ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ПРИКЛАДНАЯ МАТЕМАТИКА

УДК 004.89

А.К. Розанов

ОРГАНИЗАЦИЯ СЛОВАРЯ В СИСТЕМАХ ГЕНЕРАЦИИ И ОПРЕДЕЛЕНИЯ ФОРМ СЛОВ ЕСТЕСТВЕННЫХ ЯЗЫКОВ

Рассмотрены основные существующие подходы к организации словарей в системах генерации и определения форм слов естественных языков. Предложена иерархическая модель организации словаря, показаны её сильные стороны и преимущества, даны рекомендации по выбору модели словаря при проектировании систем генерации и определения форм слов естественных языков.

Ключевые слова: *преобразования строк, анализ форм слов, автоматическая обработка текста, структуры данных, иерархические модели.*

Введение. Задача определения форм слов естественного языка (получения грамматической информации для каждого из слов текста) и связанная с ней естественным образом обратная задача (генерация слов, обладающих требуемой грамматической информацией) – это задачи, возникающие в целом ряде практических вопросов языкознания. Примерами областей применения генераторов и анализаторов форм слов могут служить такие области знаний, как извлечение знаний (англ. data mining), статистическая обработка текста, построение экспертных систем, поддерживающих общение на естественном языке, поисковых систем, интеллектуальных электронных справочников и словарей, машинный перевод и другие задачи.

В работах [4, 5] предложен семантико-комбинаторный метод анализа слов в некотором алфавите при образовании таких слов из алфавита и слов меньшей длины по определенным правилам. В этой связи интересен вопрос об аналогичном комбинаторном подходе при морфологическом или синтаксическом анализе слов. Задача определения формы слова (или, иначе говоря, грамматической информации, соответствующей слову) необязательно имеет однозначное решение. Неоднозначности результатов анализа форм слов могут быть как следствиями недостаточной или некорректно заполненной базы данных о языке, так и омонимией в тексте (к примеру, без контекста не всегда удаётся установить часть речи, особенно в неполных предложениях).

Учитывая это, в рамках данной статьи под задачей определения формы слова будем понимать задачу получения всех возможных вариантов *трактовки* этого слова, оставляя задачу разрешения омонимии на более поздние этапы обработки текста.

Подходы к решению задачи определения форм слов могут быть как основанными на словарях [1-3,6,7], так и не использующими создаваемые человеком словари [8,9]. В рамках данной статьи будем рассматривать возможные варианты организации словарей в системах, использующих создаваемые человеком словари.

Для краткости термином «словарь» в настоящей статье будет обозначаться вся совокупность данных о конкретном языке, необходимых для выполнения анализа форм слов этого языка, за исключением случаев, когда характер словаря указан непосредственно (например, «словарь основ» или «словарь окончаний»).

Цели работы

1. Создание такой модели организации словаря в системе генерации и определения форм слов, которая, во-первых, была бы достаточно гибкой для описания формообразования в языках со сложной морфологией (например, для русского языка), во-вторых, универсальной (не нацеленной на конкретный язык), в-третьих, обеспечивающей удобство наполнения данными для инженера по знаниям.

2. Описание структуры формообразования синтетических словоформ русского языка с использованием созданной модели.

Для достижения этих целей в данной работе **ставится задача** структуризации словаря, то есть задача выбора способа эффективного представления используемых исходных данных о формообразовании конкретного языка.

1. Хранение правил преобразования слов без структуризации. Изначальная реализация алгоритмов в работе [2] не предполагала дополнительной структуризации множества правил преобразований начальных форм (цепочек).

Цепочка C – это последовательность элементарных преобразований строки (элементарной операцией является присоединение либо отделение префикса либо постфикса) такая, что существует одна, и только одна *обратная* последовательность C' такая, что последовательное применение C и C' к любой строке S, к которой применение C вообще возможно, не изменяет исходную строку S.

Анализ формы слова естественного языка в этом случае заключается в выборе набора цепочек, обратные к которым применимы к рассматриваемому слову, с последующим отсевом тех, которые не сопоставлены в таблице полученным начальным формам. Результатом анализа является набор пар {*полученная применением обратной цепочки начальная форма; информация, приписанная применённой цепочке*}, сформированный из множества найденных цепочек. При анализе применялась лишь таблица соответствий цепочек группам основ (либо начальных форм слов), имеющих одинаковые наборы окончаний.

Очевидным недостатком такого подхода является крайняя затруднительность процесса начального заполнения словаря и его последующего поддержания в актуальном состоянии, поскольку для каждого нового слова, не принадлежащего никакой группе, требуется сначала определить структуру парадигмы, и лишь затем – перечислить все его возможные формы.

Другим недостатком такой организации словаря является то, что отсутствие иерархии в представлении типов слов языка влечёт трудности визуализации словаря. Представление всего словаря для языка в виде неструктурированного множества групп слов с одинаковыми наборами окончаний воспринимается пользователем хуже, чем строгая иерархия типов и подтипов с небольшим количеством потомков у каждого из узлов дерева.

Стоит также отметить тот факт, что интеграция организованного подобным образом модуля в некоторую конкретную информационную систему потребует от этой системы собственной модели для представления морфологической информации, поскольку результаты анализа передаются не в структурированном виде.

2. Таблицы основ, окончаний и вспомогательные таблицы. В работе [6] для построения анализатора форм слов русского языка предлагается весьма компактная структура словаря, включающая таблицу основ (сгруппированных по флективным классам), таблицу окончаний, вспомогательные таблицы, кодирующие грамматическую информацию.

Хотя предложенная структура словаря вполне подходит для анализа слов русского языка, алгоритмы и таблицы нацелены исключительно на русский язык, и не являются масштабируемыми. В данной же работе ставится целью создание модели организации данных, не привязанной к конкретному естественному языку.

3. Словари, содержащие все словоформы. По существу, всякая система, способная к генерации и определению форм слов естественных языков, может быть использована для построения скоростного анализатора форм слов, использующего полный словарь всех словоформ, построенный автоматически путём их генерации из всех известных слов словаря.

Такая система будет обладать высоким быстродействием [10], но, в то же время и большими требованиями к объёму памяти. В случае когда аппаратные средства обладают большим количеством доступной памяти и требуется высокая скорость анализа, использование такой системы является предпочтительным.

Вместе с тем, словарь всех словоформ в большинстве случаев (а с учётом огромного количества словоформ в естественных языках – практически во всех возможных) создаётся автоматически на основе иных, базовых словарей системы, поэтому вопрос организации словарей систем анализа и генерации форм слов ни в коей мере не утрачивает актуальности в связи с этим обстоятельством.

4. Словари со структурированной грамматической информацией. В описанных выше подходах к организации словарей само понятие грамматической информации не подвергалось структуризации либо на структуре не был сделан акцент (такие системы либо не рассматривали результат анализа «существительное в единственном числе родительного падежа» как сложный объект [2]), за исключением ориентированной на русский язык системы организации словаря, описанной в книге [6].

В системах же, рассчитанных на частые модификации словаря пользователем, требуется максимальная гибкость структуры словаря и максимальное удобство выполнения типовых операций над ним (добавление новых слов и

групп слов), и грамматическую информацию приходится структурировать, выделяя в результатах анализа класс основы и совокупность значений, соответствующих грамматическим категориям. В таких системах отдельными сущностями являются грамматические категории («род», «число») и их значения («средний род», «множественное число»).

По этим причинам предлагаемая модель подразумевает многоуровневую структуризацию словаря. Далее следуют определения и понятия, необходимые для полного описания модели представления структурированного словаря, лишённого описанных выше недостатков, после чего описываются основные сущности словаря, их структура и иерархия.

5. Элементарные компоненты грамматической информации. В рамках предлагаемой модели используется понятие грамматической категории, или грамматического признака, как основного структурного элемента в представлении грамматической информации.

Термин «грамматическая категория» определяется как система противопоставленных друг другу рядов грамматических форм с однородными значениями [11].

Другими словами, грамматическая категория – это такая совокупность альтернатив, что для всякого слова, для которого эта категория имеет смысл, возможно не более одного значения из множества альтернатив. Сами альтернативы будем называть **граммемами** [11].

Примерами грамматических категорий русского языка могут служить категории числа (с граммемами «единственное» и «множественное») и рода («мужской», «женский», «средний»).

Грамматические категории в рамках некоторого класса слов могут быть постоянными (род существительного, вид глагола) и переменными (падеж и число существительного, лицо, род и число личной формы глагола), в зависимости от того, характеризуется ли одной и той же граммемой каждое отдельное слово этого класса вне зависимости от формы, или нет.

Одна и та же грамматическая категория может быть постоянной для одного класса слов и переменной – для другого (так, род является постоянным для имён существительных, но переменным для прилагательных и личных форм глаголов).

Значения некоторых грамматических категорий в рамках конкретного класса слов могут влиять на допустимость значений других категорий (невозможность построить страдательное причастие прошедшего времени у глагола несовершенного вида) или даже на применимость самих категорий (для прилагательного во множественном числе род не определяется), однако это не всегда так. Например, для любой комбинации числа и падежа можно указать существительное, находящееся именно в этом числе и именно в этом падеже.

