
2. Разработаны алгоритмы, обеспечивающие высокую скорость определения форм слов (достигнут полуторакратный прирост скорости определения форм слов по сравнению с существующими аналогами) даже на обычных персональных компьютерах, не опирающихся на особенности одного конкретного языка;
3. Предложен способ представления знаний о формообразовании естественного языка, использующий универсальную иерархическую структуру словаря, позволяющий эффективно описывать формообразование языка и обладающий меньшей избыточностью по сравнению с простыми моделями организации словарей, что



позволяет строить алгоритмы предсказания парадигм неизвестных слов, снижая тем самым объём работы оператора, заполняющего словарь;

4. Разработаны формат и алгоритмы морфологического представления текста, позволяющие хранить уже проанализированные тексты (что снимает необходимость их повторного анализа) и обеспечивающие сжатие текста;
5. Разработан программный комплекс, включающий в себя редактор словаря, анализатор форм слов и систему проверки знаний формообразования, построенный по модульному принципу и готовый к внедрению в прикладные продукты, требующие обработки текстов на естественных языках.

#### ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Публикации в изданиях из перечня ВАК Минобрнауки РФ

1. *Пруцков А.В., Розанов А.К.* Программное обеспечение методов обработки форм слов и числительных // Вестник Рязанского государственного радиотехнического университета. 2011. № 38. С. 78-82.
2. *Мионов В.В., Заволокин А.И., Розанов А.К.* Электронная информационно-поисковая система «Русско-английский математический словарь» // Информатизация образования и науки. 2013. №19. С. 167-176.
3. *Мионов В.В., Заволокин А.И., Розанов А.К.* Проблема формализации правил русско-английского и англо-русского переводов текстов // Информатизация образования и науки. 2014. №22. С. 149-160.
4. *Розанов А.К.* Организация словаря в системах генерации и определения форм слов естественных языков // Вестник Рязанского государственного радиотехнического университета. 2014. № 49. С. 55-63.
5. *Пруцков А.В., Розанов А.К.* Методы морфологической обработки текстов // Прикаспийский журнал: управление и высокие технологии. 2014. № 3 (27). С. 119-133.
6. *Мионов В.В., Розанов А.К.* Подходы к оптимизации алгоритма определения форм слов естественных языков, основанного на цепочках последовательных преобразований строк // Информатизация образования и науки. 2015. №25. С. 43-54.
7. *Розанов А.К., Пруцков А.В.* Способы повышения скорости работы алгоритма морфологического анализа форм слов естественных языков // Вестник Рязанского государственного радиотехнического университета. 2015. № 53. С. 65-70.

Тезисы докладов Международных и Всероссийских конференций, статьи в сборниках научных статей

8. *Розанов А.К.* Основные подходы к решению задачи генерации и определения форм слов естественных языков // Традиции и инновации в лингвистике и лингвообразовании: сборник статей по материалам второй научно-практической конференции с международным участием / отв. ред. К.А.Власова; АГПИ – Арзамас: АГПИ, 2012. С. 30-34.
9. *Розанов А.К.* Использование префиксных деревьев для оптимизации доступа к наборам строк // Новые информационные технологии в научных исследованиях: материалы XVIII Всероссийской науч.-техн. конф. студентов, молодых ученых и специалистов / Рязань, РГРТУ, 2013. С. 52-53.
10. *Розанов А.К.* Представление правил префиксных и постфиксных преобразований строк на основе префиксных деревьев // Новые информационные технологии в

научных исследованиях: материалы XVIII Всероссийской науч.-техн. конф. студентов, молодых ученых и специалистов / Рязань, РГРТУ, 2013. С. 53-55.

11. *Розанов А.К.* Метод предсинтаксического анализа текста на основе знаний о формообразовании естественного языка // Математические методы в технике и технологиях. Материалы XXVIII международной научной конференции / Рязань, 2015. С. 224-228.
12. *Розанов А.К.* Быстрый алгоритм анализа словоформ естественного языка с трёхуровневой моделью словаря начальных форм // Cloud of Science. 2016 Т. 3 №1. С. 115-124.
13. *Mironov V., Zavolokin A., Rozanov A.* Preparing electronic handbook for using active grammar during process of translation of technical texts into English // SHS Web of Conferences 29 (2016) 02029 DOI: 10.1051/shsconf/20162902029.

Публикации, включённые в каталог Web of Science

14. *Rozanov A.* The fast vocabulary-based algorithm for natural language word form analysis // ITM Web of Conferences. 6<sup>th</sup> Seminar on Industrial Control Systems: Analysis, Modelling and Computation. 2016. Vol.6. P. 03013.

Свидетельства о регистрации программ для ЭВМ:

15. *Пруцков А.В., Розанов А.К.* Свидетельство Роспатента об официальной регистрации программы для ЭВМ «Информационная система проверки знаний по формообразованию естественных языков» (SALVINIA) № 2011611621 от 17.02.2011.
16. *Миронов В.В., Бухенский К.В., Заволокин А.И., Розанов А.К.* Информационная система «Русско-английский словарь математических терминов» (Комплекс программ). Издание 1.2 (исправленное и дополненное) / М.: РАО, Объединенный фонд электронных ресурсов «Наука и образование». 2013. Рег. № 18 951.

Розанов Алексей Константинович

МАТЕМАТИЧЕСКОЕ, АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ  
ОБЕСПЕЧЕНИЕ АВТОМАТИЧЕСКОГО ПРЕДСИНТАКСИЧЕСКОГО АНАЛИЗА  
ТЕКСТА В СИСТЕМАХ УПРАВЛЕНИЯ БАЗАМИ ЛИНГВИСТИЧЕСКИХ  
ЗНАНИЙ

АВТОРЕФЕРАТ

диссертации на соискание ученой степени  
кандидата технических наук

Подписано в печать \_\_\_\_\_ 2017. Формат бумаги 60×84 1/16.

Бумага офсетная. Печать трафаретная. Усл. печ. л. 1,0.

Тираж 100 экз. Заказ \_\_\_\_\_

Рязанский государственный радиотехнический университет.

390005, г. Рязань, ул. Гагарина, 59/1.

Редакционно-издательский центр РГРТУ.