**A.Gelbukh**

A Model of Morphology of an Inflective Natural Language *(In Russian).*

// Proceedings of the III International Conference "PC Software", Tver, May 1990, Tver and Moscow, Russia.

# МОДЕЛЬ МОРФОЛОГИИ ФЛЕКТИВНОГО ЕСТЕСТВЕННОГО ЯЗЫКА

## 1. ВВЕДЕНИЕ

Настоящая работа посвящена описанию формальной модели морфологического строения естественного флективного языка (флективным языком называется язык, в грамматике которого морфологические характеристики словоформ выражаются вариацией их окончания - флексии. Русский и многие другие языки являются флективными.), а также обсуждению основанного на сформулированной модели экономичного по памяти и быстродействию алгоритма точного морфологического анализа и синтеза словоформ естественного языка.

Под задачей морфологического анализа мы будем понимать сопоставление словоформе набора значений ее морфологических характеристик (таких как часть речи, падеж, число и др.) и имени лексемы (или нормализованной формы), например:

"красными" --> <красный>, чр=прилаг, падеж=твор, чсл=множ,

или установление неправильности словоформы или во всяком случае отсутствия в системе знаний о ней. Под задачей морфологического синтеза мы будем понимать обратное сопоставление.

Назовем анализ (синтез) точным, если (а) учитываются все возможные варианты разбора (синтеза) словоформ, (б) выводы системы не носят вероятностного, а ее знания - эвристического характера; напротив, выводы системы являются достоверными настолько, насколько свободны от ошибок ее программное и лингвистическое обеспечение.

В случае невыполнения последнего требования анализ (синтез) называется приближенным. Так, системы, использующие эвристики по окончаниям слов (напр. [1]), знаменитое слово "кровать" считали неизвестным глаголом (ошиб. *кроваю, *кроваешь). Преимуществом таких систем обычно является высокая эффективность и возможность высказывания наиболее вероятных гипотез о форме незнакомых слов (вроде известной "глокой куздры").

В настоящее время имеется опыт создания систем морфологического анализа и/или синтеза, имеющих следующие основные сферы применения: (а) в составе лингвистического процессора системы искусственного интеллекта ([2, 3]); (б) в составе систем автоматизированной обработки текста в интеллектуальных СУБД ([1]); в составе системы автоматического перевода ([4]); а также в качестве средства автоматизированной диалоговой проверки грамматической правильности текста. Системы этого последнего типа для русского языка появились практически в последние месяцы и как правило являются разработками научно-технических кооперативов (среди таких разработок можно выделить как пример в известной степени противоположных подходов системы ДИАЛОГ и ОРФО).

Однако анализ спектра существующих разработок и подходов показывает, что каждая такая система является замкнутым и трудоемким программным продуктом. Разнообразие таких программ обьясняется разнообразием требований к ним и условий их применения и разнообразием подходов к их созданию. При этом оказывается, что, как правило, такие требования и подходы обычно можно разделить на чисто технические (формат интерфейсов, степень интерактивности, приоритеты среди параметров по оперативной и дисковой памяти и быстродействию, вид алгоритмов) и чисто лингвистические (набор характеристик словоформ, проведение границы между регулярностью, затрудненностью и неправильностью тех или иных словоформ (напр. [5, предисловие]), проведение границы между словообразованием и формообразованием).

Как, впрочем, и в других предметных областях, в области систем морфологического анализа/синтеза можно, как правило, четко разделить чисто программные, технические требования и особенности системы, с одной стороны, и чисто лингвистические, предметные ее свойства, с другой. Однако, насколько известно автору, до сих пор не было предпринято серьезных попыток разделения предметной и программной информации в этой области (хотя, конечно, в каждой отдельной системе эта информация частично разделена ([4])).

Таким образом, существует острая потребность в создании взаимоприемлемой для лингвиста и программиста модели предметной области, в рамках которой могли бы раздельно создаваться, отлаживаться, сопровождаться и изменяться как наборы лингвистических знаний, так и программные оболочки, подобные общеизвестным экспертным оболочкам ([6]).

В 1988 году лингвист Е.Добрушина и математик Г.Савина предложили некоторые идеи такого подхода ([7]). Настоящая разработка является продолжением и развитием этих идей.

Реализация прикладных систем на малых машинах и ПЭВМ накладывает особенности на всю архитектуру системы и модель предметной области, служащую для представления знаний в системе. Модель машины, на которой должна интерпретироваться предлагаемая лингвистическая модель и ради которой она создана, обладает тремя особенностями, свойственными современным малым машинам: (а) обьем памяти достаточен для хранения программного кода и небольших таблиц, но недостаточен для хранения словаря - под это условие подпадают машины (и ОС) с доступной одному процессу памятью 48 К - 1 М; (б) за один акт прямого доступа к внешней (дисковой) памяти на системно-аппаратном уровне может быть считан лишь фрагмент размером 128 байт - 6 Кбайт - блок файла; (в) любой разумный обьем обработки данных в памяти выполняется быстрее, чем один акт считывания одного блока с диска.

Предлагаемая модель предназначена для записи лингвистических знаний в реализованной автором системе, основное назначение которой состоит в (а) пакетной либо диалоговой обработке очень больших массивов текста с полным и точным распознаванием омонимии на малых машинах и ПЭВМ; (б) создании и независимой отладке вариантов лингвистического обеспечения. В связи с этим основными отличительными чертами разработки являются (а) высокое быстродействие без потери полноты распознавания омонимии и (б) полное отделение лингвистической информации от программной. Система поддерживает удобную технологию создания "с нуля" и пополнения таблиц и словаря. Для декларативной записи правил порождения ее словаря по тексту, имеющему формат словаря [9], автором разработан и реализован специальный простой язык.

При разработке модели и основанной на ней программной оболочки особое внимание автором было уделено проблеме понимания ошибочно написанных словоформ и выдаче разумных гипотез об их исправлении. Система эффективно распознает один класс характерных ошибок: ошибки, происходящие от выбора для слова окончания или формы, правильной для других слов языка, но не для данного слова. Такие ошибки особенно часто делают люди, знающие основы грамматики языка, но не владеющие языком свободно. По-видимому, ценной окажется способность системы подсказывать правильное написание исключений и трудных слов, отправляясь от предложенной невозможной формы слова.

