

*А.В. Пруцков*

## ГЕНЕРАЦИЯ И ОПРЕДЕЛЕНИЯ ФОРМ СЛОВ ЕСТЕСТВЕННЫХ ЯЗЫКОВ НА ОСНОВЕ ИХ ПОСЛЕДОВАТЕЛЬНЫХ ПРЕОБРАЗОВАНИЙ

*Предложен универсальный метод генерации и определения форм слов естественных языков различных семейств и групп. Метод основан на представлении формообразования в виде последовательности преобразований и предназначен для решения задач на морфологическом уровне обработки текстов. Доказаны возможность применения метода для различных естественных языков и адекватность предложенной модели формообразования.*

*Ключевые слова: морфологический анализ, морфологический синтез, автоматическая обработка текстов*

**Автоматическая обработка текстов и ее уровни.** Автоматическая обработка текстов представляет преобразование текста на естественном языке с помощью ЭВМ. Автоматическая обработка текстов используется при решении следующих задач: машинный перевод; проверка правописания; анализ текстов и выявление в них знаний; диалог ЭВМ с пользователем на естественном языке; анализ запросов в информационно-поисковых системах глобальных информационных сетей; анализ и синтез речи.

Любой пользователь сталкивается с данными задачами практически ежедневно.

Можно выделить три уровня обработки текстов: морфологический, синтаксический, семантический.

На семантическом уровне связываются слова, сочетания и их смысловое значение.

На синтаксическом уровне определяется взаимосвязь слов в предложении.

На морфологическом уровне связываются слова предложения и их грамматические значения (падеж, время и т. п.).

**Задачи генерации и определения.** Рассмотрим более подробно морфологический уровень обработки текстов.

Морфологический анализ и синтез работают с предложениями.

При морфологическом анализе для каждого слова решается задача определения. Определение заключается в нахождении по данной словоформе ее нормальной формы (основы) и грамматического значения.

Морфологический синтез состоит в решении для каждого слова задачи генерации. Генерация формы слова – это процесс получения формы с использованием в качестве начальных параметров основы и грамматического значения.

**Метод генерации и определения форм слов.** Задачей данной работы является разработка метода генерации и определения форм слов, удовлетворяющего следующим критериям универсальности:

1) универсальность метода для естественных языков различных групп и семейств;

2) универсальность структуры данных, не требующей конвертации для решения задач определения или генерации;

3) универсальность метода для возможных типов формообразования словоформ;

4) универсальность метода для любых видов аффиксов;

5) универсальность метода для обработки всей парадигмы слова, в том числе и синтетических (сложных) словоформ.

Существует множество методов решения задач генерации и определения форм слов, разработанных отечественными учеными [1-6] и зарубежными специалистами [7-9]. Однако они удовлетворяют не всем перечисленным критериям универсальности, а лишь некоторым.

Предлагается модель формообразования естественных языков (далее модель формообразования), состоящая в том, что получение словоформы с данным грамматическим значением можно представить в виде последовательности конечного числа преобразований над основой.

Можно выделить два основных типа преобразований:

1) добавление подстроки S к строке P слева (префиксы) (обозначается S+) или справа (постфиксы) (+S) без изменения самой строки;

2) замена первой слева подстроки S на подстроку P ( $S \rightarrow P$ ).

Каждое преобразование имеет обратное к нему, то есть совершающее обратное действие. Перечисленным типам преобразований соответ-

ствуют следующие типы обратных преобразований:

1) отделение подстроки S от строки P слева (S-) или справа (-S);

2) обратная замена первой слева подстроки P на подстроку S ( $P \rightarrow S$ ).

Все остальные преобразования можно представить в виде перечисленных основных и/или обратных типов преобразований.

Чтобы использовать преобразование в рамках данной модели, преобразование должно обладать следующими свойствами:

1) однозначность результата: преобразование всегда приводит к одному и тому же результату;

2) обратимость действия: применение к строке прямого, а затем обратного преобразований не изменяет ее.

