

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 23  
2017  
№ 1

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ  
Издательство "Новые технологии"

## СОДЕРЖАНИЕ

### ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Разоренов А. А., Фомичев В. А. Новый подход к формализации семантической обработки предписаний на основе теории К-представлений . . . . . 3
- Наумова В. В., Дьяков С. Е., Платонов К. А. Технологии поиска и представления информации в научной инфраструктуре "Геология Дальнего Востока России" . . . . . 15

### МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Булычев Г. Г. Метод пространственных характеристик в задачах механики деформируемого твердого тела. Часть I . . . . . 21

### ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

- Стюгин М. А. Метод построения неразличимого исполняемого кода . . . . . 28

### БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Щеглов К. А., Щеглов А. Ю. Эксплуатационная безопасность. Моделирование реализации угроз атак потенциальным нарушителем . . . . . 34
- Костин В. Н. Синтез оптимального размещения технических средств систем физической защиты критически важных объектов . . . . . 41

### ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- Скорородов А. В. Применение метода Виолы—Джонса для обнаружения атмосферных гравитационных волн над водной поверхностью по спутниковым снимкам MODIS . . . . . 50
- Полищук Ю. В. О способе мониторинга информационной энтропии больших технических систем . . . . . 56
- Чинков М. Ю. Отслеживание статуса сервисов инфраструктуры веб-приложения . . . . . 62

### Журнал в журнале

### НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Харламов А. А. Ассоциативная память — нейросетевая среда для формирования единого пространства знаний . . . . . 66
- Пекунов В. В. Извлечение информации из нейронных сетей прямого распространения в виде простых алгебраических моделей . . . . . 76

Главный редактор:  
СТЕМПКОВСКИЙ А. Л.,  
акад. РАН, д. т. н., проф.

Зам. главного редактора:  
ИВАННИКОВ А. Д., д. т. н., проф.  
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:  
БЫЧКОВ И. В., акад. РАН, д. т. н.  
ЖУРАВЛЕВ Ю. И.,  
акад. РАН, д. ф.-м. н., проф.  
КУЛЕШОВ А. П.,  
акад. РАН, д. т. н., проф.  
ПОПКОВ Ю. С.,  
акад. РАН, д. т. н., проф.  
РУСАКОВ С. Г.,  
чл.-корр. РАН, д. т. н., проф.  
РЯБОВ Г. Г.,  
чл.-корр. РАН, д. т. н., проф.  
СОЙФЕР В. А.,  
акад. РАН, д. т. н., проф.  
СОКОЛОВ И. А., акад.  
РАН, д. т. н., проф.  
СУЕТИН Н. В., д. ф.-м. н., проф.  
ЧАПЛЫГИН Ю. А.,  
акад. РАН, д. т. н., проф.  
ШАХНОВ В. А.,  
чл.-корр. РАН, д. т. н., проф.  
ШОКИН Ю. И.,  
акад. РАН, д. т. н., проф.  
ЮСУПОВ Р. М.,  
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:  
АВДОШИН С. М., к. т. н., доц.  
АНТОНОВ Б. И.  
БАРСКИЙ А. Б., д. т. н., проф.  
ВАСЕНИН В. А., д. ф.-м. н., проф.  
ВИШНЕКОВ А. В., д. т. н., проф.  
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.  
ДОМРАЧЕВ В. Г., д. т. н., проф.  
ЗАБОРОВСКИЙ В. С., д. т. н., проф.  
ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.  
ЗАРУБИН В. С., д. т. н., проф.  
КАРПЕНКО А. П., д. ф.-м. н., проф.  
КОЛИН К. К., д. т. н., проф.  
КУЛАГИН В. П., д. т. н., проф.  
КУРЕЙЧИК В. В., д. т. н., проф.  
ЛЬВОВИЧ Я. Е., д. т. н., проф.  
МАРТЫНОВ В. В., д. т. н., проф.  
МИХАЙЛОВ Б. М., д. т. н., проф.  
НЕЧАЕВ В. В., к. т. н., проф.  
ПОЛЕЩУК О. М., д. т. н., проф.  
САКСОНОВ Е. А., д. т. н., проф.  
СОКОЛОВ Б. В., д. т. н., проф.  
ТИМОНИНА Е. Е., д. т. н., проф.  
УСКОВ В. Л., к. т. н. (США)  
ФОМИЧЕВ В. А., д. т. н., проф.  
ШИЛОВ В. В., к. т. н., доц.

Редакция:  
БЕЗМЕНОВА М. Ю.  
ГРИГОРИН-РЯБОВА Е. В.  
ЛЫСЕНКО А. В.  
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.  
Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.  
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

**CONTENTS**

**INTELLIGENT SYSTEMS AND TECHNOLOGIES**

- Razorenov A. A., Fomichov V. A.** A New Approach to Formalization of Instructions' Semantic Processing Based on the Theory of K-representations . . . . . 3
- Naumova V. V., Dyakov S. E., Platonov K. A.** Technologies of Search and Submission of Information in the Scientific Infrastructure "Geology of the Russian Far East" . . . . . 15

**MODELING AND OPTIMIZATION**

- Bulychev G. G.** Method of Spatial Characteristics in Problems of a Mechanics of a Deformable Solid Body. Part 1 . . . . . 21

**DIGITAL PROCESSING OF SIGNALS AND IMAGES**

- Styugin M. A.** Indistinguishable Executable Code Generation Method . . . . . 28

**CRYPTOSAFETY INFORMATION**

- Shcheglov K. A., Shcheglov A. Yu.** Operational Safety. Potential Offender Attack Threats Implementation Modeling . . . . . 34
- Kostin V. N.** Synthesis of Optimal Placement of Technical Means of the Systems of Physical Protection Critical Facilities . . . . . 41

**APPLIED INFORMATION TECHNOLOGIES**

- Skorokhodov A. V.** Application of Viola—Jones Method for Atmospheric Gravity Waves Detection above the Water Surface on Modis Satellite Imagery . . . . . 50
- Polishuk Yu. V.** On the Method of Monitoring Information Entropy the Large Technical Systems . . . . . 56
- Chinkov M. Yu.** Tracking Services State in Web-application Infrastructure . . . . . 62

**Journal-in-journal**

**NEUROTECHNOLOGIES**

- Kharlamov A. A.** Associative Memory as a Neuronetwork Environment of Combined Knowledge Space . . . . . 66
- Pekunov V. V.** The Derivation of the Information from Artificial Feed-Forward Neural Networks in the Form of the Simple Algebraic Models . . . . . 76

**Editor-in-Chief:**

Stempkovsky A. L., Member of RAS,  
Dr. Sci. (Tech.), Prof.

**Deputy Editor-in-Chief:**

Ivannikov A. D., Dr. Sci. (Tech.), Prof.  
Filimonov N. B., Dr. Sci. (Tech.), Prof.

**Chairman:**

Bychkov I. V., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Zhuravljov Yu. I., Member of RAS,  
Dr. Sci. (Phys.-Math.), Prof.  
Kuleshov A. P., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Popkov Yu. S., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Rusakov S. G., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Ryabov G. G., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Soifer V. A., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Sokolov I. A., Member of RAS,  
Dr. Sci. (Phys.-Math.), Prof.  
Suetin N. V.,  
Dr. Sci. (Phys.-Math.), Prof.  
Chaplygin Yu. A., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Shakhnov V. A., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Shokin Yu. I., Member of RAS,  
Dr. Sci. (Tech.), Prof.  
Yusupov R. M., Corresp. Member of RAS,  
Dr. Sci. (Tech.), Prof.