Основные сферы применения автору представляются следующими: простые системы автоматического индексирования и ИПС, проверки правильности текста (с учетом согласования слов, т. е. с элементами синтаксического анализа), автоматического и полуавтоматического перевода, лингвистические процессоры; далее: построение различных обучающих систем для лиц, не владе-

ющих данным (например, русским) языком, построение систем для различных языков путем разработки соответствующего варианта таблиц, использование в упрощенных системах синтеза фраз и другие. Настоящая реализация будет в первую очередь внедрена во ВНТИЦентре ГКНТ СССР в качестве системы автоматического индексирования на ПЭВМ и системы проверки правильности текста также на ПЭВМ.

Автору представляется важным создание библиотек процедур (анализ текста или одного слова, синтез одного слова, исправление ошибки и др.), которые могут быть использованы в любой программе на Си или совместимом языке, что способствовало бы распространению лингвистических методов и стимулировало развитие вариантов лингвистического обеспечения. Кроме того, к такой библиотеке должен прилагаться готовый набор (в исходных текстах) оболочек, иллюстрирующих различные аспекты ее применения и имеющих самостоятельную прикладную ценность. Именно такой подход применен при реализации обсуждаемой системы.

Развитый в настоящей работе аппарат в перспективе будет применен к следующим задачам: описание аномалий начертания некоторых букв, как, например, начертание русской буквы "йо" как "е"; описание расстановки ударений, что является очень важной задачей; приближенный анализ незнакомых слов; диалоговые средства обучения новым словам, подобные примененным в системе ОРФО. Автор планирует также рассмотреть возможность применения этого аппарата к задачам анализа/синтеза звучащей речи.

Ниже мы подробно рассмотрим строение предлагаемой модели, а в заключение будет приведен один ключевой момент программной реализации, играющий основную роль в эффективности интерпретации данной модели.

## 2. ЛИНГВИСТИЧЕСКАЯ МОДЕЛЬ

Мы будем рассматривать некоторый флективный естественный язык, то есть такой, в котором лексема (слово) характеризуется цепочкой букв, называемой основой слова (далее мы увидим, что основы лексемы могут быть разные для разных ее форм), а морфологические характеристики конкретной словоформы задаются приписыванием к основе справа (нескольких) цепочек букв.

Основная схема предлагаемой модели состоит в следующем. Слово представляется разделенным на определенное число частей, которые мы далее назовем морфемами: чита-ющ-ий-ся. С каждой такой частью связывается некоторое смысловое значение (например, падеж и т.п.), которое мы далее назовем формой. В лингвистических таблицах декларативно представлена информация

трех типов:

(а) какой форме соответствует какое начертание (цепочка букв) - так, в именительном падеже надо писать "-ий-", а в родительном "-его-";

(б) для связи между морфемами указывется информация о взаимосвязи начертаний (букв) - так, после "-ющ-" пишется "-ий-": "-ющ-ий", но после "-ем-" пишется "-ый-": "-ем-ый";

(в) и аналогично информация о взаимосвязи смыслов (форм) - так, можно писать "-ющ-(ий)-ся" и нельзя "-ем-(ый)-ся".

При этом такая информация делится на две части: (1) словарная информация - информация, имеющая простую структуру, приписанная основам и характеризущая лексику языка, и (2) иерархически организованная информация, приписанная окончаниям и формам и характеризующая грамматику языка практически независимо от лексической информации.

Существенно, что в (а) и (б) фигурируют только правила вида "ПОСЛЕ того-то можно/нельзя/надо писать то-то", но не рассматриваются правила вида "ПЕРЕД тем-то то-то". Такое ограничение, как правило, совершенно не обременительное для лингвиста, существенно упрощает реализацию, делая проход по слову однонаправленным.

Рассмотрим теперь подробно структуру модели. Для облегчения понимания начнем с простейших и давно известных конструкций, постепенно переходя к более сложным случаям.

## 2.1. Простейший случай

Основную идею представления лингвистической информации в предлагаемой модели можно проиллюстрировать следующей общепринятой ([4, 1]) схемой, не требующей пояснений:


Рис.1. Простейшая морфологическая модель.

Алгоритм как анализа, так и синтеза очевиден. Заметим, что при анализе следует рассмотреть все возможные варианты составления входного слова из левой части, имеющейся в словаре, и правой части, имеющейся среди начертаний окончаний. Порядок и способ перебора не оговариваются - в этом программисту предоставляется полная свобода, от лингвиста же требуется только декларативная, но не алгоритмическая информация. Аналогично все возможные способы следует использовать при синтезе.

## 2.2. Таблица начертаний окончаний

Заметим теперь, что в словарь могут входить слова, имеющие различное начертание одинаковых по смыслу и во всех других отношениях окончаний (различия возникают, например, из-за чисто фонетических причин или вообще по капризам грамматики). Тогда в словаре следует каждому слову приписать номер того варианта набора начертаний, который ему соответствует.

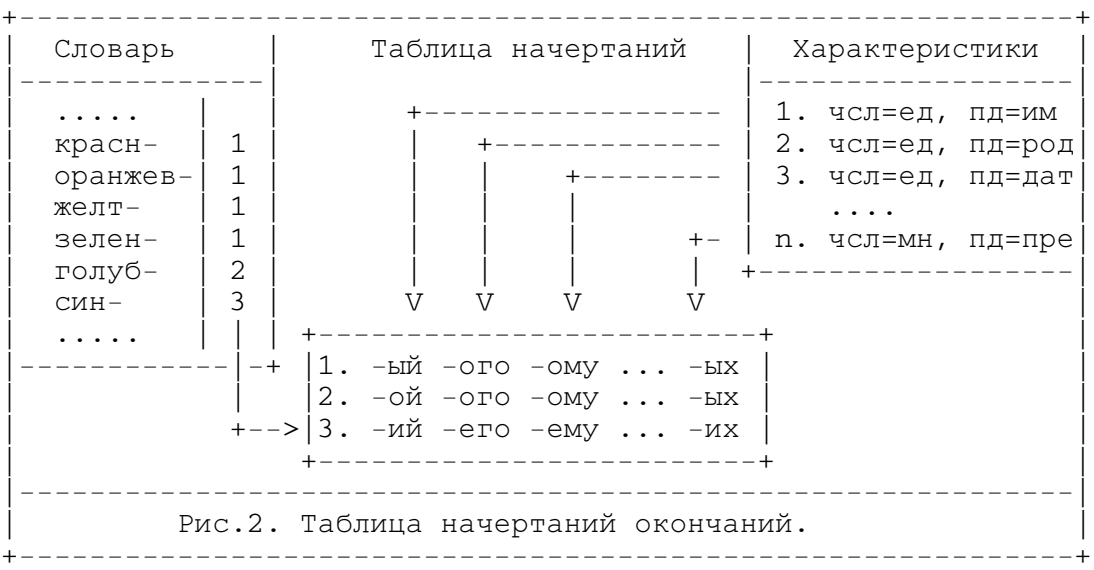



Рис.2. Таблица начертаний окончаний.