Несколько преобразований можно объединить в цепочку. Цепочка преобразований (комбинированное преобразование) – это конечная последовательность преобразований:

$$R = (Q_1, Q_2, \dots, Q_n).$$

Цепочка преобразований, трансформирующая основу слова S в форму слова F, называется прямой цепочкой преобразований (R), а форму F в основу S – обратной цепочкой преобразований (R'). Обратная цепочка преобразований R' представляет собой обратную последовательность преобразований, обратных данным:

$$R' = ((Q_n'), (Q_{n-1}'), \dots, (Q_1')).$$

Применить преобразование Q к форме F означает выполнить преобразование Q над формой F, чтобы получить форму Ф. Преобразование Q может быть неприменимо к форме F, например, невозможно отделить постфикс из-за его отсутствия или невозможно произвести замену подстрок из-за отсутствия исходной подстроки в форме. Если одно преобразование цепочки R неприменимо к форме F, то вся цепочка R считается неприменимой к форме F и процесс применения преобразований цепочки R к форме F прекращается. Если преобразование или цепочка преобразований применимы, то их результатом является преобразованная форма. Если преобразование или цепочка преобразований неприменимы к форме, то результат применения неопределен.

Для цепочек преобразований будем использовать следующие условные обозначения:

$V = A(R)$  – применить к форме A прямую цепочку преобразований R и получить форму B;

$V = A(R')$  – применить к форме A обратную цепочку преобразований R' (штрих) и получить форму B.

Слова, имеющие одинаковые цепочки для получения словоформ, соответствующих одному грамматическому значению, объединяются в один тип формообразования.

Рассмотрим примеры записи цепочек преобразований слов естественных языков.

Пример 1. Глагол немецкого языка *aufmachen* – «открывать, оформлять» в настоящем времени *das Präsens des Indikativs* имеет следующую схему спряжения:

<i>mache auf</i>	<i>machen auf</i>
<i>machst auf</i>	<i>macht auf</i>
<i>macht auf</i>	<i>machen auf</i> .

Образование глаголов немецкого языка с отделяемыми приставками можно представить в виде цепочки, состоящей из трех преобразований (в терминах немецкого языка):

1) отделение приставки (обратное преобразование первого типа);

2) добавление личного окончания (прямое преобразование второго типа);

3) добавление приставки в конец основы (прямое преобразование второго типа).

Для спряжения глагола *aufmachen* в 3-м лице единственном числе цепочка R будет состоять из трех преобразований:

Q<sub>1</sub>: отделить префикс «auf»: *auf*-;

Q<sub>2</sub>: добавить постфикс «t»: *+t*;

Q<sub>3</sub>: добавить пробел и постфикс «auf»: *+\_auf*,

и иметь вид:

$$R = (Q_1, Q_2, Q_3) = (auf^-, +t, +_auf).$$

Пусть необходимо получить форму 3-го лица единственного числа  $F = \langle \text{macht auf} \rangle$  из основы инфинитива  $S = \langle \text{aufmach} \rangle$ . Для этого необходимо применить цепочку R к основе S:

1) отделить префикс «auf»: *aufmach* → *mach*;

2) добавить постфикс «t»: *mach* → *macht*;

3) добавить пробел и префикс «auf»: *macht* → *macht auf*.

В результате применения цепочки R, состоящей из трех преобразований, из основы  $S = \langle \text{aufmach} \rangle$  получена словоформа  $F = \langle \text{macht auf} \rangle$ .

Чтобы получить из словоформы  $F = \langle \text{macht auf} \rangle$  основу  $S = \langle \text{aufmach} \rangle$ , необходимо применить обратную цепочку преобразований R', состоящую из обратных преобразований цепочки R, выполненных в обратной последовательности:

(Q<sub>3</sub>') : отделить пробел и постфикс «auf»: *-\_auf*,

(Q<sub>2</sub>') : отделить постфикс «t»: *-t*;

(Q<sub>1</sub>') : добавить префикс «auf»: *auf*+;

и иметь вид:

$$R' = ((Q_3'), (Q_2'), (Q_1')) = (-_auf, -t, auf^+).$$

Применим обратную цепочку R' к словоформе F = «*macht auf*»:

1) отделим пробел и постфикс «*auf*»: *macht auf* → *macht*;

2) отделить постфикс «*t*»: *macht* → *mach*;

3) добавить префикс «*auf*»: *mach* → *aufmach*.