При описании структур данных будем пользоваться обозначениями GType для грамматической категории («число») и GValue для конкретной альтернативы («единственное»), в соответствии с традицией наименования пользовательских типов данных при проектировании информационных систем.

6. Представление морфологических форм. Под морфологическими формами понимаются регулярные видоизменения слов определённых частей речи, несущие комплекс морфологических значений [11] (например, форма именительного падежа множественно числа существительного, форма 1-го лица единственного числа настоящего времени глагола, форма сравнительной степени прилагательного).

В рамках предлагаемой модели грамматическую информацию будем представлять совокупностью пар { GType : GValue }, описывающей грамматическую информацию, присущую некоторой форме слова, классу слов или цепочке преобразований, полностью или частично, и назовём **картой грамматических значений** (GValueMap).

С точки зрения абстрактных структур данных карта грамматических значений является ассоциативным массивом с ключом «грамматическая категория» и значением «граммема» [поскольку каждому из ключей пар, входящих в карту (конкретных грамматических категорий), соответствует одно и только одно значение – конкретная альтернатива].

Пример: карта грамматических значений, имеющая вид *{{«падеж»: «творительный»}, {«число»: «множественное»}}* полностью описывает грамматическую информацию, присущую слову «окнами», а карта *{{«число»: «множественное»}}* частично описывает любое из слов, получаемых путём склонения слова «брюки».

Одна и та же карта грамматических значений не может содержать две пары с одинаковым значением ключа (это следует напрямую из определения грамматической категории). В частности, например, число не может быть одновременно единственным и множественным.

7. Представление правил получения словоформ. В рамках модели, предложенной в работе [2], цепочка преобразований хранит в себе

не только правило получения словоформы из начальной формы слова, но и грамматическую информацию, которой наделяется словоформа, получаемая путём применения этой цепочки к некоторой начальной форме слова.

В рамках предлагаемой в настоящей работе модели цепочка преобразований, помимо команд модификации входной строки, хранит также карту грамматических значений, которыми обладает словоформа после применения цепочки.

Пример: цепочка *«(-а+и), {{«падеж» : «именительный»}, {«число» : «множественное»}}»* (операции: отделить постфикс *а* и добавить постфикс *и*), позволяющая, например, получить форму множественного числа именительного падежа слова «рука».

Легко видеть, что возможно наличие как различных цепочек с одинаковыми картами грамматических значений (в творительном падеже женского рода единственного числа слово «синий» имеет две формы – «синею» и «синей»), так и различных цепочек с одинаковыми правилами преобразования строки (слово «ручки» может быть как множественным числом именительного падежа слова «ручка», так и единственным числом родительного падежа того же слова). Второе обстоятельство приводит к появлению омонимии, которую следует разрешать на этапе синтаксического анализа текста.

8. Представление парадигм. По определению [11], все формы изменяемого слова составляют его ***парадигму***. Парадигма может быть либо избыточной (*мамой*/*мамою*), либо полной (12 форм существительного *карандаш*), либо неполной (нет формы первого лица единственного числа у глагола *победить*), либо дефектной (налить *щец*; у слова *щец* отсутствуют другие формы и слово используется только в родительном падеже множественного числа).

В рамках предлагаемой модели организации словаря будем рассматривать множество всех синтетических словоформ (образованных путём изменения начальной формы слова).

Множество всех возможных словоформ некоторой начальной формы слова в рамках предлагаемой модели описывается множеством результатов применения всех применимых к этой начальной форме слова цепочек и содержит все возможные пары *{слово; карта грамматических значений}*.

Следует заметить, что при решении задачи анализа словоформы не играет роли способ представления этого множества (то есть его структура с точки зрения применяемой модели: множество может быть представлено в виде дерева, таблицы, простого списка пар или иным образом), поскольку грамматическая информация заранее неизвестна, и перебор идёт по правилам преобразований строк имеющихся цепочек, а не по составу соответствующих им карт грамматических значений.

Тем не менее, структура этого множества играет важную роль в следующих случаях:

– при решении задачи генерации требуемой формы слова (извлечение нужной цепочки из древовидной структуры данных происходит быстрее, чем, например, из списка);

– при построении самой базы знаний о формообразовании языка (в случае структурированного множества всех возможных словоформ система автоматически способна предложить набор цепочек для добавляемого в базу нового слова, оставляя человеку возможность внести необходимые правки: такой подход при заполнении базы существенно ускорит процесс).

В случае, например, имени существительного, структура этого множества является довольно простой, и подчиняется правилу «всякое существительное находится в определённом числе и падеже», а для обработки слов-исключений (например, несклоняемых или не имеющих единственного числа) достаточно добавить в описание конкретных групп слов исключение из рассмотрения соответствующих альтернатив грамматических категорий числа и падежа.

С множеством всех возможных словоформ данного слова связано множество всех возможных карт грамматических значений. Согласно самой классификации парадигм [11], в большей части случаев (для слов с **полной парадигмой**) множество всех возможных карт грамматических значений для некоторого класса слов будет совпадать со множеством карт множества словоформ конкретного слова этого класса. Исключениями будут являться слова с **избыточной парадигмой** (например, существительные, у которых форма предложного падежа не совпадает с формой местного падежа: *в лесу*, но *о лесе*) – у таких слов словоформ больше, чем возможных карт грамматических значений, поскольку для некоторых грамматических значений допускается двоякое написание (*синей*/*синею*), а также слова с дефектной и неполной парадигмами: у таких слов словоформы существуют не для любой из возможных карт грамматических значений.

Структуру множества всех возможных карт грамматических значений для имён существительных точно описывает декартово произведение множеств граммем категорий числа и падежа, рассматриваемых как множества граммем (таблица 1).

Таблица 1 – Парадигма существительного «лес»

Падеж / Число	им.	род.	дат.	вин.	твор.	предл.
ед.	лес	леса	лесу	лес	лесом	лесе/ лесу
множ.	леса	лесов	лесам	леса	лесами	лесах

Однако простое перечисление всех изменяемых грамматических категорий оказывается неподходящим для описания, например, имён прилагательных: к изменяемым грамматическим категориям относят род, число и падеж, однако для множественного числа род определить нельзя. Проблему наглядно иллюстрирует таблица 2.

В гипотетической системе, которая в процессе внесения новых слов в базу предлагала бы пользователю заполнить всю таблицу словоформ (то есть формировала бы парадигму, используя прямое произведение множеств граммем), можно было бы применить следующие приёмы для разрешения этой проблемы:

– ввод фиктивной грамматической категории «род и число прилагательного», имеющей четыре значения: «мужской», «женский», «средний», «множественное число»;

– ручная или автоматизированная правка цепочек (удаление всех цепочек множественного числа одного и того же падежа кроме единственной и удаление информации о роде из неё);

– создание дополнительной структуры правил-ограничений.

Ни один из этих подходов не решает проблему в корне: введение фиктивных категорий потребует специальной постобработки и внесёт избыточность в структуру самих грамматических категорий, ручная правка цепочек потребует затрат времени оператора, автоматизированная правка цепочек (в том числе на основе дополнительных правил-ограничений) затруднительна без привязки модели к конкретному языку на уровне алгоритмов.

Таблица 2 – Парадигма прилагательного «ясный»

Степень сравнения, форма, падеж / Число, род, одушевлённость			Единственное число			мн.ч., одуш./ неод.
			м. род, одуш./ неод.	ж. род	ср. род	
Положительная степень сравнения	Полная форма	им.	ясный	ясная	ясное	ясные
		род.	ясного	ясной	ясного	ясных
		дат.	ясному	ясной	ясному	ясным
		вин.	ясного/ ясный	ясную	ясное	ясные/ ясных
		твор.	ясным	ясной/ ясною	ясным	ясными
		предл.	ясном	ясной	ясном	ясных
	Краткая форма		ясен	ясна	ясно	ясны
Сравнительная степень			яснее			

Следует отметить, что, хотя само по себе прилагательное и не является одушевлённым или неодушевлённым, поскольку обозначает признак предмета, а не сам предмет, в рамках предлагаемой модели конкретный вариант трактовки прилагательного также может включать информацию об одушевлённости (такая информация может помочь на этапе синтаксического анализа). Например, в предложении «*Сын умный отца уважал*» прилагательное *умный* в результате анализа получает два варианта трактовки (именительный или винительный падеж мужского рода, начальная форма «умный»), и без информации об одушевлённости могло бы быть ошибочно согласовано с существительным «отца», также находящемся в единственном числе и винительном падеже. Дополнительная информация о том, что словоформа «умный» согласуется только с винительным падежом при условии неодушевлённости, позволяет сразу отбросить этот ошибочный вариант трактовки. Можно также заметить, что информация об одушевлённости также не всегда позволяет устранить омонимию, связанную с определением главного слова словосочетания: так, в предложении «*Сын красивый цветок поливал*» устранение омонимии на этапе анализа конкретного слова невозможно (оба варианта трактовки имеют смысл), что показывает важность учёта всех вариантов трактовки анализируемых слов.