Прямоугольную матрицу из начертаний окончаний, столбцы которой занумерованы возможными наборами характеристик словоформ, а строки - различными вариантами правописания слов, мы назовем таблицей начертаний окончаний (ТН).

Оказывается, что для естественного языка, несмотря на кажущееся разнообразие вариантов правописания, обьем такой таблицы для всех слов, для которых ее вообще имеет смысл использовать, не превышает 20 - 30 столбцов и 60 - 100 строк.

## 2.3. Чередование основ и список масок

В приведенных выше примерах никак не учитывалось два

важных явления - супплетивизм и явления чередования в основах. Супплетивизмом называется наличие у одной лексемы в разных формах существенно различных основ, этимологически, как правило, происходящих от различных слов, слившихся в одно современное (при указании разбора слова на части мы будем обозначать звездочкой окончание, имеющее начертанием пустую цепочку букв: человек-*, человек-а, человек-у): человек-* - люд-и, ребенок-* - дет-и, ид-ти - ше-л. Покажем, как реализуется в предлагаемой модели этот случай. Введем понятие "маски" как строки знаков, длина которой соответствует длине строки таблицы начертаний. Каждый из этих знаков задает информацию о возможности для некоторого слова иметь данную форму (форму такого падежа, числа и т.п; более точно - о возможности для него выбрать данный столбец таблицы начертаний окончаний).

Кроме того, некоторые основы из хранящихся в словаре могут обьединяться в группы чередующихся основ, причем все основы из такой группы представляют одну и ту же лексему.

Проиллюстрируем сказанное примером. Номера масок припишем основам после номеров строк таблицы начертаний. На рисунке изобразим номером n с точкой перед основой тот факт, что она является n-й в группе чередующихся. Значки в масках имеют следующий смысл: `+' - разрешено, `-' - запрещено.

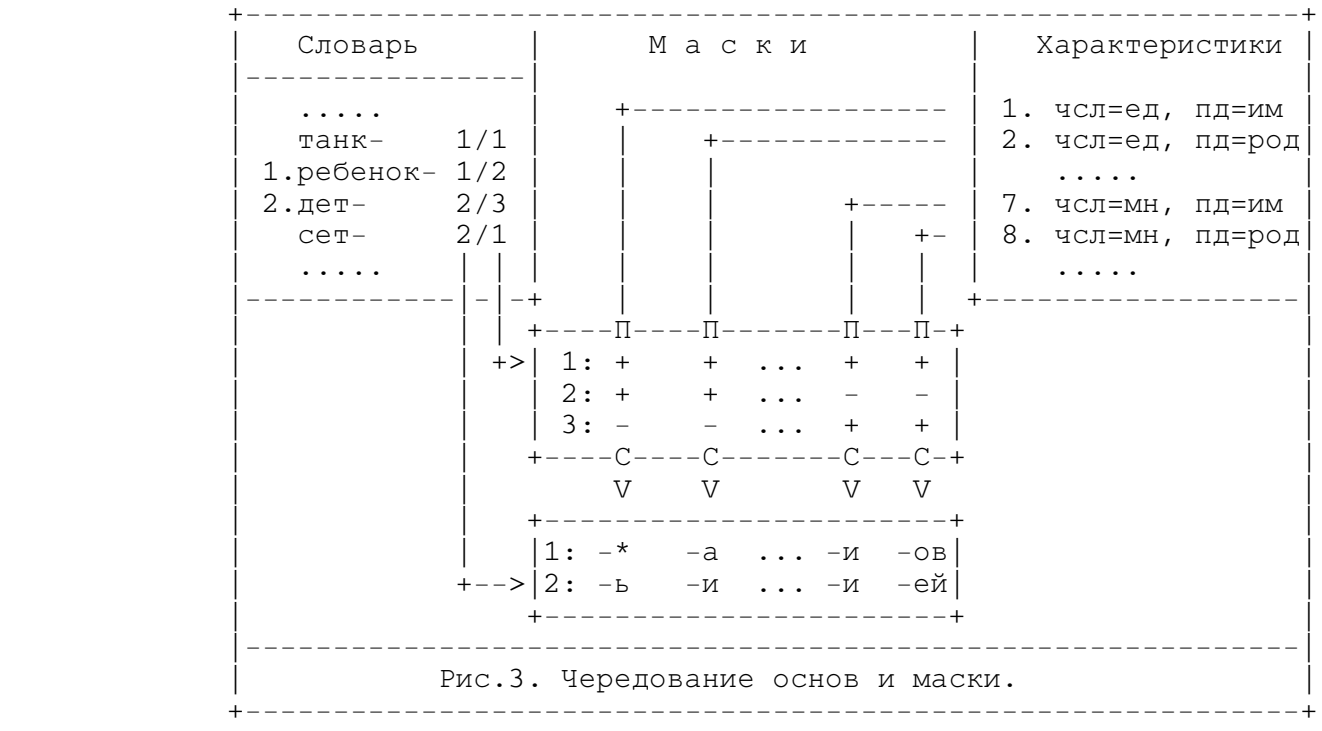



Рис.3. Чередование основ и маски.

Из рисунка видно, что основам слов "танк" и "сеть" разрешены все формы (`+' в маске), а основам "ребенок-" и "дет-"

знак `-' в маске закрывает (маскирует) возможность форм "*ребенки" и "*деть". Кроме того, видно, что основы ребенок- и дет- представляют одну лексему (так, если в качестве идентификатора лексемы используется число, то им соответствует одно и то же такое число - "лексема номер такой-то").

Кроме информации о разрешении/запрещении какой-либо формы для основы маски можно использовать для сообщения более подробной информации о форме: так, в выполненной автором реализации поддерживается значение `%', означающее затрудненность формы.

До сих пор мы не рассматривали явление чередования букв в основе: станок-* - станк-и, писа-ть - пиш-у, локальн-ый - локален-*. Например, в словаре [9] имеется более 9000 прилагательных, подобных прилагательным "локальный", "важный" (важен-*), что составляет около 10% словника. Однако в предлагаемой модели мы не будем никак отличать эти случаи от случая супплетивизма: поскольку механизм реализации супплетивизма все равно должен быть в алгоритме, он же может формально обслужить и все случаи чередования.

Третья функция механизма масок - запрещение слову некоторых форм, если они запрещены ему не из-за какого-то чередования или правила, а просто по капризу грамматики. Так, можно сказать "я рад", "она рада", но это слово почему-то не может иметь полных форм ("я *радый").