В результате применения обратной цепочки R', состоящей из трех преобразований, из словоформы F = «*macht auf*» получена основа S = «*aufmach*». □

Пример 2. Рассмотрим спряжение местоименного глагола испанского языка *lavarse* – «*умываться*» в настоящем времени *Presente de Indicativo* и представим одну из форм с помощью цепочки преобразований.

Схема спряжения местоименного глагола *lavarse* в *Presente de Indicativo* имеет следующий вид:

<i>me lavo</i>	<i>nos lavamos</i>
<i>te lavas</i>	<i>os laváis</i>
<i>se lava</i>	<i>se lavan.</i>

Получим форму *nos lavamos* из основы инфинитива *lav* с помощью следующей цепочки преобразований:

1) прибавить к основе справа окончание «*-amos*»: *lav* → *lavamos*;

2) прибавить к основе слева возвратное местоимение «*nos*» с пробелом: *lavamos* → *nos lavamos*.

В результате применения цепочки двух преобразований из основы *lav* получена словоформа *nos lavamos*. □

Пример 3. Получим существительное падежа *Adessiivi* финского языка *liikkeellä* из основы *liike* – «*торговля, коммерция*». Цепочка преобразований будет состоять из двух преобразований:

1) заменить подстроку «*ke*» на подстроку «*kkee*» в основе: *liike* → *liikkee*;

2) добавить окончание «*-llä*»: *liikkee* → *liikkee-llä*.

В результате получена форма *liikkeellä*. □

Другие примеры представления формообразования в виде цепочек преобразований приведены в работах [10-13].

Предлагаемую модель формообразования можно использовать для решения задач генерации и определения форм слов.

**Алгоритмы генерации и определения форм слов.** Способ представления формообразования и алгоритмы генерации и определения форм слов взаимосвязаны между собой. Как правило, одна часть определяет другую.

Рассмотрим алгоритмы решения задач генерации и определения форм слов, основанные на данной модели формообразования.

Введем условные обозначения исходных данных: T – тип основы; G – грамматическое значение; n – количество цепочек.

Условные обозначения операций в алгоритмах:

Y = (X) – получить значение Y по значению X;

Исходные данные организованы следующим образом:

- одной основе S соответствует один тип основы T;

- одному типу основы T соответствует несколько основ S;

- одному сочетанию типа основы T и грамматическому значению G соответствует одна цепочка преобразований R;

- одна цепочка преобразований R соответствует одному сочетанию типа основы T и грамматическому значению G.

Задачей алгоритма генерации форм слов (рис. 1) является получение словоформы F, соответствующей основе S и грамматическому значению G. Результат генерации форм слов всегда однозначен.

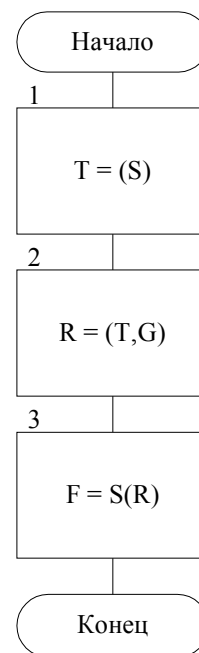


Рисунок 1 – Алгоритм генерации форм слов

Алгоритм генерации состоит из двух этапов. На первом этапе производится поиск необходимой цепочки преобразований R по входным данным: основе S и грамматическому значению G (рисунок 1, блоки 1-2). Второй этап заключается в применении найденной цепочки R к основе S для получения требуемой формы F (рисунок 1, блок 3).

Алгоритм определения форм слов (рисунок 2) определяет грамматическое значение G и ос-

нову  $S_i$ , соответствующие исходной словоформе  $F$ .