Ещё сложнее дело обстоит с описанием структуры множества всех возможных карт грамматических значений для глаголов (формами одного и того же слова являются инфинитив, личные формы, деепричастия и причастия). Попытка представить множество всех возможных карт грамматических значений в виде подмножества прямого произведения множеств граммем из разных грамматических категорий приводит к необходимости исключения такого количества недопустимых элементов (комбинаций граммем), что непрактичность подобного подхода к описанию структуры множества всех возможных карт грамматических значений становится очевидной.

В связи с этим предлагается представлять множество всех возможных карт грамматических значений в виде совокупности **карт формообразования**.

Картой формообразования (InflectionMap) будем называть структуру данных, описывающую некоторую совокупность карт грамматических значений через перечисление независимых грамматических категорий. Карта формообразования в описываемой модели содержит следующие данные:

– множество независимых грамматических категорий;

– множество ограничений вышеназванного множества грамматических категорий;

– карту постоянных грамматических значений (возможно, пустую).

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

Пример. Множество всех возможных карт грамматических значений полных прилагательных (без учёта одушевлённости) в этом случае можно было бы описать с помощью двух карт формообразования:

1) ({род, падеж}, Ø,{{число: ед.}});

2) ({падеж}, Ø, {{число: мн.}}).

В приведённом примере пустое множество Ø означает, что ограничений на сами грамматические категории в рамках перечисленных карт не накладывается.

Легко видеть, что, например, для описания формообразования имён существительных достаточно одной карты формообразования ({число, падеж}, Ø, {{число: ед.}}).

9. Иерархия типов слов в словаре. Все слова языка традиционно делят на классы в зависимости от их роли в языке, правил их формообразования, признаков и смыслового значения. Такие классы слов в рамках конкретного языка называют *частями речи*. Настоящая же работа не ориентируется на терминологию конкретных языков, поэтому будем называть ***супертипами*** (Supertypes) классы верхнего уровня в структуре базы знаний о формообразовании.

Супертипы будем характеризовать следующими признаками:

– совокупность постоянных грамматических признаков (например, род и одушевлённость для существительных);

– множество карт формообразования, задающих множества возможных форм для слов, принадлежащих супертипу (например, карты формообразования для личных форм, форм прошедшего времени и причастий у супертипа «глагол»);

– множество *типов слов* (рассмотрены ниже) данного супертипа.

Таким образом, супертипы – это объекты верхнего уровня в иерархии объектов словаря в рамках предлагаемой структуры. Назначением супертипов является описание всех возможных структур парадигм слов, принадлежащих к ним, а также описание всех возможных совокупностей постоянных грамматических признаков, которыми могут обладать слова, относящиеся к данному супертипу.

В пределах супертипа выделим ***типы слов*** (Kinds), объединяющие внутри себя все слова, парадигмы которых имеют одинаковую структуру (описываемую одним и тем же набором карт формообразования). Типы слов будем характеризовать:

– картой значений постоянных (в рамках типа) грамматических признаков (например, для типа существительных «существительные без единственного числа» таковой будет являться карта «*{ число: множественное }*»);

– совокупностью ограничений карт формообразования, доступных в рамках родительского супертипа (так, для типа «относительные прилагательные» карта формообразования, обеспечивающая получение сравнительной степени, будет недоступна);

– картой грамматических значений, соответствующей начальной форме любого из слов данного типа;

– совокупностью *семейств слов* (семейства рассмотрены ниже), принадлежащих данному типу.

Примеры для существительных: существительные женского рода неодушевлённые, женского рода одушевлённые, мужского рода одушевлённые, мужского рода неодушевлённые, среднего рода, несклоняемые существительные, существительные без единственного числа.

Таким образом, типы слов – это объекты второго уровня в иерархии объектов словаря в рамках предлагаемой модели. Назначением типов является группировка всех слов, обладающих одинаковой структурой формообразования (те или иные грамматические формы имеются или отсутствуют исключительно одновременно у всех представителей конкретного типа слов).

Наименьшим из наборов слов в рамках предлагаемой модели словаря является ***семейство*** (Family).

Семейство характеризуется:

– набором цепочек преобразований, применимых к каждому из слов семейства;

– набором слов (в начальной форме), относящихся к данному семейству.

Таким образом, семейство слов – это набор слов, подчиняющихся строго одинаковым правилам формообразования (иначе говоря, имеющих идентичную структуру парадигм и одинаковые наборы соответствующих правил получения словоформ из начальных форм слов).

Цепочка преобразований (Chain) 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 граммемы из числа переменных грамматических категорий в карте формообразования; значения прочих грамматических категорий берутся из карт родительских сущностей.

10. Обобщённая структура словаря

Рассмотрим структуру словаря, описанную в рамках предлагаемой модели, на примере представления в словаре глаголов (рисунки 1-3). Детали, обеспечивающие целостность физического представления данных (коды и идентификаторы сущностей) и детали реализации [примитивные и агрегатные типы данных (строки, коллекции, массивы и ассоциативные наборы), абстрактные тип данных, отношения наследования и ассоциации], были опущены, поскольку приведённая схема представляет собой логическую модель.

Словарь конкретного языка представляет собой совокупность автономно хранимого реестра грамматических категорий данного языка и множества супертипов слов.

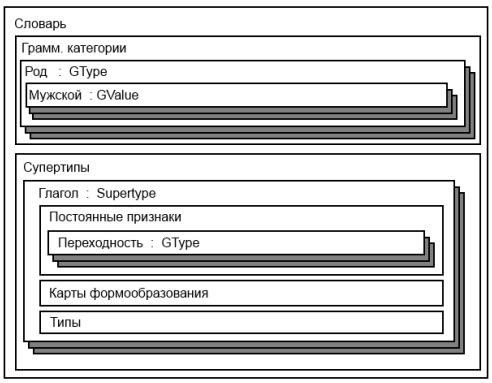

Рисунок 1 – Модель данных словаря

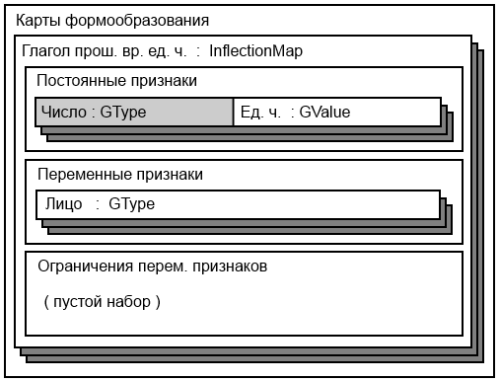

Рисунок 2 – Карты формообразования глагола

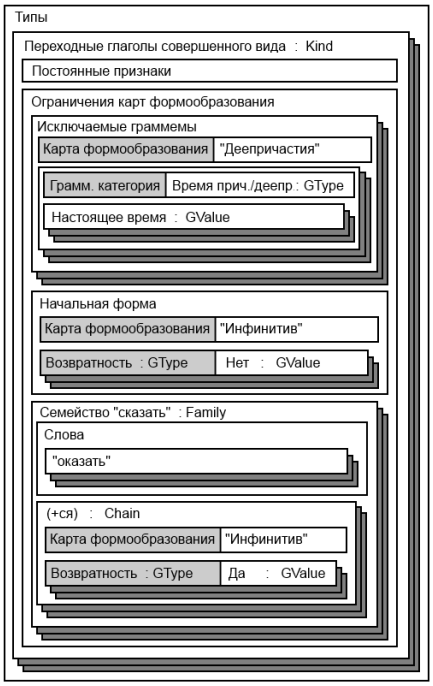

Рисунок 3 – Типы глаголов

Приведём ряд замечаний относительно логической модели:

– один экземпляр сущности «исключаемые граммемы» содержит перечисления всех исключаемых граммем для всех ограничиваемых грамматических категорий в рамках некоторой конкретной карты формообразования;

– сущность «начальная форма» содержит карту грамматических значений, соответствующую всем словам типа в их начальных формах, и содержит те же поля, что и любая цепочка, основывающаяся на той же карте формообразования за исключением, разумеется, полей, хранящих сами команды преобразования строк.

Заключение. Предлагаемая модель словаря обладает рядом достоинств по сравнению с приведёнными в начале статьи способами организации словаря:

– предлагаемая модель позволяет описывать формообразование так, что добавление новых групп слов не требует знания структуры парадигмы этих слов: с учётом определённого супертипа и типа для нового слова автоматически формируется набор карт грамматических значений: так, если пользователь указывает, что новое слово является существительным, система *автоматически информирует его о необходимости задать все 12 возможных форм*, что исключает возможность забыть указать, например, множественное число творительного падежа (особенно ценным данное преимущество предлагаемой модели становится при добавлении глаголов);

– предлагаемая модель успешно справляется с описанием формообразования глаголов и других классов слов, для которых структура парадигмы сильно отличается от простого декартова произведения множеств граммем, соответствующих независимым грамматическим категориям;

– предлагаемая модель представляет собой сложную систему типов, детально описывающую формообразование языка, и потому анализатор, построенный с использованием этой модели, может иметь широкое применение. Так, в системах, выполняющих синтаксический анализ предложений, могут использоваться не только результаты работы анализатора форм слов, но и сама система типов данных предлагаемой модели и сами по себе данные словаря.