## 2.4. Случай нескольких морфем

Рассмотрим теперь слова, в которых после основы выделяется не одна, как, например, в слове "программ-а", а несколько частей, сравнительно произвольно сочетающихся друг с другом:

| | |
|---|---|
| важн-*-ый | важн-ейш-ий |
| важн-*-ыми | важн-ейш-ими |
| чита-ющ-ий-* | чита-ющ-ий-ся |
| чита-ющ-ими-* | чита-ющ-ими-ся |
| чита-ем-ый-* | |
| чита-ем-ыми-* | |

Будем называть части, на которые делятся слова, "морфемами" (этот термин не выдерживает никакой критики со стороны лингвиста, но автору неизвестен никакой подходящий лингвистический термин, кроме совсем уж неудобного в данном изложении термина "порядок"), и говорить, что приведенные выше словоформы имеют соответственно по 3 и 4 морфемы. Будем нумеровать

морфемы начиная с 0, где 0-ая морфема есть основа словоформы, а морфемы 1, 2, ... - окончания (мы не будем отвлекаться на различение по названиям суффиксов, окончаний, постокончаний, частиц, т.к. для изложения это несущественно). В предыдущем изложениии мы игнорировали тот факт, что приведенные на рис. 1 и 2 прилагательные на самом деле имеют не 2, а 3 морфемы.

В этом случае построим все упоминавшиеся таблицы для каждой морфемы, кроме морфемы 0, например, для прилагательных:

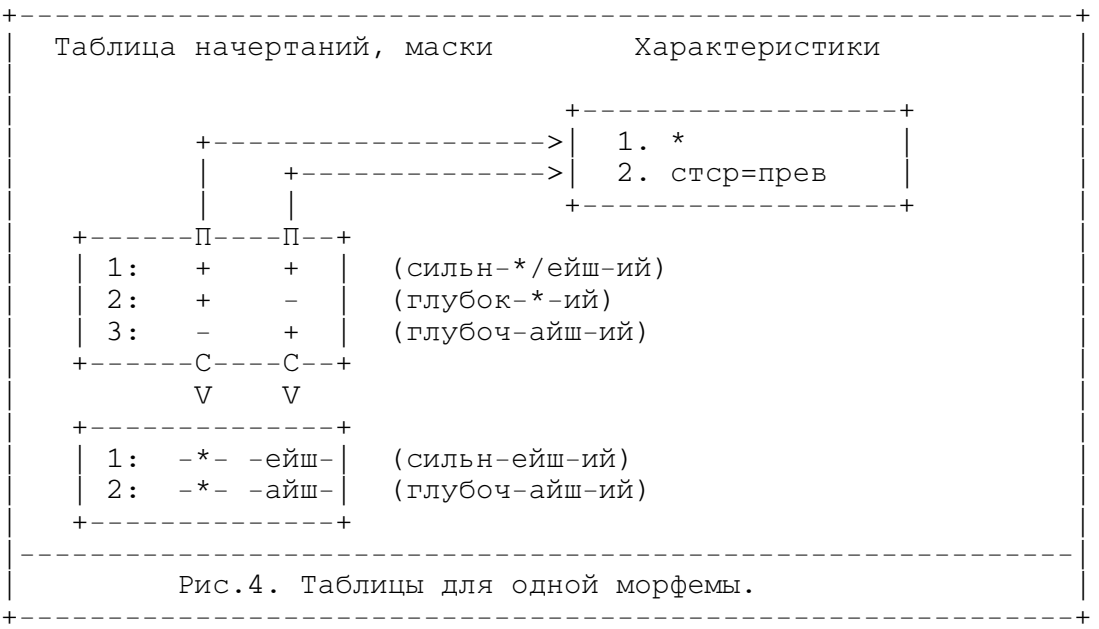
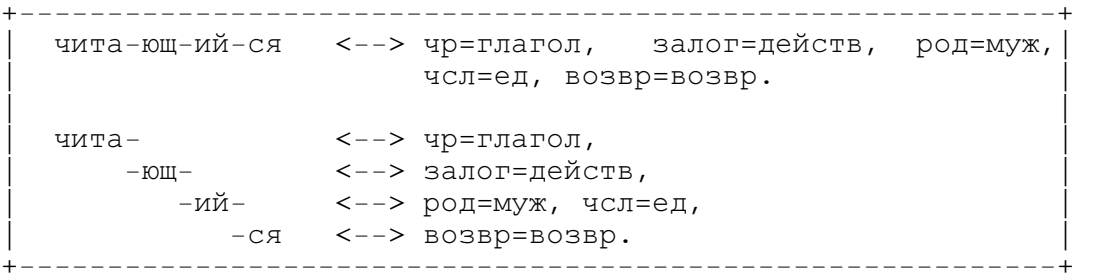


Рис.4. Таблицы для одной морфемы.

(через * обозначен пустой вклад в общий набор характеристик).

Оказывается, что в естественных языках удается не только достаточно однозначно соотнести целую словоформу с набором характеристик (что, собственно, и является задачей морфологического анализа/синтеза), но и отдельные морфемы словоформ соотнести с их вкладом в общий набор характеристик:




Собственно, такое соответствие и является предметом науки морфологии. Возможность получения от лингвиста декларативной

информации о таком соответствии является одной из ключевых идей предлагаемой модели. Именно за счет такого раздельного анализа морфем удается резко сократить количество используемых в модели правил.

Более того, именно такое представление знаний позволяет строить на основе предлагаемой модели программные реализации с элементами искусственного интеллекта. Действительно, именно такую модель морфологии использует при формообразовании слов человек, причем тем более формально, чем меньше его навык в устной речи на данном языке. Обладая адекватной моделью морфологии, программа сможет не только распознавать и строить правильные словоформы, но и правильно (т.е. так же, как человек) понимать ошибочно построенные словоформы и даже предлагать варианты их исправления. Так, программой будут правильно поняты и исправлены словоформы вроде "*пишат" вместо "пишут", "*красний", "*краснейшый" и т.п. Автор затрудняется сказать, как будет страдательное причастие от "пить", но если ему придется обьяснять роботу, снабженному лингвистическим процессором с данной морфологической компонентой, что вода "*пьемая" или "*пьимая" или "*пиемая", то он будет понят правильно. Эта особенность морфологического анализатора позволит в подобных системах применять упрощенный синтаксический анализатор (что является значительной экономией), не предусматривающий специальной обработки случая выражения "вода, которую пьют" (кто?) или "вода, которую можно пить".