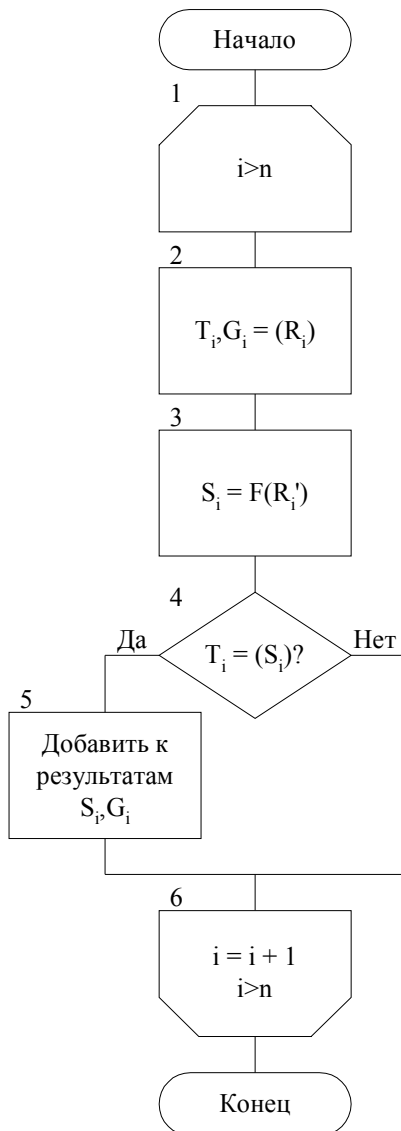


Рисунок 2 – Алгоритм определения форм слов

Алгоритм определения форм слов заключается в переборе цепочек преобразований в цикле (рисунок 2, блоки 1-6). На каждой итерации извлекается очередная цепочка  $R_i$  и соответствующие ей тип основы  $T_i$  и грамматическое значение  $G_i$  (рисунок 2, блок 2). Далее очередная обратная цепочка  $R_i'$  применяется к входной словоформе  $F$  с целью получения основы  $S_i$  (рисунок 2, блок 3).

Если найденной основе  $S_i$  соответствует тот тип основы  $T_i$ , что и очередной цепочке (рисунок 2, блок 4), то результат, состоящий из полученной основы  $S_i$  и грамматического значения  $G_i$ , добавляется к полученным результатам (рисунок 2, блок 5). На этом итерация заканчивается и происходит переход к следующей итерации (рисунок 2, блок 6). Число итераций цикла равно  $n$  – числу цепочек преобразований.

Перебор цепочек преобразований не заканчивается после нахождения первой применимой к форме  $F$  цепочки. Перебор всех цепочек необходим из-за существования в естественных языках омонимии. Одной форме  $F$  может соответствовать не одно грамматическое значение, а несколько. Следовательно, форме  $F$  могут быть применимы ни одна, а несколько цепочек преобразований.

Предлагаемые модель формообразования и алгоритмы генерации и определения форм слов составляют метод генерации и определения форм слов.

Каждое преобразование можно представить в виде графа [12]. Тогда цепочка образует путь в графе. Решение задач генерации и определения сводится к поиску в глубину решения в графе. Если вся цепочка применима, то она приводит к решению.

Были исследованы особенности формообразования следующих естественных языков с морфологией:

- русский (восточно-славянская группа индоевропейской семьи языков);
- английский (германская группа индоевропейской семьи языков);
- немецкий (германская группа индоевропейской семьи языков);
- испанский (романская группа индоевропейской семьи языков);
- финский (прибалтийско-финская группа финно-угорской семьи языков).

Данные языки выбраны по следующим причинам: принадлежность различным семействам и группам; различная степень развитости морфологии; наличие различий в формообразовании; доступность литературы по данным языкам.

В ходе проведенного анализа и тестовых экспериментов был сделан вывод о том, что предлагаемые алгоритмы способны генерировать и определять формы слов данных естественных языков.

Чтобы сделать такой вывод, необязательно генерировать и определять все формы всех слов естественного языка по следующим причинам:

1) если алгоритмы способны генерировать и определять формы некоторого слова, то алгоритмы способны генерировать и определять формы всех слов, имеющих тип основы тот же, что и у данного слова;

2) если алгоритмы способны генерировать и определять формы, получаемые с помощью длинных цепочек преобразований, включающих, например, присоединение префикса, постфикса и изменение основы, то алгоритмы способны генерировать и определять формы, получаемые с

помощью более коротких цепочек, включающие те же типы преобразований;

3) если алгоритмы способны генерировать и определять формы, получаемые с помощью некоторой цепочки преобразований, то алгоритмы способны генерировать и определять формы, получаемые с помощью той же цепочки преобразований, но с другими аргументами преобразований, составляющих цепочку.