**Editorial Board Members:**

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.  
Antonov B. I.  
Barsky A. B., Dr. Sci. (Tech.), Prof.  
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.  
Vishnekov A. V., Dr. Sci. (Tech.), Prof.  
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.  
Domrachev V. G., Dr. Sci. (Tech.), Prof.  
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.  
Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.  
Zarubin V. S., Dr. Sci. (Tech.), Prof.  
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.  
Kolin K. K., Dr. Sci. (Tech.)  
Kulagin V. P., Dr. Sci. (Tech.), Prof.  
Kureichik V. V., Dr. Sci. (Tech.), Prof.  
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.  
Martynov V. V., Dr. Sci. (Tech.), Prof.  
Mikhailov B. M., Dr. Sci. (Tech.), Prof.  
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.  
Poleschuk O. M., Dr. Sci. (Tech.), Prof.  
Saksonov E. A., Dr. Sci. (Tech.), Prof.  
Sokolov B. V., Dr. Sci. (Tech.)  
Timonina E. E., Dr. Sci. (Tech.), Prof.  
Uskov V. L. (USA), Dr. Sci. (Tech.)  
Fomichev V. A., Dr. Sci. (Tech.), Prof.  
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

**Editors:**

Bezmenova M. Yu.  
Grigorin-Ryabova E. V.  
Lysenko A. V.  
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.  
According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

А. А. Разоренов, аспирант, e-mail: arazorenov@hse.ru,

В. А. Фомичев, д-р техн. наук, проф., e-mail: vfomichov@hse.ru,

Национальный исследовательский университет "Высшая школа экономики", Москва

## Новый подход к формализации семантической обработки предписаний на основе теории К-представлений

*Предлагается новый подход к разработке алгоритмов семантического анализа (парсинга) естественно-языковых предписаний с использованием формальных средств представления входных, промежуточных и выходных данных. Основной теоретический результат заключается в разработке оригинального, широко применимого алгоритма семантического анализа предписаний. Для построения семантических представлений (СП) предписаний алгоритм использует средства теории К-представлений (концептуальных представлений) — единственной сегодня теории, удобной для построения СП произвольно сложных предписаний. Алгоритм использует также компактную формальную модель лингвистической базы данных (ЛБД) и понятие графа семантико-синтаксических связей (ГССС). Модель ЛБД и понятие ГССС модифицированы по сравнению с их описаниями в предыдущей статье авторов в журнале "Информационные технологии" (2016 год, № 12). По сравнению с известными алгоритмами новый алгоритм может обрабатывать значительно более сложные предписания, образованные несколькими глаголами с зависимыми словами. Вместе с тем новый алгоритм является весьма компактным благодаря использованию результатов теории графов. Предложенный алгоритм был использован в проектировании практически полезного естественно-языкового интерфейса файлового менеджера. Для этого на языке Haskell разработан файловый менеджер NLC-2 (Natural Language Commander — версия 2).*

**Ключевые слова:** *естественный язык, семантический анализ предписаний, семантическое представление, теория К-представлений, СК-язык, словарь лингвистических фреймов, лингвистическая база данных, граф семантико-синтаксических связей, естественно-языковое управление файловой системой, Natural Language Commander*

### Введение

В последнее десятилетие можно было наблюдать смещение интересов многих исследователей в компьютерной лингвистике от статистических методов обработки, ориентированных на лексику и синтаксис текстов естественного языка (ЕЯ), к методам, предполагающим извлечение смысла из ЕЯ-текстов и отображение его в виде некоторой формальной структуры — семантического представления (СП) исходного текста.

Основными причинами такого смещения интересов являются потребности разработки вопросно-ответных систем, способных анализировать полные ЕЯ-тексты из сети Интернет, межъязыкового семантического информационного поиска [1–3], создания Мультилингвистического Семантического Веба [4, 5], конструирования компьютерных интеллектуальных систем (КИС), пополняющих и актуализирующих онтологии посредством извлечения знаний из ЕЯ-текстов [6], ЕЯ-интерфейсов для взаимодействия с онтологиями и мультимедийными базами данных [7–10], а также ЕЯ-интерфей-

сов КИС, базирующихся на действиях (action-based applied intelligent systems) [11]; ряд дополнительных ссылок можно найти в статье [12].

Потребность в создании семантико-синтаксических анализаторов ЕЯ-текстов возникает во многих предметных областях. Поэтому актуальна проблема разработки методов проектирования переносимых алгоритмов семантического анализа (АСА).

Изучение публикаций по компьютерной лингвистике показывает, что, как правило, АСА не описываются достаточно полно (скрыты в программном коде). Публикации излагают ключевые идеи, они поясняются рисунками и таблицами. Обычно научные работы не дают информации для непосредственной разработки компьютерных программ, разные исследователи по одной и той же публикации построят сильно различающиеся компьютерные программы.

В этом отношении довольно типичным является алгоритм Sentence, представленный в монографии [13]. В этой книге алгоритм занимает примерно одну страницу. Одной из его отличительных

черт является отсутствие привязки к какой-либо лингвистической базе данных (ЛБД). Алгоритм включает следующую процедуру *noun*: *начало получить понятие для существительного конец*. Эта тривиальная процедура не учитывает, что существительное может быть связано с несколькими семантическими единицами, и для выбора конкретной схемы нужно анализировать контекст, задаваемый глаголом. Столь же проста процедура *verb*: *начало получить надежный фрейм для глагола конец*. Эта процедура не учитывает, что глагол может иметь несколько значений.

Суть ситуации в том, что для программиста, желающего использовать процедуру *Sentence* для разработки АСА, ничего не изменится, если алгоритм *Sentence* заменить указанием "Разработать алгоритм семантического анализа предложений".

Качественный скачок в разработке переносимых АСА в русскоязычной научной литературе произошел в 2005 г. после выхода статьи [14] и монографии [15], а в англоязычной литературе — в 2010 г. после издания монографии [16]. В этих книгах были описаны соответственно АСА *SemSyn* и *SemSynt1*. АСА *SemSyn* является композицией алгоритмов *BuildMatr* и *BuildSem*. Первый алгоритм по входному ЕЯ-тексту *T* строит его семантико-синтаксическое представление в виде специальной строково-числовой матрицы *Matr*, называемой *матричным семантико-синтаксическим представлением (МССП)* текста *T*. Алгоритм *BuildSem* преобразует МССП *Matr* в СП текста *T*.

Алгоритмы *SemSyn* и *SemSynt1* являются важными компонентами теории К-представлений (ТКП). Это оригинальная теория проектирования семантико-синтаксических анализаторов ЕЯ-текстов с широким использованием формальных средств представления входных, промежуточных и выходных данных [14–33]; дополнительные ссылки можно найти в работах [12, 32, 33].

Базовая математическая модель ТКП описывает систему, состоящую из 10 частичных операций на концептуальных структурах. Эта модель включает определение нового класса формальных языков, называемых СК-языками (стандартными концептуальными языками) [15, 16, 23–30]. ТКП является главной составляющей интегральной формальной семантики ЕЯ — научного направления на стыке математической информатики, математической лингвистики и компьютерной лингвистики [20, 22].

СП входного текста *T*, строящееся алгоритмами *SemSyn* и *SemSynt1*, является выражением используемого СК-языка, т. е. возможным К-представлением (КП) входного текста *T*.

Для алгоритмов *SemSyn* и *SemSynt1* характерен высокий уровень детализации основных процедур. Каждый из этих алгоритмов можно непосредственно запрограммировать, добавляя для конкретного

входного языка подсистему морфологического анализа (в этом направлении есть множество научных публикаций). Однако объем описания (вместе со структурами данных) каждого из алгоритмов *SemSyn* и *SemSynt1* необычно высок, более ста печатных страниц. Это вызывает объективные трудности использования этих алгоритмов в учебном процессе.

Данная статья продолжает линию работ [12, 33–37] и составляет единое целое с публикацией [12]. Статья имеет следующую структуру. В разделах 1–2 поясняется смысл двух компонентов формальной модели ЛБД, введенной в работе [12]: морфологического базиса и множества шаблонов К-представлений. Модифицируется понятие морфологического базиса. В разделе 3 расширяется формальная модель ЛБД: добавляется компонент, называемый терминологической базой данных. В разделе 4 модифицируется и иллюстрируется понятие графа семантико-синтаксических связей — новой формы семантико-синтаксического представления предписания, предложенной в работе [12]. В разделах 5 и 6 описывается новый алгоритм семантического анализа предписаний. Раздел 7 посвящен математическому исследованию свойств предложенного алгоритма. Раздел 8 описывает применение разработанного алгоритма к проектированию файлового менеджера с ЕЯ-интерфейсом NLC-2 (Natural Language Commander — Version 2).

## 1. Модифицированное определение класса морфологических базисов

В статье [12] предложена компактная формальная модель ЛБД, одним из ее компонентов является определение морфологического базиса (м.б.). Каждый м.б. *Morph* является упорядоченной десяткой формальных объектов вида

$(Wd, Wsep, Wconn, Wprep, Mprop, M, < =:, rm, rc, ru), (1)$

компоненты этой десятки взаимосвязаны определенными условиями. Каждый м.б. задается в контексте предварительно выбранного концептуального базиса (см. [12]). Интерпретация компонентов м.б. рассматривается ниже.

Компоненты *Wd*, *Wsep*, *Wconn*, *Wprep* являются конечными множествами символов, интерпретируемыми следующим образом. *Wd* — это множество слов и коротких словосочетаний ("мастер спорта", "органическая химия" и т.д.); *Wsep* — множество разделителей (точка, запятая и др.); *Wconn* — множество логических связей, в том числе ИЛИ, И; *Wprep* — расширенное множество предлогов, включающее, в частности, сочетание "в течение" и пустой предлог *nil*.