## 2.5. Управление начертаниями между морфемами

Оказывается, что для каждой морфемы все предыдущие морфемы играют роль, подобную роли основы, а все последующие подчиняются ей аналогично тому, как в простейшем случае выбор начертания подчинен основе. Поясним эту мысль. На рис. 2 мы видели, как начертание окончания в одной и той же форме зависит от индивидуальных свойств основы: красн-*-ый, но син-*-ий, голуб-*-ой. Заметим теперь, что если для первой морфемы выбрано начертание "- ейш-", то выбор начертания из приведенных альтернатив полностью определен: правильное написание - "...-ейш-ий" независимо от основы. Здесь окончание (суффикс) -ейш- полностью играет роль основы для окончания -ий. В соответствии с этой идеей составим аналог словаря основ для морфемы 2 (-ий), состоящий из всех возможных начертаний морфемы 1 (-ейш-). В этом словаре так же как и на рис. 2 припишем каждому начертанию номер строки таблицы начертаний для следующей морфемы, используя номера строк таблицы для морфемы 2 с рис. 2 и начертания суффиксов из таблицы для морфемы 1 с рис. 4:

| Начертание | Номер_строки_морфемы_2 | Пример |
|---|---|---|
| -айш- | 3 | ближ-айш-ий |
| -ейш- | 3 | сильн-ейш-ий |
| -*- | * | син-*-ий, |
| | | красн-*-ый. |

Рис.5. Список начертаний окончаний.

Из рисунка видно, что, независимо от номера строки, указанного в словаре, номер строки ТН морфемы 2 полностью определен выбором начертания -ейш- для морфемы 1. Напротив, некоторые начертания (обычно это -*-) не могут управлять в указанном смысле выбором номера строки для следующей морфемы. В этом случае в графе номера строки ставится специальное значение *, означающее, что номер следует взять из таблиц для предыдущей морфемы (если это морфема 0 - то из словаря основ). Заметив, что теперь в словаре на рис. 2 основам будет приписан не один номер строки, а два - один для -айш-/-ейш-, а другой для -ый/-ой/-ий, приходим к следующим соотношениям (на примере синтеза):

```
На входе : красн- (т.е. номер такой-то), стср=*, пд=им.
Морфема 0: красн
Морфема 1: строка 1 (от морфемы 0), столбец 1: -*-
Морфема 2: строка 1 (от морфемы 0), столбец 1: -ый
Получили : красный.
```

```
На входе : красн- (т.е. номер такой-то), стср=прев,пд=им.
Морфема 0: красн
Морфема 1: строка 1 (от морфемы 0), столбец 2: -ейш-
Морфема 2: строка 3 (от морфемы 1), столбец 1: -ий
Получили : краснейший.
```

Точно так же действует механизм управления начертаниями в случае большего числа морфем: в этом случае в списке начертаний i-й морфемы (в словаре при i=0) при каждом начертаниии будет указано по одному номеру строки для каждой последующей морфемы, и действующей для каждой морфемы будет строка, выбираемая для нее самой ближней левой морфемой, у которой это значение отлично от *.

## 2.6. Взаимосвязь форм

Мы видим, что ключевое значение во всех наших рассмотрениях имеет номер столбца таблицы начертаний окончаний (для данной морфемы), он же номер элемента маски, он же номер набора характеристик, приписываемого словоформе при выборе дан-

ного столбца. Назовем этот номер "формой" (что тоже, конечно, не выдерживает никакой критики). Итак, мы будем говорить, что слово красн-ейший имеет форму номер 2 для морфемы 1 (-ейш-) и форму номер 1 для морфемы 2 (-ий) (см. рис. 2 и 4). Однако упоминать формы мы будем не по номерам, а - для удобства - по характерным окончаниям из соответствующего столбца таблицы начертаний ("форма -ейш-") или по характеристикам.

Взаимосвязь форм заключается в несовместимости (невозможности употребления в одной словоформе) или, редко, в затрудненности совмещения этих форм. Так, основа чита- может иметь как отдельно форму залог=страд, соответствующую -ем (чита-ем-ый-*), так и отдельно возвратную форму -ся (чита-ющ-ий-ся), однако не может иметь сочетание форм *чита-ем-ый-ся.

Однако оказывается, что в естественном языке всегда удается так подобрать набор форм, что такие запреты (а) являются свойствами самих форм, а не индивидуальными свойствами слов (так, *...-ем-ый-ся запрещено для всех слов), и (б) касаются только пар форм, а не более сложных их сочетаний (в случае невыполнения последнего всегда можно разбить одну форму на несколько по-разному сочетающихся с другими, причем в естественном языке это обычно вполне соответствует лингвистической неоднозначности формы).

Таким образом, каждой форме (столбцу таблицы начертаний) можно сопоставить по одной маске на формы каждой из последующих морфем. Лингвисту удобно понимать такую запись как "данная форма разрешает или запрещает такую-то".

Дадим теперь некоторую поправку к предложению (б), сформулированному выше. Может встретиться ситуация, когда для одной и той же формы некоторой морфемы имеется разрешение от одной формы (другой морфемы) и запрещение от другой формы (третьей морфемы). Примем следующее естественное соглашение, аналогичное соглашению для начертаний: действующим будем считать ближайшее (слева) значение. Такое соглашение вполне соответствует тенденции естественного языка ставить текстуально рядом семантически связанные морфемы.

Нам снова понадобится специальное значение (обозначим его `*') для элемента маски, означающее отсутствие управляющей информации. Таким образом, в списке масок возможны, например, маски, содержащие `-' для запрета некоторых форм и `*' для отсутствия запрета или разрешения других форм. Как и раньше, значение `*' будет игнорироваться при нахождении ближайшего управляющего значения. Кроме того, мы будем использовать следующее сокращение: номер маски "***...*", состоящей

из всех `*', будем обозначать *.