Таблица 3 позволяет сравнить характеристики разных моделей представления словарей в системах генерации и определения форм слов естественных языков по критериям скорости анализа, удобства наполнения словаря человеком (в смысле возможности автоматизации процесса языконезависимыми алгоритмами), удобства представления структуры грамматической информации (на примере русского языка – справляется ли модель с описанием морфологии глаголов, учитывая формы причастия и деепричастия), избыточности модели грамматической информации (доля запрещённых комбинаций в коде описания грамматической информации).

Таблица 3 – Сравнение моделей словарей в системах определения форм слов

Критерий	Неструктурированные словари	Словари без иерархии типов	Полные словари всех словоформ	Предлагаемая модель
Скорость анализа	Не оптимизирована	Высокая	Максимальная	Высокая
Потребление памяти	Обычное	Обычное	Очень большое	Обычное
Удобство внесения данных	Очень низкое	Низкое	Очень низкое	Высокое
Структурированное представление грамм. информации о словах со сложным формообразованием	Возможно	Затруднительно	Зависит от вида базовых словарей, порождающих полный словарь	Возможно
Избыточность модели грамм. информации	Зависит от структуры словаря грамм. информации	Высокая	Зависит от вида базовых словарей	Низкая
Интеграция в прикладные системы, сопряжение с синтаксическим анализатором	Затруднена отсутствием категорий у слов словаря	Возможна	Зависит от вида базовых словарей	Модель ориентирована на интеграцию

Выводы. Таким образом, на качественном уровне можно утверждать, что предлагаемая модель словаря *предпочтительна* по сравнению с более простыми моделями в системах, не требующих максимальной скорости анализа, и может быть использована в качестве исходных данных для построения полных словарей словоформ в системах, требующих максимальной скорости.

Задача структурирования словаря системы генерации и определения форм слов была решена (разработанная модель описана в настоящей работе).

Поставленные цели были достигнуты:

– построена модель организации словаря, обладающая преимуществами по сравнению с существующими аналогами;

– в рамках построенной модели полностью описана структура формообразования всех частей речи русского языка, включая особые формы глаголов – причастия и деепричастия, а также слова-исключения с неполными, дефектными и избыточными парадигмами.

Результаты работы могут применяться для решения задач статистической обработки текстов, машинного перевода и других задач, подразумевающих ввод, выдачу или обработку информации на естественном языке с использованием ЭВМ.

Библиографический список

1. *Пруцков А.В.* Морфологический анализ и синтез текстов посредством преобразований форм слов // Вестник Рязанской государственной радиотехнической академии. 2004. № 15. С. 70-75.

2. *Пруцков А.В.* Генерация и определения форм слов естественных языков на основе их последовательных преобразований // Вестник Рязанского государственного радиотехнического университета. 2009. № 27. С. 51-58.

3. *Пруцков А.В., Розанов А.К.* Программное обеспечение методов обработки форм слов и числительных // Вестник Рязанского государственного радиотехнического университета. 2011. № 38. С. 78-82.

4. *Миронов В.В*. Объединение одночленных многообразий алгебр // Матем. заметки. 1984. Т. 35. № 6. С. 789-794.

5. *Миронов В.В.* Отсутствие конечного базиса тождеств свободных 2-ступенно разрешимых алгебр конечной степени свободы // Матем. заметки. 1988.Т.43.№ 3. С. 320-326.

6. *Белоногов Г.Г., Богатырев В.И.* Автоматизированные информационные системы / под ред. К.В.Тараканова. М.: Сов. радио, 1973. 328 с.

7. *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.

8. *Ножов И.М.* Прикладной морфологический

анализ без словаря // Тр. конф. по искусственному интеллекту КИИ-2000. – М.: Физматлит, 2000. – Т. 1. – С. 424-429.

9. *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.

10. *Гельбух А.Ф., Сидоров Г.О.*. К вопросу об автоматическом морфологическом анализе флективных языков // Труды межд. конференции Диалог-2005, М., 2005, стр. 92-96.

11. *Жеребило Т.В.* Словарь лингвистических терминов: изд. 5-е, испр-е и дополн. - Назрань: Изд-во "Пилигрим". 2010. – 486 с.

УДК 004.932

М.В. Колмыков, В.Н. Ручкин

ПРИМЕНЕНИЕ ФРАКТАЛЬНОГО АЛГОРИТМА СЖАТИЯ В ЦЕЛЯХ МИНИМИЗАЦИИ МОБИЛЬНОГО ТРАФИКА

Рассмотрен подход для оптимизации мобильного интернета. Предложен улучшенный алгоритм фрактального сжатия изображений, оптимизированный для нейропроцессора NM 640X. Показана целесообразность применения программно-аппаратного комплекса ImCoWeb для эффективного сжатия изображений и сокращения мобильного трафика.

Ключевые слова: *мобильный трафик, фрактальный алгоритм, нейропроцессор.*

Введение. В последнее время наиболее популярным типом доступа к Интернету становится доступ с мобильных устройств. Причиной этого послужило сочетание нескольких факторов, главными из которых стали: возможность доступа к сети из любого места, компактность и легкость самих устройств. Для мобильных ПК сложилась следующая ситуация: старые устройства, имеющиеся в огромном количестве у потребителей, невозможно дешево модернизировать под последние стандарты связи, а новые вынуждены использовать связь предыдущего поколения из-за отсутствующей инфраструктуры мобильных операторов. В связи с этим попрежнему актуальными являются сети, построенные на технологиях GPRS (EDGE). Возникает проблема «бутылочного горлышка»: возрастающее потребление интернет-контента ложится на мобильные сети, не предназначенные для таких объемов данных. К тому же мобильные операторы отдают приоритет речевому трафику, что еще более уменьшает полосу пропускания для интернета. Помимо этого, имеется проблема недоступности веб-ресурсов в определенных регионах в связи с накладываемыми законодательными ограничениями.

Постановка задачи. Задачей настоящей работы является создание такого способа обработки данных, при котором осуществляется значительное снижение потребляемого трафика. Для этих целей предполагается использовать посредника в передаче данных – программно-аппаратный комплекс, на котором и будет производиться основная работа минимизации. Необходимо определить меру количества потерь информации при сжатии. Предлагается использовать меру отношения сигнала к шуму (peak-to-peak signal-to-noise ratio – PSNR). Следует также учесть технические характеристики системы, которая будет решать прикладную задачу, количество ресурсов, выделяемых для ее решения.

Ресурсоемкость процессов кодирования и декодирования характеризуется симметричностью – отношением характеристик алгоритма при кодировании к характеристикам при декодировании. Важнейшими показателями являются симметричность по времени и симметричность по памяти. Минимизация характеристик кодирования необходима для решения задачи сжатия трафика, а минимизация характеристик декодирования – для задач воспроизведения изображений на мобильном устройстве. *Цель работы* – показать применимость фрактального алгоритма к решению имеющейся проблемы и продемонстрировать эффективность сочетания аппаратной платформы и оптимизированного под нее алгоритма компрессии данных.

Описание программно-аппаратного комплекса ImCoWeb. Основным методом уменьшения данных является использование прокси-серверов. Помимо своей главной задачи – сжатия трафика – они могут также выполнять другие действия: помогать обходить ограничения доступа к заблокированным ресурсам, а также

проверять мобильный трафик на вирусы и другие угрозы. Работу прокси-серверов можно представить следующими этапами:

1. Клиент запрашивает незашифрованную веб-страницу.
2. Сервер закачивает ее себе, обрабатывая имеющееся в ней содержимое.
3. Сервер отсылает клиенту полученную сжатую веб-страницу.

Зашифрованные страницы идут напрямую потребителю, следовательно, не подвергаются обработке и сжатию.

Основные реализации сжимающих прокси-серверов обычно можно встретить у производителей мобильных браузеров. К примеру, Google в своем браузере Google Chrome предоставляет функцию «Сократить использование данных», при активации которой весь нешифрованный трафик идет через серверы Google. У других браузеров (Opera, Internet Explorer) имеются схожие функциональные возможности (при включенном режиме экономии трафик пропускается соответственно через серверы Opera и Microsoft). Для связи прокси-серверов с браузером, как правило, используется более быстрый протокол SPDY. У данных реализаций имеется следующий недостаток: прокси-сервер доступен только для «своего» браузера, что обеспечивает различность отображения веб-страниц в разных браузерах при включенном режиме экономии трафика. Также в представленных решениях недоступна возможность настройки степени сжатия. В связи с этим целесообразно отделить реализацию сжимающего прокси-сервера от браузера, обеспечив тем самым возможность одинакового отображения в привычных для пользователя браузерах.