Элементы конечного множества символов *Mprop* интерпретируются как значения морфологических признаков, соответствующих лексическим едини-

цам входного языка: названия частей речи и их подклассов, значения грамматического падежа и числа для существительных и прилагательных, значения залога и времени для глаголов и т. д.

Шестым и седьмым компонентами м.б. вида (1) являются конечное множество  $M$  и частичный порядок  $< =$ : на  $M$ . Множество  $M = \{S_1, \dots, S_n\}$ , где  $n > 1$  и для каждого  $k = 1, \dots, n$  элемент  $S_k$  — подмножество множества  $Mprop$ . При этом  $S_k$  интерпретируется как совокупность значений морфологических признаков, которая может быть ассоциирована с каким-либо словом  $wrd$  входного языка в результате полного или частичного морфологического анализа этого слова. Например, пусть  $wrd = \text{"папку"}$ , тогда со словом  $wd$  могут быть ассоциированы множества  $\{\text{сущ}\}$ ,  $\{\text{сущ}, \text{сущ-нарицат}\}$ ,  $\{\text{сущ}, \text{сущ-нарицат}, \text{ед. число}\}$ ,  $\{\text{сущ}, \text{сущ-нарицат}, \text{ед. число}, \text{вин. падеж}\}$ ,  $\{\text{сущ}, \text{сущ-нарицат}, \text{ед. число}, \text{вин. падеж}, \text{жен. род}\}$ . Из этих множеств только последнее является результатом полного морфологического анализа слова "папку".

Факт  $S_i < =: S_k$ , где  $S_i, S_k$  принадлежат  $M$  (т. е. множества  $S_i$  и  $S_k$  связаны частичным порядком  $< =$ ) означает, что множество  $S_i$  совпадает с множеством  $S_k$  или  $S_k$  передает больше информации по сравнению с  $S_i$ . Например, если  $S_i = \{\text{сущ}, \text{сущ-нарицат}\}$ ,  $S_k = \{\text{сущ}, \text{сущ-нарицат}, \text{ед. число}\}$ , то  $S_i < =: S_k$ ,  $S_k < =: S_k$  (так как частичный порядок является рефлексивным отношением) и  $S_i < =: S_k$ .

Последние три компонента морфологического базиса — это функции, заданные на множестве  $W$ -объединении множеств  $Wd, Wsep, Wconn, Wprep$ . Пусть  $unit$  — произвольный элемент множества  $W$ , тогда  $rm(unit)$  — множество возможных коллекций морфологических признаков (т.е.  $rm(unit)$  — подмножество множества  $Mprop$ ) или пустое множество. Если  $unit$  — лексическая единица из  $Wd$ , то  $rc(unit)$  — это базовая морфологическая форма единицы  $unit$ . Если  $unit$  входит в  $W$ , то  $ru(unit)$  — множество, содержащее одну или несколько семантических единиц из первичного информационного универсума  $X(B)$ , где  $B$  — рассматриваемый концептуальный базис (к.б.). Например,  $ru(\text{мышью}) = \{\text{мышь-комп}, \text{мышь-биолог}\}$ .

В процессе разработки алгоритма семантического анализа предписаний возникла потребность в модификации определения м.б. посредством расширения области значений отображения  $ru$ . В работе [12] компонент м.б. вида (1)  $ru$  является функцией из множества элементарных значащих единиц текстов  $W$  в булеан  $X(B)$ , где  $B$  — рассматриваемый концептуальный базис,  $X(B)$  — первичный информационный универсум к.б.  $B$ , булеан  $X(B)$  — множество, состоящее из всех подмножеств множества  $X(B)$ , в частности, включающее  $X(B)$  и пустое множество. Таким образом, для произвольной единицы текста  $wrd$  из  $W$  выполняется соотношение  $ru(wrd) = \{Set_1, \dots, Set_n\}$ , где  $n > 1$  и для каждого

$k = 1, \dots, n$  элемент  $Set_k$  — подмножество множества  $X(B)$ .

В данной статье мы будем считать, что  $ru$  является функцией из  $W$  в булеан  $Ls(B)$ , где  $Ls(B)$  — СК-язык в базисе  $B$ .

**Пример.** Если множество  $W$  включает словоформы  $wrd1 = \text{папку}$ ,  $wrd2 = \text{мышь}$ , то могут выполняться соотношения  $ru(wrd1) = \{\text{нек папка1}, \text{нек папка2}\}$ ,  $ru(wrd2) = \{\text{нек мышь-комп}, \text{нек мышь-биолог}\}$ , где  $нек$  — квантор референтности концептуального базиса  $B$  (интерпретируется как семантическая единица — значение слова "некоторый" для русского языка и сочетания "a certain" для английского языка),  $папка1$  — обозначение понятия "канцелярская папка",  $папка2$  — обозначение понятия "компьютерная папка", т. е. обозначение некоторой структуры данных.

## 2. Структура шаблонов К-представлений

В работе [12] введено понятие шаблона К-представления. Множество всех таких шаблонов задается рассматриваемым к.б.  $B$  и обозначается  $Templates(B)$ . Это множество состоит из упорядоченных троек вида  $(frame, x1, x2)$ , где  $x1, x2$  — переменные к.б.  $B$ , а  $frame$  — некоторая цепочка символов, которая может быть преобразована в выражение СК-языка  $Ls(B)$  заменой вхождений символов  $x1$  и  $x2$  на некоторые выражения (К-цепочки) языка  $Ls(B)$ .

Простейшими выражениями СК-языка  $Ls(B)$  являются элементы первичного информационного универсума  $X(B)$  и переменные из  $V(B)$ . Структурированные выражения получаются комбинированным применением частичных операций  $Op[1], \dots, Op[10]$ , задаваемых правилами  $P[1], \dots, P[10]$ . Кратко рассмотрим суть этих операций.

Выражения СК-языков строятся из первичных информационных единиц, задаваемых концептуальным базисом, и нескольких служебных символов с помощью индуктивного применения оригинальных правил (утверждений)  $P[0], P[1], \dots, P[10]$ . Формальный язык, порожденный к.б.  $B$  с помощью этих правил, обозначается  $Ls(B)$ .

Правило  $P[1]$  позволяет соединять интенциональные кванторы и обозначения (простые или составные) понятий. В частности, позволяет строить выражения  $нек файл1, нек файл1 *$  (Расширение-прогр, "doc"),  $все файл1 *$  (Расширение-прогр, "doc"). Правило  $P[2]$  используется для конструирования выражений вида  $f(t_1, \dots, t_n)$ , где  $f$  — имя функции,  $t_1, \dots, t_n$  — обозначения  $n$  аргументов функции (пример:  $Дата-создания(нек файл1)$ ). Правило  $P[3]$  позволяет строить выражения вида  $(a \equiv b)$ , где  $a$  и  $b$  — К-цепочки. Пример:  $(документ1 \equiv файл1 *$  (Расширение-прогр, "doc")).

С помощью правила  $P[4]$  строятся выражения вида  $rel(t_1, \dots, t_n)$ , где  $rel$  — имя отношения с  $n$  атрибутами (пример:  $Раньше(Дата-создания(нек файл1), Вчера (\#сейчас\#))$ ). Правило  $P[5]$  дает нам возмож-

ность пометить переменной формулу или ее часть. Пример: *все файл1\*(Расширение, "doc"): S1*.

Правило P[6] позволяет присоединять связку отрицания ( $\neg$ ) к формулам (пример:  $\neg$ файл1). Правило P[7] управляет использованием логических связок  $\wedge$  (И) and  $\vee$  (ИЛИ). Пример: *файл1\*(Расширение-прогр, ("doc"  $\vee$  "docx"))*. Применяя правило P[8] на последнем шаге вывода, можно строить компактные обозначения понятий. Пример: *файл1\*(Расширение-прогр, ("doc"  $\vee$  "docx"))(Локализация, нек рабоч-стол)*.

Правило P[9] позволяет использовать в формулах кванторы всеобщности и существования ( $\forall$  и  $\exists$ ). Правило P[10] дает возможность строить СП конечных последовательностей (упорядоченных наборов) как цепочки вида  $\langle c_1, \dots, c_n \rangle$ , где  $c_1, \dots, c_n$  — обозначения элементов последовательности. Пример:  $\langle$ Источник1, нек архив2\*(Расположение, нек папка2\*(Наименование1, "Текущий год")) $\rangle$ . Это правило может быть использовано, в частности, для представления координатами точек в пространстве (пример:  $\langle 16, 52, 23 \rangle$ ).