Рассмотрим действие данного механизма на примере. Дополним список форм на рис. 4 номерами масок для морфемы 2 (в словаре номера строк и масок для одной морфемы будем писать через косую черту; все номера для большей наглядности сделаем различными):

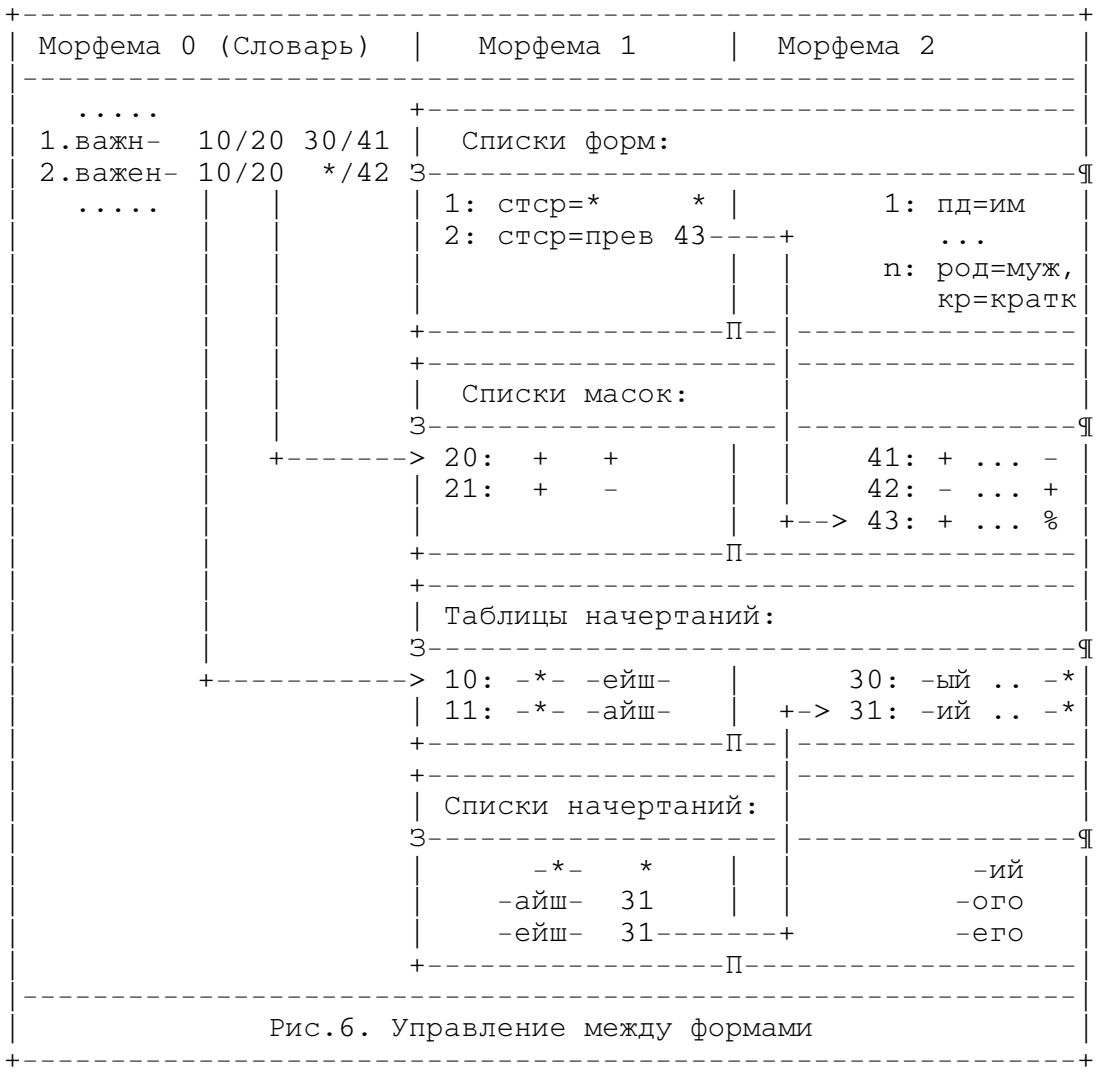



Рис.6. Управление между формами

По рисунку легко проверяется возможность построения словоформы важн-*-ый (формой 1 (-ый) морфемы 2 управляет морфема 0 - словарь, поскольку форма 1 (-*-) морфемы 1 не берет на себя управление), возможность построения затрудненной краткой словоформы (мы оставляем за пределами обсуждения лингвистическую спорность данного конкретного примера) важн-ейш-* (формой n морфемы 2 управляет форма 2 морфемы 1, выбирающая для нее маску 43), и в то же время невозможность построения словоформы *важн-*-*.

Изложенный механизм играет важную роль, например, при описании русских причастий.

## 2.7. Части речи и словарь основ

До сих пор мы пользовались двумя вариантами таблиц - как на рис. 6, с одной стороны, и как на рисунке 3, с другой. При этом мы рассматривали совершенно одинаково устроенные таблицы и алгоритмы, различающиеся (а) количеством морфем, (б) конкретным заполнением.

Будем теперь считать, что все слова языка делятся на небольшое число непересекающихся классов слов с одинаковыми законами формоизменения. В этом случае построим для каждого такого класса свой набор таблиц (кроме словаря основ) так, как это излагалось выше. Таким образом мы получим несколько различных реализаций лингвистической модели, как будто мы имеем дело с несколькими совершенно разными языками. Экзотическим применением такого подхода была бы система, настроенная на анализ/синтез текстов, содержащих смесь слов различных языков. Программная оболочка для излагаемой модели не потребует в этом случае никаких изменений.

Назовем такой класс одинаково изменяющихся слов "технической частью речи". Дело в том, что как правило технические части речи хорошо соответствуют известным грамматическим частям речи. Однако это соответствие не всегда удобно сохранять. Так, например, возможно, что лингвист пожелает отнести слова адьективного склонения типа "булочная", "хорунжий" к одной технической части речи с прилагательными; слова типа "учащийся" - с глаголами (причастиями); все неизменяемые слова (наречья, предлоги, союзы) будет удобно отнести к одной технической части речи. Собственно грамматическая часть речи является при этом словарной информацией, хранение которой реализуется в точности тем же механизмом, как для форм. Таким образом, словарь играет в некотором смысле роль таблицы начертаний и списка начертаний для морфемы 0.

Мы будем называть технические части речи не только по номерам, но и (в чисто мнемоническом смысле) 0 - "неизменяемые", 1 - "существительные", 2 - "прилагательные", 3 - "глаголы", поскольку в примерах будут употребляться именно такие варианты заполнения таблиц. Конечно, количество и смысл технических частей речи и количество морфем в каждой для программной оболочки является параметром и задается лингвистом.

## 2.8. Список форм

В завершение уточним, чему соответствует в предлагаемой модели номер столбца таблицы начертаний, он же номер элемента

маски, который мы не очень удачно назвали формой. Известно, что в языке бывают такие наборы морфологических характеристик, что словоформы, соответствующие этим наборам, никогда не различаются по написанию и могут быть различены исключительно по смыслу.

Таковы, например, для русских существительных наборы "пд=им, род=муж, чсл=ед, од=неодуш" ("стоит станок") и "пд=вин, род=муж, чсл=ед, од=неодуш" ("вижу станок"), для прилагательных - наборы "род=жен, чсл=ед, пд=род/дат/твор/предл" ("нет белой бумаги", "к белой бумаге", "белой бумагой", "о белой бумаге").