**Условие применимости алгоритмов генерации и определения к языку и теорема о представлении формообразования в виде цепочки преобразований.** Предложенные алгоритмы способны генерировать и определять формы слов рассмотренных выше естественных языков. Этот вывод был сделан на основе анализа формообразования данных языков.

Алгоритмы основаны на модели формообразования, которая заключается в представлении формообразования в виде конечной последовательности преобразований над основой. Поэтому можно сформулировать свойство языка, наличие которого позволяет алгоритмам генерировать и определять формы слов, в виде условия применимости алгоритмов к языку.

Условие применимости алгоритмов генерации и определения к языку заключается в возможности записи любого формообразования данного языка цепочками преобразований. Если такая запись возможна, то будем говорить, что алгоритмы применимы к языку, иначе алгоритмы считаются неприменимыми к языку.

Из условия применимости можно сделать следующие выводы.

- Разработчику, желающему использовать предлагаемый метод для решения задач автоматической обработки текстов, не нужно разбираться в алгоритмах генерации и определения, чтобы определить будут ли они применимы к языку или нет. Разработчик должен просто проверить условие применимости и сделать вывод, применимы алгоритмы к языку или нет.

- Применимость разработанных алгоритмов не ограничивается только пятью указанными языками, так как для других естественных языков с морфологией получение формы слова можно записать в виде цепочки преобразований.

Второй вывод из условия применимости алгоритмов к языку раздвигает область применимости алгоритмов к языкам. Докажем, что условие применимости алгоритмов к языку выполняются у любого языка с морфологией в виде следующей теоремы.

Теорема 1 (о представлении формообразования в виде цепочки преобразований). Получение любой грамматической формы любого языка

(даже не естественного) с морфологией можно представить в виде цепочки преобразований.

Доказательство. Пусть в некоторой грамматической категории основе  $S$  соответствует грамматическая форма  $F$ .

Получение формы  $F$  из основы  $S$  представляет собой замену подстроки  $S$  на подстроку  $F$ :

$$S \rightarrow F.$$

Замена подстроки является одним из типов преобразований. Следовательно, получение формы  $F$  из основы  $S$  можно представить цепочкой преобразований, состоящей из одного преобразования: замены подстроки  $S$  на подстроку  $F$ . Теорема доказана. ■

Задачей теоремы 1 не является формулировка конкретного указания, как представить формообразование в виде цепочки преобразований, так как способ представления формообразования цепочкой, приведенный в доказательстве теоремы, не единственный. Теорема 1 дает утвердительный ответ на вопрос, можно ли представить формообразование в виде цепочки преобразований.

Следствия из теоремы 1.

1. Теорема утверждает, что получение любой формы любого языка с морфологией можно представить в виде цепочки преобразований. В то же время по условию применимости алгоритмы применимы к языку, если получение словоформы можно представить в виде цепочки преобразований. Следовательно, алгоритмы применимы к любому языку с морфологией.

2. Предложенная модель формообразования адекватна, так как предположение, лежащее в основе модели, что получение словоформы с данным грамматическим значением можно представить в виде последовательности конечного числа преобразований над основой, доказано теоремой.

3. Теорема 1 доказывает, что любое формообразование можно записать в виде цепочек преобразований. Цепочки можно представить в виде графа. А в графе всегда можно найти решение, если оно существует, с помощью известных методов, например метода поиска в глубину. Следовательно, при представлении формообразования цепочкой преобразования, решение задач генерации и определения будет найдено, если оно существует.

**Цепочка преобразований как алгоритмическая система.** Цепочку преобразований  $R$  можно рассматривать как алгоритм преобразования формы  $F$  в форму  $\Phi$ .