**Пример.** Предписание "Заархивировать документы в папке "Проект" и переслать на адрес somebody@example.org" может быть ассоциировано с КП *Semrepr1* вида

*(Выполнение Действия(#сейчас#, нек архивирование1\*(Объект, нек конечн.множ\*(Качеств-состав, документ1\*(Локализация1, нек папка2\*(Наименование1, "Проект"))): S1)(Результат1, x1), e1)  $\wedge$  ВыполнениеДействия(#сейчас#, нек электрон-пересылка\*(Объект1, x1)(Электрон-адрес, "somebody@example.org"), e2)  $\wedge$  Непосредственно-после(e2, e1))*.

### 3. Расширение формальной модели лингвистической базы данных

В предписаниях часто встречаются термины, которые могут интерпретироваться компьютерной системой только после их экспликации. Например, пусть входом является предписание "Скопируй музыкальные файлы из папки "Download" в папку с именем "Моя музыка". Тогда для выполнения этой команды нужно заменить термин "музыкальный файл" на термин "файл с одним из расширений "mp3", "ogg", "wav", "aac". Поэтому одним из компонентов лингвистической базы данных должна являться *терминологическая база данных*.

**Определение.** *Экспликативной лингвистической базой данных* называется упорядоченная четверка *LingDbase* вида  $(B, Morph, LexFrames, TermsBase)$ , где  $B$  — произвольный к.б., *Morph* — произвольный морфологический базис, согласованный с к.б.  $B$ , *LexFrames* — некоторый словарь лексических фреймов, согласованный с к.б.  $B$  и морфологиче-

ским базисом *Morph*, и *TermsBase* — конечное множество, состоящее из выражений вида

$$(a \equiv b * (rel_1, d_1) \dots (rel_n, d_n)),$$

где  $a$  и  $b$  — такие элементы первичного информационного универсума  $X(B)$ , что их типы  $tp(a)$  и  $tp(b)$  начинаются с символа  $\uparrow$  (этот символ сигнализирует о том, что это типы понятий, а не предметов), и цепочка  $b * (rel_1, d_1) \dots (rel_n, d_n)$ , где  $n \geq 1$ , является выражением СК-языка  $Ls(B)$ , полученным в результате одного применения правила P[8]. Множество *TermsBase* называется *терминологической базой данных* (т.б.д.).

Таким образом, цепочка  $b * (rel_1, d_1) \dots (rel_n, d_n)$ , является составным обозначением понятия.

В этом определении термин "морфологический базис" понимается в модифицированном виде, предложенном выше в разделе 1.

**Пример.** Т.б.д. *TermsBase* может, в частности, включать выражения

*(музык.файл  $\equiv$  файл1\*(Расширение-прогр, ("mp3"  $\vee$  "ogg"  $\vee$  "wav"  $\vee$  "aac")))*,

*(документ1  $\equiv$  файл1\*(Расширение-прогр, ("doc"  $\vee$  "docx"  $\vee$  "odt")))*.

### 4. Модификация и пояснение понятия графа семантико-синтаксических связей

В подразделе 3.2. статьи [12] понятие шаблона К-представления используется для определения класса формальных объектов, называемых *словарями лексических фреймов* (с.л.ф.). Каждый с.л.ф. является конечным множеством *LexFrames*, состоящим из упорядоченных семерок формальных объектов. Одним из элементов семерки является некоторый шаблон КП.

Каждая семерка из множества *LexFrames* задает необходимые условия существования семантической связи в предложении между двумя лексическими единицами. При этом предъявляются определенные требования как к наборам значений морфологических признаков, так и к семантическим характеристикам лексических единиц. Ниже рассматривается пример, иллюстрирующий структуру лексического фрейма.

Компактная формальная модель ЛБД, построенная в статье [12], позволила предложить в той же работе новую интерпретацию понятия "семантико-синтаксическое представление естественно-языкового текста". С этой целью введено понятие графа семантико-синтаксических связей (ГССС) входного текста для ЛБД *Lingdb* =  $(B, Morph, LexFrames)$ . Использование понятия ГССС при разработке алгоритма семантического анализа предписаний вызвало необходимость модификации этого поня-

тия. С этой целью используем одно из фундаментальных свойств СК-языков.

В монографиях [15, 16, 30] для произвольного к.б.  $B$  определено отображение

$$tpl: Ls(B) \rightarrow Tp(S(B)),$$

где  $Ls(B)$  — СК-язык в базисе  $B$ ,  $S(B)$  — сортовая система, являющаяся первым компонентом к.б.  $B$ ,  $Tp(S(B))$  — множество типов, задаваемое сортовой системой  $S(B)$ . Первичный информационный универсум  $X(B)$  и множество переменных  $V(B)$  являются подмножествами СК-языка  $Ls(B)$ ; на элементах этих подмножеств отображение  $tpl$  принимает те же значения, что и отображение  $tp$  из объединения  $X(B)$  и  $V(B)$  в множество типов  $Tp(S(B))$ . Легко заметить, что для произвольного к.б.  $Tp(S(B)) = Types(B)$ , где  $Types(B)$  — обозначение, введенное в разделе 1 работы [12].

В настоящей статье мы модифицируем понятие ГССС, используя в пунктах 1 и 3 его определения (см. раздел 4 работы [12]) отображение  $tpl: Ls(B) \rightarrow Tp(S(B))$  вместо отображения  $tp$ .

Понятие ГССС играет ключевую роль в описываемом ниже алгоритме семантического анализа предписаний SemSynRA. Поясним суть этого понятия.

**Пример.** Пусть входом является ЕЯ-текст  $T = t_1, \dots, t_n$ , где  $n > 1$ , и для  $k = 1, \dots, n$   $t_k$  — элементарная значащая единица текста из множества  $W$ -объединения множеств  $Wd, Wsep, Wconn, Wprep$  (см. раздел 1). Тогда ГССС текста  $T$  является ориентированным графом с вершинами, имеющими метки  $1, \dots, n$ . Поясним на примере условия, позволяющие соединить ребром вершины графа с метками  $i$  и  $j$ . Эти условия задаются какой-то конкретной упорядоченной семеркой вида

$$(type_1, morph_1, type_2, morph_2, prep, template, dir), (2)$$

из словаря лексических фреймов *LexFrames*.

Пусть  $T1 =$  "Переместить (1) файл (2) "a.txt" (3) в (4) папку (5) "Документы" (6). Пусть  $i = 1, j = 5$ , тогда  $t_i =$  "Переместить",  $t_j =$  "папку".

**Условие 1.** Пусть  $Setsem1$  — это конечное множество, состоящее из всех семантических единиц, которые могут быть значениями слова  $t_i$ . Тогда в множество  $Setsem1$  должен входить такой элемент  $sem1$ , что его тип является конкретизацией типа единицы  $t_i$  (в частности, совпадает с типом единицы  $t_i$ ).

Допустим, что для некоторой ЛБД1 выполняются соотношения  $ru(переместить) = \{нек перемещение1\}$ ,  $sem1 = нек перемещение1$ ,  $Setsem1 = \{sem1\}$ ,  $tpl(sem1) = действие$ . Пусть также множество *LexFrames* включает набор

$$(действие, \{глаго\}, простр.объект, \{сущ, вин.над.\}, v, (3))$$

$$(нек x1*(Назначение1, x2), x1, x2), 0).$$

Тогда  $type_1 = действие$ , поэтому  $type_1 = tpl(sem1)$ , т. е. тип  $tpl(sem1)$  является конкретизацией типа  $type_1$ . Поэтому Условие 1 выполнено.

Лингвистические базы данных, сформированные разными группами специалистов, могут иметь отличия. Поэтому предположим, что для некоторой ЛБД2  $tpl(нек перемещение1) = tpl(sem1) = физич.действие$  и сорт  $физич.действие$  является конкретизацией сорта  $действие$ . Тогда и в этом случае будет выполнено Условие 1.

**Условие 2.** Слово "переместить" является глаголом. Поэтому элемент  $morph_1 = \{глаго\}$  лексического фрейма вида (3) задает такое требование к возможным коллекциям значений морфологических признаков слова  $t_i$ , которое выполнено.