Программно-аппаратный комплекс ImCoWeb представляет собой доступный для любого браузера сжимающий прокси-сервер. При его использовании сжимаются текст и изображения, не подвергаются сжатию JavaScript, AJAX и Flash, видеоролики, SVG-графика и GIF-анимация. Зашифрованные страницы доступны пользователю напрямую.

Современные сайты содержат большое количество графики, а для некоторых (к примеру, популярного Instagram.com) она значительно преобладает над текстом. Следовательно, одной из задач, которую должен реализовывать сжимающий прокси-сервер для уменьшения потребляемого трафика, является обработка и сжатие проходящих через него изображений. Для сжатия на прокси-сервере изображений требуется выбрать графический формат, обеспечивающий, по возможности, максимальную степень сжатия. В качестве такого для комплекса ImCoWeb был выбран формат, основанный на фрактальном алгоритме. Данный алгоритм обладает неоспоримым преимуществом по сравнению с другими, а именно минимальным размером результата кодирования (в 100 раз лучше по сравнению с ДКП алгоритмами). Также возможно гибкое управление коэффициентом компрессии, что, при доступных настройках в ImCoWeb, позволяет пользователю самому выбирать качество загружаемых изображений. Также к сжатым изображениям можно применить фрактальное масштабирование и фрактальную интерполяцию [1]. В решении Google для изображений, встречающихся в трафике, применяется свой формат WebP, основанный на алгоритме сжатия неподвижных изображений (ключевых кадров) из видеокодека VP8. Преимуществом фрактального алгоритма по сравнению с WebP является меньший размер сжатого файла. К тому же формат WebP не поддерживается большинством мобильных браузеров, что ставит его в равное с фрактальным положение по распространенности.

Классический фрактальный алгоритм обладает следующим недостатком: при достаточно быстром декодировании долгое время занимает процесс кодирования. В отличие от алгоритмов ДКП, в которых происходит однозначное преобразование элементов изображения, во фрактальном алгоритме происходит поиск максимально соответствующих частей изображения за счет их полного перебора и попиксельного сравнения, что и приводит к замедлению работы. Следовательно, основной задачей при исследовании данного алгоритма является разработка методов сокращения времени кодирования [2]. Наиболее эффективными оказались два направления исследований: метод выделения особенностей (feature extraction) и метод классификации доменов (classification of domains). Первый метод заменяет попиксельное сравнение частей изображения сравнением векторов его характеристик (заранее вычисленных и нормированных). Второй метод, использующий карты Кохонена, логическое продолжение первого метода, значительно уменьшает число операций перебора доменных блоков. Однако достигнутые результаты не привели к резкому уменьшению времени кодирования.

В предлагаемом программно-аппаратном комплексе ImCoWeb используется модернизированный автором фрактальный алгоритм [3]. Для ускорения классического алгоритма были применены нейронные сети, в результате было достигнуто значительное сокращение времени ко-

дирования. Кратко схему кодирования предложенным алгоритмом можно представить следующим образом.

1. Задаем пороговое значение e_c для ошибки и минимальный размер $r_{\min}$ для ранга. (Значение ошибки определяется как среднее абсолютной разности уровней серого пикселей ранга и соответствующего домена. В домене значения 4 соседних пикселей усредняются и сравниваются со значением уровня серого пикселя соответствующего ранга. Чем ниже ошибка e_c, тем выше степень подобия.)

2. Разделяем изображение на неперекрывающиеся блоки – ранги с начальным размером 32x32 и перекрывающиеся блоки – домены, в 2 раза большие по размеру, чем ранги.

3. Для каждого ранга i находим домен j, для которого ошибка между i и j меньше или равна порогового значения ошибки e_c. Затем определяем функцию трансформации τ_i для ранга i. Обновляем веса ϖ_{ij} для всех пикселей ранга i и всех пикселей домена j нейросети, используя дельта-правило обучения для настройки контрастности s_i и яркости o_i в функции трансформации τ_i.

4. Если для ранга i не удается подобрать подходящий домен j (т.е. ошибка между i и j больше порогового значения e_c), то разделяем ранг i на 4 подблока, размер каждого из которых не меньше $r_{\min}$. Переходим к шагу 3 для нахождения функции трансформации для каждого уменьшенного ранга.

5. Если размер ранга i равен $r_{\min}$ и подходящего домена не найдено, тогда ранг i больше не разбивается, и выбирается домен j с минимальной ошибкой. В этом случае также определяется функция трансформации τ_i.

Особенностью предложенного алгоритма, позволяющей сократить время кодирования, является возможность выполнять вычисления параллельно, используя нейросеть, в отличие от традиционного последовательного алгоритма. Также предложенный подход генерирует лучшего качества изображение благодаря своей гибкости и надежности, более точному сопоставлению подблоков изображения.

Сжатие изображений на прокси-сервере должно реализовываться в реальном времени. Возникает потребность освобождения ресурсов центрального процессора от затратной задачи сжатия. Для решения данной задачи предлагается использование дополнительных ускорителей.

В качестве основы ускорителей были рассмотрены специализированные цифровые сигнальные процессоры, предназначенные для цифровой обработки сигналов (обычно в реальном масштабе времени). Было произведено сравнение дополнительных устройств на их базе, на которых должна будет осуществляться обработка и сжатие изображений. Одним из наиболее оптимальных устройств для работы с модернизированным фрактальным алгоритмом являются те, которые используют в своей работе нейропроцессор по причине того, что возможности DSP не полностью соответствуют задачам, которые возникают при моделировании нейронных систем. В качестве устройства для комплекса ImCoWeb был выбран отечественный нейропроцессор семейства NeuroMatrix [4], показавший отличные результаты в системах дистанционного зондирования земли, системах слежения за объектами и др. благодаря своей универсальности.

Задача обработки изображений в реальном масштабе времени является весьма ресурсоемкой. Классические алгоритмы сжатия, как правило, реализованы на языках программирования подобно Си. Они рассчитаны на последовательное выполнение команд, характерных для процессоров x86. Для оптимизации программы сжатия для нейропроцессора недостаточно простой адаптации кода под него, ведь в этом случае нельзя получить выигрыша в производительности. Необходимо использование распараллеливания вычислений. Наличие у процессора матричного вычислительного узла позволяет вести вычисления результата сразу нескольких шагов преобразования. Конечно, каждый шаг может быть реализован отдельно, однако эффективность параллельной обработки нескольких слоев будет несравнимо выше. Процесс накопления значений кадров можно выполнять параллельно с другими командами, что существенно сокращает время обработки.

Также одной из возможностей процессора семейства NM640X является программное изменение разрядности обрабатываемых данных. Это означает, что, задав соответствующее разбиение его рабочей матрицы, участвующей в вычислениях, можно в течение нескольких шагов выполнять преобразования над восьмиразрядными данными, а когда теоретически рассчитанная разрядность результатов потребует выхода за 8 бит, преобразовать данные к 16-разрядному виду и продолжить вычисления и т. д. Максимально возможная разрядность накопителя, реализованная в процессоре NM6403, составляет 64 бита.

Структурно схема комплекса ImCoWeb представлена на рисунке.

Структурная схема комплекса ImCoWeb

Описание реализации алгоритма фрактального сжатия на вычислительном комплексе. Для реализации фрактального алгоритма сначала были разработаны макросы простых операций на языке нейроассемблер. Это макросы сложения, умножения, а также макросы работы с матрицами. Затем макросы были объединены между собой. Таким образом, были получены наиболее распространенные операции для данного алгоритма сжатия изображения, а также для многих других программ обработки изображений. Для разработки макросов использовался интерактивный помощник Evecom [5]. На последнем этапе было выполнено непосредственно само написание программы на нейроассемблере с использованием разработанных макросов. Для тестирования и отладки полученной программы использовался инструментальный модуль МС 51.03 на базе процессора 1879ВМ5Я (NM6406), предназначенный для работы в составе ПЭВМ с системной шиной PCI для отработки функционального программного обеспечения вычислительных систем на базе процессора NM6406.

Модуль содержит один процессор 1879ВМ5Я с двумя банками синхронной динамической памяти по 64 Мбайт (по одному банку на каждой шине процессора). Один банк памяти доступен для записи и чтения как со стороны процессора, так и со стороны шины PCI.

Для предоставления возможности просмотра изображений, закодированных фрактальным алгоритмом, была разработана библиотека на языке Си, содержащая реализацию фрактального декодера. Данная библиотека может быть использована со всеми популярными мобильными веб-браузерами, а также легко портирована на Java при необходимости использования на устройствах, имеющих поддержку только Java Platform Micro Edition. Для современных мобильных устройств также доступна реализация декодера на языке JavaScript, позволяющая не устанавливать дополнительный плагин.

Экспериментальные исследования. Было проведено сравнение сжатия трафика комплексом ImCoWeb с другими реализациями прокси-серверов. Результаты представлены в таблице.