В этом случае нет никакого смысла различать их до тех пор, пока речь не идет собственно о наборе характеристик словоформы. Поэтому допускается возможность поставить несколько наборов характеристик в соответствие одному столбцу таблицы начертаний (форме). Окончательно получаем следующий формат для списка форм:

```
+-------------------------------------------------------------+
|  1. (NN масок след. морфем) - пд=им,род=муж,чсл=ед          |
|                             - пд=вин,род=муж,чсл=ед,        |
|                               одуш=неод                     |
|                                                             |
|  2. (номера масок...)       - пд=им,род=сред,чсл=ед         |
|                             - пд=вин,род=сред,чсл=ед        |
|                                                             |
|  3. (номера масок...)       - пд=род,род=муж,чсл=ед         |
|                             - пд=вин,род=муж,чсл=ед,        |
|                               одуш=одуш                     |
|                             - пд=род,род=сред,чсл=ед        |
|                                                             |
|-------------------------------------------------------------|
|                      Рис.7. Список форм.                    |
+-------------------------------------------------------------+
```

### 2.9. Связь с теорией И.А.Мельчука

Во время своего недавнего (1989) визита в СССР крупнейший русскоязычный лингвист И.А.Мельчук на лекции в МГИАИ рассказал о своих взглядах в области морфологии, которые автор понял следующим образом.

Морфология - наука о строении слов. Рассмотрим, что такое слово. Слово - это соединение трех сущностей:

<смысл; знак; синтактика>.

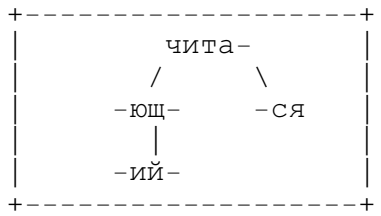
Смысл - то, что означает слово, то есть та информация, которая привносится этим словом в предложение. Знак - тот

элемент изображения, который дает возможность упоминуть слово в надлежащем месте предложения. Синтактика - правила употребления слова, связь его с другими словами, влияние его на употребление и формы других слов предложения.

Теперь заметим, что в рамках изложенной автором модели морфологии можно провести аналогию между строением предложения, состоящего из отдельных слов, с одной стороны, и строением слова, состоящего из отдельных морфем, с другой. Особенно четко такая аналогия просматривается при переводе с синтетического флективного языка на аналитический язык: "книга Пети" - "book of Peter", "книгой" - "by book".

Сопоставим морфологическое описание отдельного слова по И.А.Мельчуку с описанием в рассматриваемой модели отдельной морфемы. В этом случае можно усмотреть следующую аналогию: набор знаков такого мини-языка - это начертания из списка начертаний; набор смыслов, привносимых "словами" в целое "предложение" (слово) - это наборы характеристик из списка форм. Синтактика же представляется в виде таблиц начертаний и информации об управлении по номерам строк и масок.

Заметим в заключение, что истинная схема связи морфем в слове могла бы с учетом данной аналогии представляться, например, для русского глагола следующим графом:




с очевидными смыслами в узлах (почерпнутыми из списка форм). Именно с учетом такого графа должно было бы быть устроено управление совместимостью форм (но не начертаний!). Однако автор счел возможным предложить лингвисту для заполнения модель с полным графом "все со всеми" морфем, что упрощает формат таблиц.

## 2.10. Исправление ошибок

Как видно из структуры модели, при анализе будут выдаваться сообщения по следующим трем типам ошибок:

1. Несовместимые формы ("*чита-ем-ый-ся");
2. Неправильная основа ("*глубоч-ий", "*человек-и");
   Недопустимая форма ("*рад-ый");
3. Неверное начертание ("*красн-ий").

+------------------------------------------------------------+

При этом система в двух последних случаях сможет даже выдать некоторый совет по исправлению такой ошибки (путем синтеза по полученной полной форме). Однако, конечно, задача исправления ошибок принадлежит скорее к психологии или школьной статистике, чем к морфологии, автор же не ставил перед собой цели всерьез отнестись к этой задаче (так, по слову "буд-*-те" вместо "буд-ь-те" система, вероятнее всего, выдаст ошибку "несовместимые формы"). Главное то, что пользователь получит указание на ошибку, однако и совет системы по ее исправлению в подавляющем большинстве случаев будет полезен.

## 3. РЕАЛИЗАЦИЯ

### 3.1. Общие характеристики

Программное обеспечение системы реализовано в среде MS-DOS, а также в среде ОС Демос-86 (UNIX, VENIX, XENIX), с соблюдением полной UNIX-совместимости, на ПЭВМ типа PC с памятью 640 К (процессор типа Intel 8088) и жестким диском 20 М, и в среде ОС Демос (UNIX) на СМ-1420. Язык реализации - стандарт Си (K&R). Потребность в оперативной памяти не превышает 64 Кбайт, дисковой - ок. 2 Мбайт. Быстродействие на различных ПЭВМ колеблется от 8 слов/секунду (IBM PC, PC XT) до 30 слов/секунду (PC AT), однако автор планирует улучшить этот показатель в 2-3 раза.

Система снабжена комплексом сервисных процедур для пополнения таблиц и автоматизации формирования и пополнения словаря по тексту любого словаря, подобного [9].

### 3.2. Дисциплина доступа к словарю

Организация словаря является ключевым моментом в реализации всякой подобной системы, поскольку от эффективности доступа к нему зависит эффективность работы всего комплекса. Поэтому из всех многочисленных применяемых в системе эвристик и оптимизаций вычислительного процесса рассмотрим только организацию доступа к словарю.

Существует два варианта порядка разбора текста словоформы: "начиная с окончания" и "начиная с основы". В настоящее время организация "начиная с окончания" применяется в большинстве подобных систем. В этом случае сначала всевозможными способами от слова отделяются окончания, а затем с каждым вариантом основы (известной длины) происходит обращение к СУБД словаря. Такой подход, вполне приемлемый для систем проверки

правильности текста, где перебор можно прекратить при обнаружении первого же правильного варианта, малоприемлем для целей точного анализа с полным различением омонимии.

Насколько известно автору, для русского языка имеется в настоящее время только три системы морфологического анализа: [1, 2, 4]. При этом в системе [1], например, принято решение проводить анализ по словарю только для первого варианта разбора. Это дает хорошее быстродействие, но приводит, например, к тому, что в словосочетании "несколько электрических полей" система обнаруживает только глагол "полить" (что делает невозможным дальнейший комбинаторный или синтаксический анализ). В настоящей разработке применен метод анализа "начиная с основы": из входного текста выделяются все начальные цепочки, имеющиеся в словаре основ, и затем производится попытка разобрать остаток слова на окончания. Попутно такой способ дает возможность хранить в словаре цепочки, содержащие знаки препинания (такие как "г." - год и город, "летчик-космонавт", "так, как" и др.).