Рассмотрим основные признаки алгоритма [14, 15] и соответствие цепочки преобразований данным признакам.

1. Элементарность шагов. Цепочка преобразований состоит из конечной последовательности преобразований. Каждое преобразование неделимо и локально и обладает по определению свойством однозначности результата.

2. Детерминированность. После выполнения очередного преобразования выполняется следующее преобразование или выполнение цепочки заканчивается. Выполнение цепочки заканчивается в двух случаях: в случае неприменимости цепочки к форме или окончании цепочки.

3. Результативность. Цепочка состоит из конечного числа преобразований и является линейным алгоритмом, без разветвлений и циклов. Поэтому цепочка всегда будет заканчивать свое выполнение за конечное число шагов. Если цепочка применима к форме, то результатом является преобразованная форма, иначе результат неопределен.

4. Массовость. С помощью цепочек преобразований можно описать формообразование целой группы слов, имеющих один тип формообразования.

Алгоритмы создаются в рамках алгоритмической системы. Чтобы определить алгоритмическую систему, необходимо определить входные и выходные данные, набор инструментов и способ их записи [15].

Входными, промежуточными и выходными данными цепочек преобразований являются формы слов – последовательность символов некоторого алфавита.

Набор инструментов составляют два основных типа преобразований: добавление подстроки  $S$  к строке  $P$  слева (обозначается  $S+$ ) или справа ( $+S$ ); замена первой слева подстроки  $S$  на подстроку  $P$  ( $S \rightarrow P$ ) и обратные им преобразования.

Примем за один шаг выполнение одного преобразования. В случае применимости цепочки к форме слова, трудоемкость цепочек преобразований будет равняться длине цепочки. Цепочка преобразований является линейным алгоритмом класса  $P$ , а значит эффективным.

Данная алгоритмическая система не является универсальной, в отличие от машины Тьюринга и нормальных алгоритмов Маркова [14], и предназначена только для описания формообразования естественных языков в рамках предложенной модели формообразования.

**Количественные характеристики системы генерации и определения форм слов.** На

основе предлагаемого метода создана система генерации и определения форм слов [16].

Основной количественной характеристикой работы системы является скорость обработки форм слов, то есть скорость определения и генерации.

Реализация рассмотренных методов морфологической обработки текстов затруднена, так как включает заполнение словарей декларативной части, что вызывает следующие проблемы: необходимость классифицировать слова по типам формообразования, что требует значительных временных затрат; структура и правила заполнения словарей не всегда понятны из описаний методов.

Поэтому проведем сравнение количественных характеристик со следующими аналогичными коммерческими системами морфологической обработки текстов (см. таблицу), которые доступны в глобальной информационной сети Интернет:

- МЛ Морфология SDK компании Медиалингва [17];

- RCO Morphology Professional SDK библиотека полного морфологического анализа русских слов компании RCO [18].

Тестирование предлагаемой системы проводилось на компьютерной системе DELL Latitude C640 с ОЗУ объемом 256 Мб и жестким диском объемом 20 Гб.

Из количественных характеристик, представленных в таблице, можно сделать вывод, что недостатком предлагаемой системы является низкая скорость анализа форм слов в данной программной реализации. На веб-страницах разработчиков отсутствуют описания алгоритма работы и структуры систем, что затрудняет возможность сделать вывод, за счет чего достигнуто такое преимущество скорости работы.

Характеристики	Сравниваемые системы		
	МЛ Морфология SDK (компания Медиа-лингва)	RCO Morphology Professional SDK (компания RCO)	Предлагаемая система генерации и определения форм слов
Тип и частота процессора системы, на которой проводилось тестирование	Intel Pentium; 0,2 ГГц	AMD Athlon; 1 ГГц	Intel Pentium IV Mobile; 1,8 ГГц
Скорость определения, тыс. слов/с	0,5	100	0,7
Скорость генерации, тыс. слов/с	Нет данных	Нет данных	5
Цена 1 лицензии разработчика	14 500 руб.	24 000 руб.	–

**Выводы.** Рассмотрим, удовлетворяет ли предложенный метод генерации и определения форм слов, сформулированным в начале данной статьи критериям универсальности.