Сравнение ImCoWeb с другими прокси-серверами

Прокси-серверы	Сайты (экономия трафика в %)			
	lib.ru (текст)	freeimages.com (изображения)	flickr.com (изображения)	deviantart.com (изображения)
Opera	42	53	55	72
Chrome	33	49	48	65
UC Browser	46	66	62	78
ImCoWeb	15	54	58	77

По результатам работы комплекса можно сделать следующий вывод: чем больше изображений на веб-странице, тем лучше работает ImCoWeb по сокращению трафика в сравнении с аналогами.

Выводы. Таким образом, были сделаны следующие выводы о целесообразности применения данного программно-аппаратного комплекса для обработки изображений, его возможностях, проблемах и способах их решения.

- Реализованный подход позволяет существенно сократить время кодирования фрактального алгоритма.
- Использование специализированного ускорителя позволяет снизить загрузку центрального процессора, освободив его для других задач.
- Использование в качестве ускорителя нейропроцессора семейства NeuroMatrix позволяет создать законченный и оптимизированный аппаратно-программный модуль, доступный для установки на любом сервере.
- Использование сжимающего прокси-сервера позволяет существенно минимизировать мобильный трафик, позволяя быстрее загружать веб-страницы на имеющихся каналах связи.

Заключение. В результате разработанная реализация сжимающего прокси-сервера позволила ускорить обработку изображений на сервере и существенно сократила потребляемый трафик. Таким образом, в статье показана целесообразность применения фрактального алгоритма для использования в мобильном сегменте Интернет.

Библиографический список

1. *Ватолин Д., Ратушняк А., Смирнов М., Юкин В.* Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М.: ДИАЛОГ-МИФИ, 2002. 384 с.

2. *Колмыков М.В.* Методы ускорения фрактального сжатия изображений // Информатика и прикладная математика. Рязань: Рязанский государственный университет им. С. А. Есенина, 2008. С. 69-72.

3. *Колмыков М.В., Ручкин В.Н.* Применение нейронной сети для фрактального сжатия изображений // Цифровая обработка сигналов. 2010. № 1. С. 51-52.

4. *Колмыков М.В.* Анализ DSP-процессоров в задаче обработки изображений // Аспирантский вестник РГУ им. С.А. Есенина. 2009. №14. С. 14-16.

5. *Романчук В.А., Ручкин В.Н.* Разработка программных средств анализа нейропроцессорных систем // Вестник Рязанского государственного радиотехнического университета. 2010. №32. С.61-67.

УДК 004.492.2

Л.С. Крупнов

РАЗРАБОТКА АЛГОРИТМА ФОРМАЛИЗАЦИИ ПАРАМЕТРОВ КОМПЬЮТЕРНОЙ СИСТЕМЫ ДЛЯ ОЦЕНКИ ОПАСНОСТИ СЕТЕВЫХ АТАК

Рассматривается задача нахождения зависимости между параметрами компьютерной системы и её состоянием после воздействия сетевых атак. Для решения применяется метод группового учета аргументов (МГУА), так как в этом случае обеспечивается наибольшая точность полученной математической модели. Ряд проведенных экспериментов подтверждает возможность прогнозирования реакции компьютерной системы в ответ на сетевые воздействия в автоматическом режиме. Анализ данной зависимости состояния компьютерной системы от наблюдаемых параметров позволяет обнаружить уязвимые места в системе безопасности и дать рекомендации по устранению найденных недостатков.

Ключевые слова: *метод группового учета аргументов, сетевые атаки, защита, администрирование.*

Введение. При проектировании систем защиты компьютерных сетей ключевую роль играет оценка степени опасности той или иной сетевой атаки [1, 5]. На данный момент не существует единого подхода для решения этой задачи, поскольку компьютерные системы (КС) сильно отличаются друг от друга. *Цель работы* – разработать алгоритм оценки опасности различных сетевых атак на основе объективных наблюдений за показателями качества КС, проверить работу алгоритма на примере простой КС.

Теоретическая часть. Для оценки опасности той или иной сетевой атаки недостаточно получить экспертную оценку предполагаемого ущерба для конкретной КС, поскольку невозможно предугадать состав и средства конкретной сетевой атаки, ввиду бесконечного множества возможных реализаций атак. Поэтому необходимо получить зависимость изменения состояния КС от изменения показателей качества КС, вызванных воздействием сетевых атак.

Формально данная подзадача звучит следующим образом – на основе ряда наблюдений за показателями качества КС, подвергаемой воздействию различных сетевых атак, требуется построить зависимость $y = f(x_1, x_2, ..., x_n)$ состояния системы от показателей качества системы $x_1 ... x_n$ для дальнейшего принятия решения об оптимальной атакующей или защитной стратегии изменения показателей качества КС. Для построения математической модели КС используются экспериментальные данные, полученные в результате наблюдений за поведением устройств в условиях воздействия различных сетевых атак. На начальном этапе в качестве входных данных принимаются все значимые параметры КС, чтобы в дальнейшем исключить из математической модели избыточные входные данные. На первом этапе исключаются сильно коррелированные переменные (путем исключения одной из переменных либо введением новой обобщенной переменной).

Для начала требуется разделить все данные о КС на подмножества ограничений, условий и

показателей качества [2]. *Показатели качества* условно разбиты на группы, отвечающие за работоспособность КС, обеспечение конфиденциальности информации в КС, защитные воздействия на КС (таблица 1). Чтобы избежать субъективности в оценке состояния КС, показатели качества выбираются максимально простыми и хорошо наблюдаемыми экспериментально. Более сложные и комплексные показатели качества выводятся в процессе реализации алгоритма.

Таблица 1 — Показатели качества КС

	Показатель качества системы	Единицы измерения, разъяснение
Работоспособность компьютерной системы	Нагрузка на устройство (отдельно для каждого устройства)	Процент занятости полосы пропускания, памяти, процессора и т.п.
	Вероятность отказа устройства (отдельно для каждого устройства)	Вероятность выхода из строя целевого устройства за определенный промежуток времени
	Количество потерянных транзакций в единицу времени в отношении к общему числу транзакций	Транзакцией может считаться минимальная операция, характерная для конкретного сервиса [обращение к базе данных (БД), отправленный пакет данных и т.п.]
	Целостность важных участков данных (отдельно для каждого участка либо устройства)	Процент целостности критичного, с точки зрения работоспособности системы, участка данных
	Среднее время транзакции	Среднее время обработки транзакции в миллисекундах
	Максимальное время проведение транзакции	Максимальное время обработки транзакции в миллисекундах
Конфиденциальность информации в системе	Количество несанкционированных транзакций (отдельно для каждого вида транзакций)	Количество транзакций, проведенных за время атаки, которые запрещены администратором системы (обращение к БД от незарегистрированного пользователя, некорректный пакет и т.п.)
	Конфиденциальность секретных участков данных (отдельно для каждого участка данных и группы лиц)	Количество информации, которое стало доступно для чтения лицам, не имеющим до этого подобного права, либо неидентифицированным лицам
Защитное воздействие	Время проведения атаки	Время непосредственного атакующего воздействия на компьютерную систему
	Стоимость воздействия	Условная стоимость воздействия в пересчете на ресурсы системы (как, например, дополнительная память, процессорное время, полоса пропускания канала и т.п.)
	Время реакции воздействия	Время, прошедшее с момента фактического появления опасности до момента применения защитных мер

Таким образом, для экспериментальной оценки указанных показателей качества требуется определить в системе следующие сущности.

1. Перечень объектов системы.
2. Перечень критических участков данных системы (с точки зрения целостности и конфиденциальности информации).
3. Перечень видов транзакций системы.
4. Перечень групп лиц в системе (пользователи, сотрудники, неидентифицированные лица и т.п.).
5. Перечень разрешений (т.е. перечень разрешенных транзакций отдельно для каждой из групп лиц).

В данном случае показатели качества системы приведены к нормальному виду, т. е.:

1) КС тем лучше, чем меньше каждый из ее показателей качества, т.е. задача оптимизации защиты КС сводится к минимизации целевой функции

$$F(x_1, x_2, ..., x_n) \to \min;$$

2) целевая функция монотонно убывает по каждому из показателей качества, т. е.:

$$F(x_1, x_2, ..., x_i, ..., x_n) < F'(x_1, x_2, ...x_i', ..., x_n),$$

если $x_i < x_i'$. При этом мы будем называть систему с показателями качества K' безусловно худшей, по сравнению с K. Систему с показателями качества К мы будем называть нехудшей, если выполняется векторное неравенство:

$$F(X) < F(X'), m.e. \forall x \in X, \forall x' \in X', x \le x'.$$

При данной постановке задачи для поиска оптимальной конфигурации системы требуется определить вид целевой функции. Для упрощения поиска целевой функции мы можем проводить анализ только для нехудших систем, так как они объективно обладают наилучшими па-

раметрами по сравнению с худшими при любом виде целевой функции [2].