**Условия 3 и 4.** В лексическом фрейме (3) можно следующим образом выделить компоненты  $type_2, morph_2$  набора вида (2):  $type_2 = простр.объект$ ,  $morph_2 = \{сущ, вин.над.\}$ . В рассматриваемом примере  $t_j =$  "папку", где  $j = 5$ . Пусть для некоторой ЛБД1 выполняются следующие соотношения:

$$\begin{aligned} папку &— элемент множества  $Wd, ru(папку) = \\ &= \{нек папка1, нек папка2\}, tpl(нек папка1) = \\ &= дин.физ.об, sem2 = нек папка2, tpl(sem2) = \\ &= простр.объект * структ.данных. \end{aligned}$$$

Тогда тип  $tpl(sem2)$  — конкретизация типа  $type_2$ , поэтому выполнено Условие 3.

Пусть для ЛБД1  $ru(папку) = \{y\}$ , где  $y = \{сущ, сущ-нарицат, вин.над, ед.число\}$ . Тогда, очевидно, для множества значений морфологических признаков  $y$  имеет место соотношение  $morph_2 \leq y$ . Поэтому выполнено и Условие 4.

**Условие 5.** Пусть  $k = 4$ , тогда  $rc(t_k) = -$  это предлог "в"; так как  $i = 1, j = 5$ , то  $i < k < j$ . Поэтому выполнено и Условие 5.

Таким образом, в графе семантико-синтаксических связей для рассматриваемого предписания  $T1 =$  "Переместить (1) файл (2) "a.txt" (3) в (4) папку (5) "Документы" (6) можно провести ребро из вершины с меткой 1 в вершину с меткой 5. Извлекая данные из лексического фрейма вида (3), это ребро можно ассоциировать с упорядоченной четверкой

$$(нек перемещение1, нек папка2,4, (нек x1*(Назначение1, x2), x1, x2)).$$

## 5. Алгоритм семантического анализа предписаний

Изложенные выше идеи лежат в основе разработанного алгоритма семантического анализа предписаний *SemSyntRA*, представленного ниже. Этот алгоритм преобразует предписания на ограниченном естественном языке в К-представления.

## Описание алгоритма SemSyntRA

**Назначение:** преобразование ЕЯ-текста, являющегося предписанием, в множество всех возможных К-представлений.

**Вход:**  $T$  — текст;  $LingDbase$  — лингвистическая база данных.

**Выход:**  $Result$  — список К-представлений.

### Внешние спецификации вспомогательных алгоритмов

#### Спецификация процедуры Tokenize

**Назначение:** разделение текста на элементарные значащие единицы.

**Вход:**  $T$  — текст.

**Выход:**  $WT$  — массив элементарных значащих единиц текста.

**Комментарий:** Алгоритм зависит от входного языка и должен быть разным, например, для русского и английского языков. Поэтому в данной работе он не рассматривается.

#### Спецификация процедуры BuildGSSR

**Назначение:** построение графа семантико-синтаксических связей (ГССС) входного текста.

**Вход:**  $WT = wt_1, \dots, wt_n$  — массив единиц текста из  $Wd$ , несущих смысловую нагрузку;

$LingDbase$  — экспликативная лингвистическая база данных.

**Выход:**  $GR$  — ГССС входного текста.

#### Алгоритм BuildGSSR

начало {BuildGSSR}

$E := \{i \mid i = 1, \dots, n: wt_i \text{ belongs to } Wd(Morph(LingDbase))\};$

$V :=$  пустой список;

цикл по  $i$  от 1 до  $n$

    цикл по  $j$  от 1 до  $n$

        если для пары  $(wt_i, wt_j)$  существуют элементы словаря  $LexFrames(LingDbase)$ , удовлетворяющие условиям определения 3 из [12], то  
            добавить в список  $V$  соответствующие кортежи  $(i, j, (sem1, sem2, k, templ))$ ;

        конец если;

    конец цикла по  $j$ ;

конец цикла по  $i$ ;

$GR := (E, V)$ ;

конец {BuildGSSR}.

#### Спецификация процедуры BuildSpanningTrees

**Назначение:** построение списка остовных деревьев по графу семантико-синтаксических связей (ГССС) входного текста.

**Вход:**  $GR$  — ГССС входного текста.

**Выход:**  $Ltr$  — список допустимых остовных деревьев.

**Комментарий:** В этой процедуре может быть использован любой алгоритм поиска остовных деревьев в ориентированных графах, возможно, с последующей фильтрацией.

#### Спецификация процедуры BuildRepresentations

**Назначение:** построение возможных К-представлений для остовных деревьев графа семантико-синтаксических связей

**Вход:**  $Ltr$  — список допустимых остовных деревьев

**Выход:**  $Result$  — список К-представлений.

#### Алгоритм BuildRepresentations

начало {BuildRepresentations}

$Result :=$  пустой список;

цикл

    выбрать очередное дерево  $Tree$  из списка  $Ltr$ :

    построить К-представление по дереву  $Tree$ ;

    если получилось, то

        добавить полученное К-представление в список  $Result$ ;

    конец если;

конец цикла по  $Ltr$ ;

конец {BuildRepresentations}.

#### Алгоритм SemSyntRA

начало {SemSyntRA}

    Tokenize( $T, WT$ );

    BuildGSSR( $WT, LingDbase, GR$ );

    BuildSpanningTrees( $GR, Ltr$ );

    BuildRepresentations( $Ltr, Result$ );

Конец {SemSyntRA}.



## 6. Примеры, иллюстрирующие работу алгоритма семантического анализа предписаний

Поясним на примерах этап построения ГССС. Для построения основных деревьев мы использовали алгоритм из работы [38]. Этот алгоритм был модифицирован для обработки ориентированных графов. Алгоритм генерирует всевозможные подграфы, включающие все вершины ГССС, и проверяет, является ли деревом рассматриваемый подграф и все ли его вершины достижимы из определенной корневой вершины.

К-представление по каждому остовному дереву строится подъемом от конечных элементов дерева к его корню — сверткой снизу вверх. На каждом этапе сворачиваются поддеревья единичной глубины. При выполнении свертки сворачиваемое на данном этапе поддерево некоторой вершины заменяется вершиной, которой ставится в соответствие КП, получаемое из шаблона сворачиваемого ребра заменой вхождений переменной  $x_j$  на элемент первичного универсума вершины поддерева и переменной  $x_2$  на КП, связанное с подчиненной вершиной сворачиваемого ребра.

Например, для приведенного выше примера ГССС ребра

$(2, 3, (\text{нек файл}1, "a.txt", 0, (\text{нек } x2^*(\text{Наименование}1, x4), x2, x4)))$

и  $(5, 6, (\text{нек папка}2, "Документы", 0, (\text{нек } x2^*(\text{Наименование}1, x4), x2, x4)))$

свернутся в КП  $\text{нек файл}1^*(\text{Наименование}1, "a.txt")$  и КП  $\text{нек папка}2^*(\text{Наименование}1, "Документы")$  соответственно.

При этом если из вершины сворачиваемого поддерева выходит несколько ребер, то результаты свертки ребер совмещаются. Совмещение КП ребер происходит следующим образом:

1) КП вида  $c^*(r_1, z_1), \dots, c^*(r_n, z_n)$  совмещаются в КП  $c^*(r_1, z_1) \dots (r_n, z_n)$ ;

2) интенциональный квантор  $\text{нек } c$  с КП вида  $c^*(r_1, z_1) \dots c^*(r_n, z_n)$  совмещаются в КП  $\text{нек } c^*(r_1, z_1) \dots (r_n, z_n)$ ;

3) К-цепочки  $a$  и  $(b_1 C b_2 C \dots C b_n)$ , где  $C$  — логическая связка  $\wedge$  (конъюнкция) или  $\vee$  (дизъюнкция),  $n > 1$ , совмещаются в КП  $(a C b_1 C \dots C b_n)$ , если  $\text{tpl}(a) = \text{tpl}(b_1) = \dots = \text{tpl}(b_n)$ .

Вернемся к рассматриваемому примеру. Ребра

$(1, 2, (\text{нек перемещение}1, \text{нек файл}1, 0, (\text{нек } x1^*(\text{Объект}1, x2), x1, x2)))$  и

$(1, 5, (\text{нек перемещение}1, \text{нек папка}2, 4, (\text{нек } x1^*(\text{Назначение}1, x3), x2, x3)))$

дадут

$\text{КП нек перемещение}1^*(\text{Объект}1, \text{нек файл}1^*(\text{Наименование}1, "a.txt"))$

и  $\text{нек перемещение}1^*(\text{Назначение}1, \text{нек папка}2^*(\text{Наименование}1, "Документы"))$ .

Выполнение процедуры совмещения даст КП

$\text{нек перемещение}1^*(\text{Объект}1, \text{нек файл}1^*(\text{Наименование}1, "a.txt"))(\text{Назначение}1, \text{нек папка}2^*(\text{Наименование}1, "Документы"))$ .