Критерий 1. Анализ морфологии языков, разбор примеров, реализация системы генерации и определения форм слов показали способность предложенных алгоритмов генерировать и определять формы слов естественных языков различных групп и семейств.

Критерий 2. По одним и тем же данным осуществляется как генерация, так и определение форм слов, и при этом не требуется их конвертация.

Критерий 3. Теорема 1 доказывает, что любой тип формообразования можно представить цепочкой преобразований, а значит обработать их предложенными алгоритмами генерации и определения форм слов.

Критерий 4. Теорема 1 доказывает, что метод не зависит от вида аффикса.

Критерий 5. Теорема 1 доказывает, что получение любой формы можно описать в виде цепочки преобразований, в том числе и синтетической.

Следовательно, метод удовлетворяет всем критериям универсальности.

#### **Библиографический список**

1. Андреев А.М., Березкин Д.В., Брик А.В. Лингвистический процессор для информационно-поисковой системы // Компьютерная хроника. – 1998. – № 11. – С. 79-100.
2. Белоногов Г.Г., Богатырев В.И. Автоматизированные информационные системы / под ред. К.В. Тараканова. – М.: Сов. радио, 1973. – 328 с.
3. Демьянков В.З. Морфологическая интерпретация текста и ее моделирование. – М.: Изд-во МГУ, 1994. – 206 с.
4. Мальковский М.Г. Диалог с системой искусственного интеллекта. – М.: Изд-во МГУ, 1985. – 214 с.

5. Ножов И.М. Прикладной морфологический анализ без словаря // КИИ-2000. Труды конф. – Т. 1. – М.: Физматлит, 2000. – С. 424-429.

6. Segalovich I. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine // MLMTA-2003, Las Vegas, 2003 June.

7. Goldsmith J. Unsupervised Learning of the Morphology of a Natural Language // University of Chicago, 1998.; John Goldsmith (ed.), The Last Phonological Rule, pp. 173-194. Chicago: University of Chicago Press.

8. Koskeniemi K. (1983). Two-level Morphology: A General Computational Model for Word-form Recognition and Production. University of Helsinki, Department of General Linguistics, Publications No. 11.

9. Porter M.F. An algorithm for suffix stripping. Program, Vol. 14, No. 3, July 1980, pp 130-137.

10. Пруцков А.В. Информационно-справочный ресурс по словообразованию естественных языков // Информационные ресурсы России. – 2004. – № 6. – С. 22-24.

11. Пруцков А.В. Модульный подход к построению дистанционных систем проверки знаний // Информационные ресурсы в процессе подготовки современного специалиста: Межвуз. сб. – Липецк: ГОУ ВПО «Липецкий гос. педагогический ун-т», 2005. – С. 131-136.

12. Пруцков А.В. Методы поиска решений в лингвистических автоматизированных обучающих системах // Научно-техническая информация. Серия 2. Информационные процессы и системы. – 2006. – № 4. – С. 15-18.

13. Пруцков А.В. Определение и генерация сложных форм слов естественных языков при морфологическом анализе и синтезе // Вестник Таганрогского государственного радиотехнического университета / Таган. гос. радиотехн. ун-т. – Таганрог, 2006. – С. 70-75.

14. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. – М.: Энергоатомиздат, 1988. – 480 с.

15. Новичков В.С., Парфилова Н.И., Пылькин А.Н. Алгоритмизация и программирование на

Турбо Паскале: учеб. пособие. – М.: Горячая линия – Телеком, 2005. – 438 с.

16. Пруцков А.В. Статический и динамический подходы к проектированию подсистем проверки знаний автоматизированных обучающих систем // Информационные ресурсы России. – 2006. – № 1. – С. 27-29.

17. [http://www.medialingua.ru/products.asp?category\\_id=8800&product\\_id=8808](http://www.medialingua.ru/products.asp?category_id=8800&product_id=8808). Дата просмотра 07.07.2008.

18. [http://www.rco.ru/product.asp?ob\\_no=14](http://www.rco.ru/product.asp?ob_no=14). Дата просмотра 07.07.2008.