Наиболее сложным вопросом является определение выходного параметра y, который, с одной стороны, отражает состояние КС в целом при различных воздействиях на нее, а с другой – хорошо наблюдаем при проведении тестовых испытаний системы. Поскольку в каждой КС имеются свои приоритеты, постольку и важность каждого отдельного устройства будет различаться от системы к системе. Это говорит о неизбежности экспертной оценки состояния КС при воздействии различных сетевых атак [3].

Подмножество условий. При синтезе оптимальной конфигурации системы защиты $\theta_{опт}$ необходимо принимать во внимание, что конфигурация КС ограничена возможностями устройств (или функциями перехода F), из которых она состоит. Иными словами, переход КС из одного состояния θ в другое возможен только при воздействии на КС определенными функциями перехода F: $\exists F,\ \theta_{опт} = F\theta$.

Выбор опорной функции. Выбор опорной функции во многом определяет точность полученной математической модели. Наибольшая точность достигается в том случае, если вид полинома опорной функции напоминает предполагаемую реальную зависимость. В нашем случае состояние КС $y = f(x_1, x_2, ..., x_m)$ определяется экспертным путем, исходя из степени выполнения требований и задач КС. Для прогнозирования вида итоговой зависимости предположим, что работу КС удалось разбить на независимые рабочие процессы r_i. Таким образом, работоспособность КС в первом приближении складывается из уровней работоспособности рабочих процессов по аддитивному критерию:

$$y(r_1, r_2 ... r_n) = \sum_{i=1}^{n} a_i r_i .$$

Для вычисления итоговой зависимости хорошо подходит метод группового учета аргументов (МГУА). Согласно теореме Вейерштрасса, данная зависимость может быть описана с помощью полинома Колмогорова - Габора со сколь угодной точностью:

$$y(r_1, r_2 ... r_n) = a_0 \sum_{i=1}^{n} a_i r_i + \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} r_i r_j + ...$$

Здесь a_i – весовые коэффициенты полинома. В качестве моделей-претендетов будут выступать полиномы второй степени с зависимостями вида $ar_i r_j$ и ar_i^2. При малом количестве рабочих процессов $n < 4$ есть возможность рассматривать полные полиномы третьей степени. Комплексные параметры в данном случае необходимы, так как далеко не все КС разбиваются на полностью независимые рабочие процессы. Полный полином второй степени будет иметь вид:

$$y = a_0 + a_1 r_1 + a_2 r_2 + a_3 r_2^2 + a_4 r_1 r_2 .$$

Для вычисления параметров системы $r_1 ... r_n$ используется мультипликативный алгоритм МГУА, так как предполагаемая зависимость работоспособности отдельного рабочего процесса не должна допускать компенсации одних параметров системы другими либо минимизировать ее. Таким образом, мультипликативный полином будет иметь вид:

$$r = a_0 x_1^{k_1} x_2^{k_2} ... x_m^{k_m} = a_0 \prod_{j=1}^{m} x_j^{k_j} .$$

Здесь k_i – весовые коэффициенты полинома, а $x_1 ... x_n$ - показатели качества КС. В общем случае каждый рабочий процесс так или иначе зависит от всех показателей качества КС, однако для упрощения расчетов рекомендуется отбрасывать заведомо незначимые для конкретного рабочего процесса показатели качества. Прологарифмировав это выражение, получим:

$$\ln r = \ln a_0 + k_1 \ln x_1 + k_2 \ln x_2 + ... + k_m \ln x_m .$$

По выборке данных измерений величин $y, x_1, x_2, ..., x_m$ составим выборку прологарифмированных значений указанных факторов. Далее, применяя вышеуказанный алгоритм МГУА с линейными частными описаниями при единичной степени факторов, находим логарифмическую модель оптимальной сложности. Потенциируя ее, получаем искомую оптимальную (в смысле минимума некоторого внешнего критерия) мультипликативную модель заданного процесса. Для того чтобы получить обобщенную мультипликативно-аддитивную модель, по обычному алгоритму МГУА выбираем некоторое количество F лучших моделей, используемых на втором этапе алгоритма в качестве регрессоров. Таким образом, обобщенный вид исследуемой зависимости имеет вид:

$$y(x_1, x_2, ..., x_m) = \sum_{i=1}^{n} a_i \left(\prod_{j=1}^{m} x_j^{k_j} \right).$$

Для получения оптимальной модели КС требуется формализовать *внешние критерии* выбора [4]. Претенденты на описание рабочих процессов r_i должны быть выбраны по следующим критериям.

Критерий точности краткосрочного прогноза $\Delta^2(C)$. Вся выборка исходных данных (N) разделена на обучающую (A), контрольную (B) и тестовую (C). На тестовой выборке вычисляется среднеквадратичное отклонение вычисленного с помощью модели значения y_t^M от истинного y_t :

$$\Delta^2(C) = \frac{\sum_{t \in C} \left(y_t^M - y_t\right)^2}{\sum_{t \in C} y_t^2} \to min .$$

Данный критерий необходим, так как итоговая модель предназначена в основном для краткосрочных прогнозов. Более того, он хорошо сочетается с критерием минимума смещения.

Критерий минимума смещения $n_{см}^2$. Его интерпретация такова: модель, оценка которой получена по данным определенного интервала наблюдения или в определенной точке наблюдения (y_t^A), должна как можно ближе совпадать с моделью, полученной по данным другого интервала наблюдения или в другой точке наблюдения (y_t^B). Критерий минимума смещения равен среднеквадратичному значению отклонений выходов моделей A и B для всей выборки $(t \in N)$:

$$n_{см}^2 = \frac{\sum_{t \in N} \left(y_t^A - y_t^B\right)^2}{\sum_{t \in N} y_t^2} \to min .$$

На основе данного критерия составляется комплексный критерий p. Поскольку оба этих критерия нормированы, мы можем выразить их следующим образом:

$$p = \sqrt{n_{см}^2 + \Delta^2(C)} \to min .$$

Данный комплексный критерий применяется для выбора математической модели оптимальной сложности для рабочих процессов r_i. Глубина минимума комбинированного критерия является мерой эффективности. Далее выбирается несколько самых эффективных моделей с уже известными коэффициентами lna_0, k_i, на основе которых строится обобщенная мультипликативно-аддитивная модель.

Экспериментальная часть: В качестве испытуемой КС будет выступать сеть передачи данных, смоделированная на тестовом стенде, с использованием средств виртуализации. Сеть состоит из 10 различных устройств, среди которых выделим: сервер базы данных, корневой коммутатор, администратор системы, пользователи базы данных, вспомогательные устройства коммутации.

Структура сети достаточно простая по нескольким причинам. Во-первых, это облегчает получение экспертных оценок реакции сети на различные воздействия, а во-вторых - позволяет применять как классический регрессионный анализ напрямую, так и предложенный мультипликативно-аддитивный.

В качестве показателей качества были выбраны следующие.

1. D1 … D10 – Процент потерь сетевых транзакций для каждого из объектов. В качестве транзакции выбран сетевой пакет TCP или UDP.

2. P1 … P3 – Процент потерь транзакций до базы данных от каждого из пользователей базы данных (под транзакцией подразумеваются: выполнение команд на чтение, запись, администрирование базы данных и т.п.).

3. S_1 – Процент целостности информации в базе данных.

4. C_1 – Количество информации БД (по отношению к общему количеству информации БД), ставшее доступным для чтения несанкционированной группе лиц. В данную группу входят все лица, не причастные к пользователям или администраторам БД.

5. X_1 … X_{10} – Количество несанкционированных транзакций управления для каждого из объектов со стороны неидентифицированных пользователей. В данный вид транзакций включены любые команды управления устройством, что означает захват управления со стороны злоумышленника.

Данный набор показателей качества определяется перечнем тестовых сетевых атак, которые, воздействуя на систему, не вызывают изменений в других показателях качества либо эти изменения незначительны.

На первом этапе был получен полином второй степени с помощью регрессионного анализа, коэффициенты которого предоставлены ниже (таблица 2). Некоторые параметры не приведены в таблице ввиду сильной корреляции с остальными. В качестве выходного параметра была принята условная величина работоспособности КС в диапазоне значений от 0 до 100.

Рассмотрим расхождения для полученной модели предсказанного значения от наблюдаемого (рисунки 1, 2). В данном случае максимальное отклонение достигает 17 %. Интересно также будет отметить, что в полученном полиноме отсутствуют квадратичные степени переменных, отражающих доступ к соответствующим устройствам, поскольку они имеют высо-

кую корреляцию со способностью устройств использовать сервисы КС.