Полученное таким образом КП и будет искомым представлением. Критерием отсева "лишних" КП может выступить то, что не все деревья дадут корректные КП. Также можно использовать дополнительную фильтрацию остовных поддеревьев, основанную на следующих наблюдениях: во-первых, все слова между предлогом и связанным с ним существительным будут подчинены существительному. Во-вторых, маловероятно применение кросс-последовательных зависимостей (cross-serial dependencies) в пользовательских инструкциях. Примером такой зависимости может служить ребро  $(2, 6, (\text{нек файл}1, \text{литерал}, 0, (\text{нек } x2^*(\text{Наименование}1, x4), x2, x4)))$  приведенного выше примера ГССС.

Рассмотренный пример не содержал в себе однородных членов предложения. Проанализируем новый пример — предписание

*Переместить(1) файлы(2) "a.txt"(3) и(4) "b.txt"(5) в(6) папку(7) "Документы"(8).*

В этом случае граф семантико-синтаксических связей будет выглядеть следующим образом (множество вершин:  $\{1, 2, 3, 5, 7, 8\}$ ):

{  
 $(1, 2, (\text{нек перемещение}1, \text{нек файл}1, 0, (\text{нек } x1^*(\text{Объект}1, x2), x1, x2)))$ ,  
 $(1, 7, (\text{нек перемещение}1, \text{нек папка}2, 4, (\text{нек } x1^*(\text{Назначение}1, x3), x2, x3)))$ ,  
 $(2, 3, (\text{нек файл}1, "a.txt", 0, (\text{нек } x2^*(\text{Наименование}1, x4), x2, x4)))$ ,  
 $(2, 5, (\text{нек файл}1, "b.txt", 0, (\text{нек } x2^*(\text{Наименование}1, x4), x2, x4)))$ ,  
 $(7, 8, (\text{нек папка}2, "Документы", 0, (\text{нек } x2^*(\text{Наименование}1, x4), x2, x4)))$ ,  
 $(2, 8, (\text{нек файл}1, "Документы", 0, (\text{нек } x2^*(\text{Наименование}1, x4), x2, x4)))$   
}

Ребро  $(2, 8, \dots)$  будет отброшено в силу правила 2). Оставшиеся ребра образуют остовное дерево. Посмотрим, как будет происходить сборка соответствующего КП, точнее, как будет происходить сборка поддерева узла 2. Собирать его нужно, совмещая КП  $\text{нек файл}1^*(\text{Наименование}1, "a.txt")$  и КП  $\text{нек файл}1^*(\text{Наименование}1, "b.txt")$ .

Естественно, что в результате совмещения будет получено КП

нек файл1 \* (Наименование1, ("a.txt"  $\vee$  "b.txt")),

где  $\vee$  — логическая связка  $\wedge$  или  $\vee$ . Надо определиться, какую из связок использовать. Левым и правым числом для поддерева вершины 2 будут 2 и 5. В диапазоне от 2 до 5 есть слово-связка "и" из множества  $W_{conn}$ . Пусть  $ru("и") = \{\wedge\}$ , тогда  $C = \wedge$ . Следовательно, поддереву вершины 2 будет соответствовать КП

нек файл1 \* (Наименование1, ("a.txt"  $\wedge$  "b.txt")),

а итоговое КП будет таким:

нек перемещение1 \* (Объект1, нек файл1 \* (Наименование1, ("a.txt"  $\wedge$  "b.txt")))(Назначение1, некап-ка2\*(Наименование1, "Документы")).

Нужно отметить, что по каждому остовному дереву может быть построено не более одного КП, так как алгоритм построения и разметка ребер не допускают вариативности.

## 7. Исследование свойств предложенного алгоритма

**Теорема 1.** Пусть  $n$  — длина входного текста  $T$  (в элементарных значащих единицах текста), и  $m$  — число элементов словаря лексических фреймов. Тогда для построения графа семантико-синтаксических связей требуется  $O(n^2m)$  операций.

Иными словами, сложность построения графа семантико-синтаксических связей является квадратичной по длине входного текста и линейной по мощности словаря лексических фреймов.

*Доказательство.* Как видно из реализации процедуры BuildGSSR (см. раздел 5), для обработки текста требуется проход по всем несовпадающим парам массива лексических единиц текста — см. вложенные циклы по  $i$  и  $j$  в реализации алгоритма. Число таких пар составляет  $1/2(n^2 - n)$ . Для каждой пары необходимо найти подходящие элементы словаря лексических фреймов, что требует перебора  $m$  элементов списка, если не прибегать к каким-либо эвристикам. Таким образом, алгоритм требует для своей работы  $O(n^2m)$  операций. Что и требовалось доказать.

Оценим сложность построения  $K$ -представления по остовному дереву.

**Теорема 2.** Пусть  $Tree$  — остовное дерево ГССС  $GR = (E, V)$ , где  $E$  — множество вершин, а  $V$  — множество ребер графа  $GR$ . Тогда для построения КП по данному дереву требуется  $O(k)$  операций, где  $k$  — число вершин графа  $GR$ .

*Доказательство.* Для построения КП для дерева  $Tree$  требуется построить КП для каждого из его ребер, число которых равно  $k - 1$ .  $K$ -представление для каждого ребра строится по шаблону, заключенному в разметке данного ребра. Поэтому можно считать, что сложность построения не превышает

некоторой константы  $C$ . Далее требуется выполнить не более чем  $k - 1$  совмещений полученных шаблонов. Итого, не более чем  $2C(k - 1)$  операций. Таким образом получается, что требуется не более чем  $O(k)$  операций для построения КП по остовному дереву. Что и требовалось доказать.

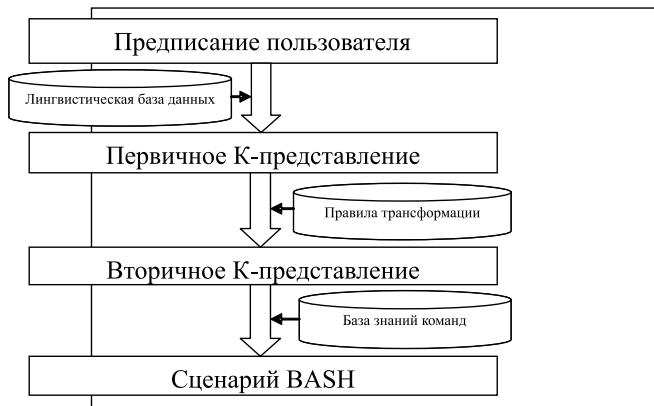
Отсюда естественным образом вытекает идея оценки сложности всего алгоритма преобразования ЕЯ-текста  $T$  в КП: как видно из реализации алгоритма SemSyntRA, алгоритм состоит из последовательного применения процедур Tokenize, BuildGSSR, BuildSpanningTrees, BuildRepresentations. Сложность реализации процедур Tokenize и BuildSpanningTrees зависит от выбранных для их реализации алгоритмов. Обозначим ее как  $A(n)$  и  $B(n)$  соответственно, где  $n$  — число лексических единиц текста  $T$ . Сложность реализации процедур BuildGSSR и одного шага (по входному списку Ltr) BuildRepresentations была оценена в теоремах 1 и 2. Таким образом, суммируя сложность шагов алгоритма SemSyntRA, получаем оценку  $A(n) + O(n^2m) + B(n) + O(dn)$ , где  $d$  — число остовных деревьев для графа семантико-синтаксических связей.

## 8. Создание системы управления файлами NLC-2 на основе разработанных формальных моделей и алгоритма семантического анализа

Актуальным направлением применения алгоритмов семантической обработки предписаний является естественно-языковое управление компьютером или мобильным устройством, например, смартфоном или планшетом. В этом направлении были успешно реализованы такие проекты, как система речевого управления мобильными устройствами Apple Siri [39] и ее русскоязычный аналог Собеседник HD [40]; Braina Project — программное обеспечение для управления компьютером в целом [41]; NLUI Server — серверное программное обеспечение для построения сценариев взаимодействия с программами на ЕЯ и встраивания поддержки ЕЯ-интерфейса в приложения [42].

Публикации [33—37] описывают результаты исследования, относящегося к смежному научно-техническому направлению. В этих работах излагаются теоретические основы управления файловой системой компьютера с помощью ЕЯ-интерфейсов. Первая версия такой системы получила название NLC-1 (Natural Language Commander — Version 1). Система NLC-1 была разработана для исследований и экспериментов в области ЕЯ-интерфейсов приложений, основанных на действиях.