Однако такой подход приводит к невозможности использования традиционных методов построения СУБД. Реализованная автором СУБД словаря предназначена для обработки следующего запроса: выдать все записи, ключи которых являются начальными подцепочками предъявленной цепочки (достаточно большой длины).

Следующий пункт посвящен обсуждению архитектуры такой СУБД. Заметим, что эта СУБД не предназначена для заполнения в диалоге и подразумевает сравнительно ресурсоемкую операцию начального формирования базы и добавления новых записей. Однако для наших целей это вполне приемлемо.

### 3.3. Организация словаря

К упомянутой ранее СУБД предъявляется два основных требования: (а) считывание информации в память с внешнего носителя должно происходить отрезками порядка 0.5 килобайта из-за фрагментированности файлов в современных ОС, и (б) минимизация количества обращений к диску для поиска с одним словом.

Ввиду этого будем считать, что словарь разделен на блоки размером порядка 0.5 килобайта. Таких блоков при обьеме словника 100 тысяч слов будет около 3000. Единственный известный автору способ быстрого ориентирования среди них - алфавитное упорядочение ключей и хранение в памяти оглавления - информации о первой записи каждого блока (ее удается существенно скомпактировать). Хеширование не подходит уже хотя бы из-за неизвестности длины ключа. Алгоритм поиска очевиден: находим алфавитное место входного текста в словаре, затем урезаем

текст по первому несовпадению с предыдущим (найденным) ключем и повторяем поиск снова. Процесс обязательно остановится. Мы не будем останавливаться на применяемых автором оптимизациях данного алгоритма.

Однако в этом случае возникает следующая проблема. При алфавитном поиске, скажем, слова "пары" оно будет найдено после всех слов "паровоз", "пароход" и т.п. Однако истинная основа слова "пары" - пар- (пар-а, пар-у). Она будет найдена при повторном поиске урезанного слова пар-. Однако слову "пар" соответствует блок, отстоящий (по измерениям автора) на 10 блоков от первоначального. Второе: для слов, допускающих неоднозначный разбор, как упоминавшееся слово "полей", различные основы ("поле-й" глагол и "пол-ей" существ.) могут находиться в разных блоках.

Для разрешения этой ситуации автор предлагает следующее решение, сыгравшее ключевую роль в эффективной реализации всей стратегии разбора "слева направо" при декларативном представлении грамматической информации в виде правил типа "ПОСЛЕ того-то пишется то-то". Следует все слова, для которых рассмотренная выше ситуация - первый поиск дает не тот блок - возможна, продублировать во все блоки, где они могут быть найдены при первом обращении. Таким образом эффективность обращения к диску станет предельной - одно обращение на слово, поскольку в первом же считанном блоке имеется информация для всех начальных подцепочек исследуемого слова.

Рассмотрим, насколько увеличится словарь при таком дублировании. Из алгоритма лексикографического поиска видно, что истинная основа слова является всегда начальной подцепочкой того ключа, который будет найден при первой итерации поиска. Таким образом, следует просто в каждый блок словаря сдублировать все записи с ключами, являющимися начальной подцепочкой ключа какой-либо записи, находящейся в данном блоке.

Рассмотрим лексикографически упорядоченное конечное множество W слов в конечном алфавите ("словарь основ"). Обозначим через F (a) для a из W множество слов из W, являющихся начальными подцепочками слова a. Для подмножества U слов из W обозначим через F (U) обьединение всех F (u) по u из U. Введем также обозначение Ф (U) = F (U) \ U (разность множеств). Множество всех слов из W, расположенных по алфавиту между словами a и b из W, назовем отрезком [a,b].

В нашем случае блок словаря как раз является некоторым отрезком [a,b], а множество словарных статей, которые придется сдублировать в него - как раз Ф ([a,b]). Легко доказать следующий замечательный для данного приложения факт: для любого отрезка [a,b] имеет место равенство

$$\Phi([a,b]) = \Phi(a).$$

Таким образом мы получили, что, независимо от размера блока словаря, в него придется сдублировать всего несколько слов (автор наблюдал максимум 3 для слова "паровозный"). Увеличение размера словаря составляет менее 10% первоначального обьема (в текущей реализации блок имеет размер 0.5 килобайт и вмещает 25 - 35 слов).

Существует быстрый однопроходный алгоритм формирования блоков с сдублированными в них вложенными основами, использующий тот факт, что во время просмотра упорядоченного текста словаря можно хранить для каждого текущего слова все его вложенные основы в стеке, глубина которого ввиду вышедоказанного не превышает максимальной длины одного слова. Кроме того, для упорядоченных ключей автором используется экономичный метод хранения: для каждого следующего ключа хранится только его отличная от предыдущего ключа часть. Алгоритмы обращения с таким массивом также однопроходны и нересурсоемки.

## Литература


1. Белоногов Г.Г., Зеленков Ю.Г. Алгоритм морфологического анализа русских слов // В сб.: Вопросы информационной теории и практики. М.: 1985, N 53.
2. Мальковский М.Г. Диалог с системой искусственного интеллекта // М.: МГУ, 1985, 216 стр.
3. Перцова Н.Н., Перцов Н.В. Проект лингвистического процессора для базы знаний РЕЗОН // В сб.: Семиотические аспекты формализации интеллектуальной деятельности. М.: ВИНИТИ, 1988.
4. Апресян Ю.Д. и др. Лингвистическое обеспечение системы автоматического перевода Этап-2 // М.: Наука, 1989.
5. Орфоэпический словарь русского языка под редакцией Р.И. Ованесова // М.: Русский язык, 1985.
6. Экспертные системы. Принципы работы и примеры. Под редакцией Р. Форсайта // М: Радио и связь, 1987.
7. Добрушина Е.Р., Савина Г.Б. Морфологический анализ и синтез: информационно-поисковый подход // В сб.: Семиотические аспекты формализации интеллектуальной деятельности. М.: ВИНИТИ, 1988.
8. Добрушина Е.Р., Савина Г.Б., Гельбух А.Ф. Система точного морфологического анализа и синтеза // В сб.: Программное обеспечение новой информационной технологии. Калинин: АН СССР, НПО ЦПС, 1989.
9. Зализняк А.А. Грамматический словарь русского языка. Словоизменение // М.: Русский язык, 1980, 880 стр.