Таблица 2 — Коэффициенты полинома второй степени для регрессионного классического алгоритма МГУА

Переменная	Коэффициент
Свободный член, a_0	-88,3752
P_1	68,7825
P_1^2	-16,8148
D_1	13,2742
D_1^2	0,0000
D_2	13,2742
D_2^2	0,0000
D_3	-1,7258
D_3^2	0,0000
D_4	-1,7258
D_4^2	0,0000
S_1	-59,7190
S_1^2	150,0313
X_5	-0,3709
X_5^2	0,0016
X_1	-0,0094
X_1^2	-0,0001
X_6	-2,6751
X_6^2	0,0223
C_1	0,0379
C_1^2	-0,0088

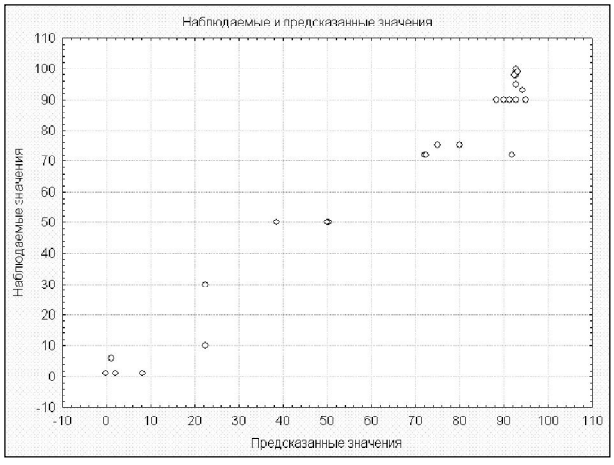

Рисунок 1 — Графики распределения предсказанных и наблюдаемых значений для классического алгоритма МГУА

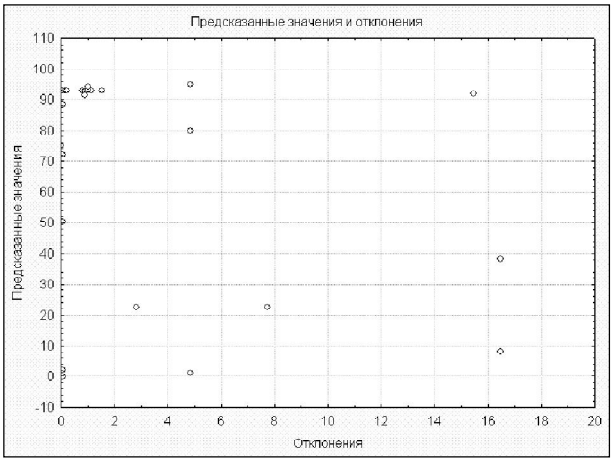

Рисунок 2 — Графики распределения отклонений предсказанного значения от наблюдаемого для классического алгоритма МГУА

Аналогичный анализ проводится для тех же самых исходных данных, только в этом случае с применением мультипликативно-аддитивного алгоритма МГУА (рисунки 3, 4).

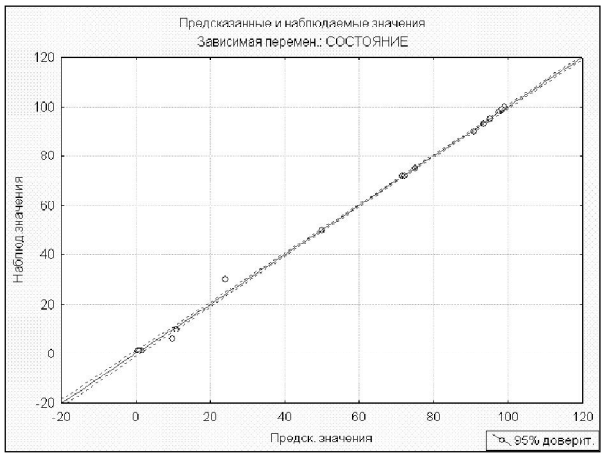

Рисунок 3 — Графики распределения предсказанных и наблюдаемых значений для мультипликативно-аддитивного алгоритма МГУА

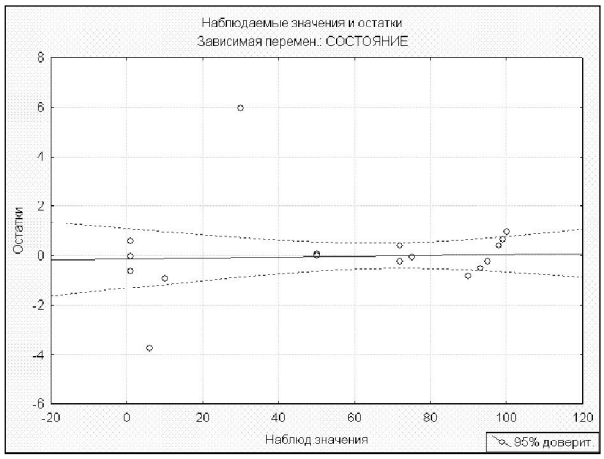

Рисунок 4 — Графики распределения отклонений предсказанного значения от наблюдаемого для мультипликативно-аддитивного алгоритма МГУА

Применение мультипликативно-аддитивного алгоритма снизило максимальное значение ошибки предсказания до 6 %. Также уменьшилась дисперсия ошибки, что позволило разместить 91 % всех наблюдаемых значений в 95 % доверительном диапазоне.

Полученная зависимость позволяет нам проводить дополнительный анализ с целью выявления наиболее опасных сетевых воздействий. Графической иллюстрацией наиболее чувствительных к атакам устройств является карта Парето (рисунок 5). На данном рисунке видно, что для данной КС наиболее критической задачей является защита базы данных (целостность и работоспособность). Причем задача сохранения данных является наиболее приоритетной. Таким образом, мы получаем возможность оценивать опасность сетевых атак, не участвующих в процессе получения полинома.

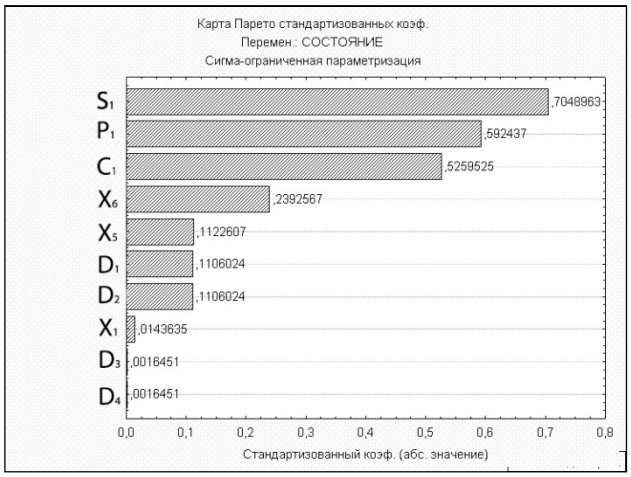

Рисунок 5 — Карта Парето входных параметров, отражающая относительный вес отдельных входных параметров

Итоговая зависимость выглядит следующим образом:

$$y(r_1, r_2, r_3, r_4) = -4{,}1459 + 0{,}623774 r_1 + \\ + 0{,}134698 r_2 + 0{,}861934 r_3 - 0{,}001343 r_4 - \\ - 0{,}485193 r_1^2 - 0{,}094928 r_2^2 - 0{,}922664 r_3^2 - \\ - 0{,}890662 r_4^2.$$

Непосредственно рабочие процессы имеют вид:

$$r_1 = S_1^{0{,}92231} P_1^{1{,}23341} C_1^{1{,}34111},$$

$$r_2 = S_1^{0{,}79402} D_1^{-0{,}09412} D_2^{-0{,}10031},$$

$$r_3 = S_1^{0{,}31141} X_1^{-0{,}84214},$$

$$r_4 = X_5^{0{,}39402} X_6^{1{,}23112}.$$

Отсутствие в полиноме множителей $r_i r_j$ говорит о хорошем разбиении входных переменных на независимые рабочие процессы. Данное поведение характерно для простых компьютерных систем.

Заключение. Разработан алгоритм оценки опасности различных сетевых атак на основе объективных наблюдений за показателями качества КС. Разработанный алгоритм успешно прошел проверку на тестовой КС. В ходе эксперимента все измерения показателей качества КС производились в автоматическом режиме с использованием ЭВМ. Полученные в ходе эксперимента распределения говорят о приемлемой точности спрогнозированных значений, что позволяет использовать полученные зависимости для оценки опасности произвольных сетевых атак.

Разработанный алгоритм может быть адаптирован для широкого спектра КС. Анализ аналитической зависимости состояния КС от наблюдаемых параметров позволяет найти узкие места в системе безопасности и дать рекомендации по устранению найденных недостатков.

Библиографический список

1. *Девянин П. Н.* и др. Теоретические основы компьютерной безопасности. М.: Радио и Связь, 2000 г. - 192 с.

2. *Гуткин Л.С.* Оптимизация радиоэлектронных устройств по совокупности показателей качества. М.: Советское радио, 1975 г. - 368 с.

3. *Домарев В.В.* Безопасность информационных технологий. Методология создания систем защиты. М.: ООО "ТИД "ДС", 2002 г. - 688 с.

4. *Ивахненко А.Г., Юрачковский Ю.П.* Моделирование сложных систем по экспериментальным данным. - М.: Радио и связь, 1987 г. - 120 с.

5. *Брэгг Р.* Network Security: The Complete Reference / *Роберт Брэгг, Марк Родс-Оусли, Кит Страссберг.* – М.: Эком, 2006. – 912 с.