Формальные средства, предложенные в работе [12] и в данной статье, а также алгоритм, представленный выше в разделе 5, стали основой конструирования новой версии файлового менеджера с ЕЯ-интерфейсом, названного NLC-2. На вход данной программы поступают команды пользователя из некоторого ограниченного подмножества есте-



Этапы обработки предписаний системой NLC-2

ственного языка (русского, английского или другого). Далее программа выполняет три этапа преобразований входного предписания (см. рисунок). Сначала с помощью алгоритма *SemSynRA* программа преобразует введенную пользователем команду в ее первичное КП.

**Пример 1.** Команда "Заархивировать документы в папке "Проект" и переслать на адрес somebody@example.org" будет преобразована в первичное КП, данное в конце раздела 2.

На втором этапе сложные понятия из первичного КП команды, в частности, понятия "документ", "музыка", "видео", заменены более простыми понятиями с дополнительными квалификаторами. Для этого используется информация из терминологической базы данных. Например, "документ" интерпретируется как файл с расширением doc, docx или odt. В результате строится вторичное КП, принадлежащее тому же СК-языку, что и исходное представление.

Такое преобразование возможно благодаря новым выразительным механизмам СК-языков. В частности, можно определить концептуальный базис  $B$  так, что СК-язык  $Ls(B)$  включает выражение

$$\begin{aligned} &(\text{документ}1 \equiv \text{файл}1 * \\ &(\text{Расширение-прогр}, ("doc" \vee "docx" \vee "odt"))). \end{aligned}$$

**Пример 2.** Первичное КП команды "Заархивировать документы в папке "Проект" и переслать на адрес somebody@example.org" будет преобразовано во вторичное КП *Semrepr2* вида

$$\begin{aligned} &(\text{ВыполнениеДействия}(\#\text{сейчас}\#, \text{нек архивирование}1 * (\text{Объект}, \text{нек конечн.множ} * \\ &(\text{Качеств-состав}, \text{файл}1 * (\text{Расширение-прогр}, ("doc" \vee "docx" \vee "odt")) (\text{Локализация}1, \text{нек папка}2 * \\ &(\text{Наименование}1, "Проект"))): S1)(\text{Результат}1, x1), \\ &e1) \wedge \text{ВыполнениеДействия}(\#\text{сейчас}\#, \text{нек электрон-пересылка} * (\text{Объект}1, x1)(\text{Электрон-адрес}, \\ &"somebody@example.org"), e2) \wedge \text{Непосредственно-после}(e2, e1)). \end{aligned}$$

На третьем этапе вторичное КП преобразуется в сценарий на языке командной оболочки операционной системы. NLC-2 использует для этих целей *Born Again Shell (BASH)* как для POSIX-совместимых операционных систем, так и для Windows.

**Пример 3.** Вторичное КП команды "Заархивировать документы в папке "Проект" и переслать на адрес somebody@example.org" будет преобразовано в сценарий

```
zip /tmp/z000129.zip "Проект/*.doc"
"Проект/*.docx" "Проект/*.odt"; sendfile
somebody@example.com /tmp/z000129.zip.
```

На заключительном этапе построенный сценарий выполняется.

## Заключение

Статья предлагает новую парадигму разработки алгоритмов семантического анализа (семантического парсинга) предписаний на естественном языке (русском, английском и других). Новая парадигма имеет несколько особенностей и преимуществ. Во-первых, предложенная математическая модель лингвистической базы данных является значительно более общей, менее детальной по сравнению с двумя моделями лингвистической базы данных, рассматриваемыми в теории К-представлений. В новой модели уменьшено число используемых сущностей, в частности, за счет унификации словарей глагольно-предложных и предложных фреймов. Во-вторых, описание предложенного алгоритма является весьма компактным по сравнению с другими алгоритмами, представленными в научной литературе. Для построения СП предписаний алгоритм использует класс СК-языков (стандартных концептуальных языков), предложенный теорией К-представлений. Был изложен новый подход к формальному описанию семантико-синтаксической структуры как промежуточной структуры при формировании К-представления исходного текста.

Ценной чертой предложенной парадигмы является возможность ее использования не только для английского языка, но и для русского и многих других естественных языков. Поэтому научные результаты, изложенные в данной статье и в работе [12], расширяют теоретические основы мультилингвистического семантического веба. Следует отметить, что эти научные результаты могут найти применение не только в проектировании семантических анализаторов предписаний, но и как теоретический инструмент решения задач, возникающих в других направлениях семантической обработки ЕЯ-текстов.

По сравнению с ранее предложенными алгоритмами *SemSyn* [15], *SemSynt1* [16], *SemSynt2* [33–37] был расширен входной язык: поддержана обработка союза "или"; расширена поддержка об-

работки различных частей речи и взаимосвязей между словами, принадлежащими этим частям речи, а также обработки сочетаний "модальное слово + инфинитив". Алгоритм и модель лингвистической базы данных стали основой разработки файлового менеджера с ЕЯ-интерфейсом NLC-2. С точки зрения трудоемкости для данной тематической области реально наполнить ЛБД содержанием. Алгоритм программно реализован на языке Haskell.

### Список литературы

1. **Lei Y., Uren V., Motta E.** SemSearch: A Search Engine for the Semantic Web / S. Staab, V. Svatek (Eds.) // EKAUW 2006. — Proc. 15th Intern. Conf. on Knowledge Engineering and Knowledge management, 2–6 Oct. 2006, Podybrady, Czech Republic. Lecture Notes in Artificial Intelligence. 2006. Vol. 4248. Berlin, Heidelberg: Springer, P. 238–245.
2. **Fomichov V. A., Kirillov A. V.** A Formal Model for Constructing Semantic Expansions of the Search Requests about the Achievements and Failures // Artificial Intelligence: Methodology, Systems, and Applications / A. Ramsay, G. Agre (Eds.). Lecture Notes in Computer Science. 2012. Vol. 7557. Berlin, Heidelberg: Springer. P. 296–304.
3. **Aggarwal N., Polajnar T., Buitelaar P.** Cross-Lingual Natural Language Querying over the Web of Data / Metais E., Meziane F., Saraee M., Sugumaran V., Vadera S. (Eds.) // Natural Language Processing and Information Systems. 18<sup>th</sup> Intern. Conference on Applications of Natural Language to Information Systems, NLDB 2013, Salford, UK, June 2013, Proceedings. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. 2013. Vol. 7934. P. 152–163.
4. **Buitelaar P., Choi K.-S., Cimiano P., Hovy E. H. (Eds.)** Report from Dagstuhl Seminar 12362 "The Multilingual Semantic Web" (2–9 September, 2012). Schloss Dagstuhl: Leibniz-Zentrum fuer Informatik. 2012. 94 p.
5. **Fomichov V. A.** Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web // Informatica. An Intern. Journal of Computing and Informatics (Slovenia). 2010. Vol. 34, N. 3. P. 387–396.
6. **Dahab M. Y., Hassan H. A., Rafea A.** TextOntoEx: Automatic ontology construction from natural English text // Expert Systems with Applications. 2008. Vol. 34, N. 2. P. 1474–1480.
7. **Wang C., Xiong M., Zhou Q., Yu Y.** PANTO: a portable NL-interface to ontologies // 4<sup>th</sup> European Semantic Web Conference Proceedings. Springer, 2007. P. 473–487.
8. **Cimiano P., Haase P.** et al. Towards Portable Natural Language Interfaces to Knowledge Bases — the Case of the ORAKEL System // Data and Knowledge Engineering. 2008. Vol. 65, N. 2. P. 325–354.
9. **Li W., Yang C.** A semantic search engine for spatial web portals // IGARSS 2008 — 2008 IEEE Symposium on International Geoscience and Remote Sensing, Boston, MA, 7–11 July 2008. Vol. 2. P. 1278–1281.
10. **Erozel G., Cicekli N. K., Cicekli I.** Natural language querying for video databases // Information Sciences. 2008. Vol. 178, N. 12. P. 2534–2552.
11. **Bastianelli E., Castellucci C., Croce D., Basili R.** Textual Inference and Meaning Representation in Human Robot Interaction // Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora, 2013. P. 65–69. URL: <https://aclweb.org/anthology/W/W13/W13-3820v2.pdf>, date of access 9.03.2016.
12. **Разоренов А. А., Фомичев В. А.** Компактная формализация входных и промежуточных данных алгоритмов семантического анализа предписаний // Информационные технологии. 2016. Т. 22, № 12.
13. **Люгер Дж. Ф.** Искусственный интеллект. Стратегии и методы решения сложных проблем. 4-е изд. Пер. с англ. М.: Изд. дом "Вильямс", 2003. 864 с.
14. **Фомичев В. А.** Новый метод преобразования естественно-языковых текстов в семантические представления // Информационные технологии. 2005. № 10. С. 25–35.
15. **Фомичев В. А.** Формализация проектирования лингвистических процессоров. М.: МАКС Пресс. 2005. 368 с.
16. **Fomichov V. A.** Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms. New York, Dordrecht, Heidelberg, London: Springer, 2010. 354 p.
17. **Фомичев В. А.** Представление информации средствами К-исчислений: Учебное пособие // М.: Московский институт электронного машиностроения (МИЭМ), 1988.
18. **Фомичев В. А.** К-языки и разработка новых информационных технологий // Новые информационные технологии в системотехнике / Под ред. Л. С. Болотовой. М.: Радио и Связь, 1990. С. 53–62.
19. **Fomichov V. A.** Mathematical Models of Natural-Language-Processing Systems as Cybernetic Models of a New Kind // Cybernetica. Quarterly Review of the International Association for Cybernetics (Belgium, Namur). 1992. Vol. 35, N. 1. P. 63–91.
20. **Fomichov V. A.** Towards a Mathematical Theory of Natural Language Communication // Informatica. An Intern. Journal of Computing and Informatics (Slovenia). 1993. Vol. 17, N. 1. P. 21–34.
21. **Fomichov V. A.** K-Calculuses and K-Languages as Powerful Formal Means to Design Intelligent Systems Processing Medical Texts // Cybernetica. Quarterly Review of the International Association for Cybernetics (Belgium, Namur). 1993. Vol. 36, N. 2. P. 161–182.
22. **Fomichov V. A.** Integral Formal Semantics and the Design of Legal Full-Text Databases // Cybernetica. Quarterly Review of the International Association for Cybernetics (Belgium, Namur). 1994. Vol. 37, N. 2. P. 145–177.
23. **Fomichov V. A.** A Mathematical Model for Describing Structured Items of Conceptual Level // Informatica. An Intern. Journal of Computing and Informatics (Slovenia). 1996. Vol. 20, N. 1. P. 5–32.
24. **Fomichov V. A.** An Ontological Mathematical Framework for Electronic Commerce and Semantically-structured Web // Zhang Y., Fomichov V. A., Zeleznikar A. P. (Eds.), Special Issue on Database, Web, and Cooperative Systems. Informatica. An Intern. Journal of Computing and Informatics (Slovenia). 2000. Vol. 24, N. 1. P. 39–49.
25. **Фомичев В. А.** Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть I // Информационные технологии. 2002. № 10. С. 16–25.
26. **Фомичев В. А.** Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть II // Информационные технологии. 2002. № 11. С. 34–45.
27. **Фомичев В. А.** Стандартные К-языки как универсальный и гибкий инструмент формирования контрактов и протоколов переговоров в области электронной коммерции // Информационные технологии. 2005. № 3. С. 26–29.
28. **Фомичев В. А.** Класс формальных языков и алгоритм для построения семантических аннотаций Веб-документов // Вестник МГТУ им. Н. Э. Баумана. Сер. "Приборостроение". 2005. № 3 (60). С. 73–86.
29. **Fomichov V. A.** Standard K-languages as a Powerful and Flexible Tool for Building Contracts and Representing Contents of Arbitrary E-Negotiations // Bauknecht K., Proell B. (Eds.) E-Commerce and WebTechnologies. 6<sup>th</sup> International Conference, EC-Web 2005, Copenhagen, Denmark, August 2005, Proceedings. Lecture Notes in Computer Science. Springer, Heidelberg. 2005. Vol. 3590. P. 138–147.
30. **Фомичев В. А.** Математические основы представления содержания посланий компьютерных интеллектуальных агентов. М.: ГУ-ВШЭ, издательство "ТЕИС". 2007. 176 с.
31. **Fomichov V. A.** A Broadly Applicable and Flexible Conceptual Metagrammar as a Basic Tool for Developing a Multilingual Semantic Web / Metais E., Meziane F., Saraee M., Sugumaran V., Vadera S. (Eds.) // Natural Language Processing and Information Systems. 18<sup>th</sup> Intern. Conf. on Applications of Natural Language to Information Systems, NLDB 2013. Salford, UK, June 2013, Proceedings. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013. Vol. 7934. P. 249–259.
32. **Захлебин И. В., Фомичев В. А.** Разработка метода семантического поиска специалистов в корпоративной базе данных по естественно-языковым запросам // Информационные технологии. 2015. № 5. 323–331.
33. **Фомичев В. А., Разоренов А. А.** Значение теории К-представлений для исследований по автоматическому выявлению

семантических ролей // Информационные технологии. 2015. № 6. С. 403—411.

34. **Fomichov V. A., Razorenov A. A.** A New Method of Extracting Structured Meanings from Natural Language Texts and Its Application // E. Metais, M. Roche, M. Teisseire (Eds.). *Natural Language Processing and Information Systems. 19<sup>th</sup> International Conference on Applications of Natural Language to Information Systems, NLDB 2014*, Montpellier, France, June 18—20, 2014, Proceedings. Cham: Springer International Publishing Switzerland. 2014. P. 81—84.

35. **Razorenov A. A., Fomichov V. A.** The Design of a Natural Language Interface for File System Operations on the Basis of a Structured Meanings Model // *Procedia Computer Science*, Elsevier. 2014. Vol. 31. P. 1005—1011; open access, URL: <http://authors.elsevier.com/sd/article/S1877050914005304>.

36. **Разоренов А. А.** Построение естественно-языкового интерфейса управления файловой системой на основе теории К-представлений // Информационные средства и технологии. Труды XXII Международной научно-технической конференции. В 3 т. Т. 3. М.: Издательский дом МЭИ, 2014. С. 59—65.

37. **Разоренов А. А.** Разработка метода управления программами на естественном языке / Под общ. ред.: А. Н. Тихонов, В. Н. Азаров, У. В. Аристова, М. В. Карасев, В. П. Кулагин, Ю. Л. Леохин, Б. Г. Львов, Н. С. Титкова, С. У. Увайсов // Научно-техническая конференция студентов, аспирантов и молодых специалистов НИУ ВШЭ им. Е. В. Арменского. Материалы конференции. М.: МИЭМ НИУ ВШЭ, 2015. С. 59—60.

38. **Web-resource:** Ninety-Nine Haskell Problems: Construct all spanning trees. URL: [https://wiki.haskell.org/99\\_questions/Solutions/83](https://wiki.haskell.org/99_questions/Solutions/83), date of acces21.05.2015.

39. **Siri (Электронный ресурс)**, URL: <http://www.apple.com/ios/siri/> (проверено 10.02.2014).

40. **Sobesednik HD (Электронный ресурс)**, URL: <http://sobesednik.idevelops.com/> (проверено 10.02.2014).

41. **Braina Project (Электронный ресурс)**, URL: <http://www.brainasoft.com/braina/#overview> (проверено 10.02.2014).

42. **NLUI Server (Электронный ресурс)**, URL: [http://www.linguasys.net/web\\_production/server-item/NLUI%20Server](http://www.linguasys.net/web_production/server-item/NLUI%20Server) (проверено 11.02.2014).

**A. A. Razorenov**, Ph. D., Postgraduate, e-mail: [arazorenov@hse.ru](mailto:arazorenov@hse.ru),

**V. A. Fomichov**, Professor, e-mail: [vfomichov@hse.ru](mailto:vfomichov@hse.ru),

National Research University Higher School of Economics (HSE), Moscow, Russia

## A New Approach to Formalization of Instructions' Semantic Processing Based on the Theory of K-representations

*The paper proposes a new approach to the development of algorithms for semantic parsing of natural language instructions, using the formal means for representing the input, intermediate, and output data. The principal theoretical result is the development of an original, broadly applicable algorithm of instructions' semantic parsing. For building a semantic representations (SR) of an instruction, the algorithm uses the theory of K-representations (knowledge representations) — the only theory today being convenient for constructing SR of arbitrarily complex instructions. The algorithm also uses a compact formal model of a linguistic database (LDB) and the notion of a graph of semantic-syntactic ties (GSST). The model of LDB and the notion of GSST are modified in comparison with their descriptions in the previous article of the authors in the journal "Informational Technologies" (Moscow, 2016, no. 12). In comparison with the known algorithms, the new algorithm can handle much more complex instructions formed by several verbs with dependent words. On the other hand, the new algorithm is very compact due to the use of the results of graph theory. The proposed algorithm has been used in the design of practically useful file manager NLC-2 with a natural language interface (Natural Language Commander — Version 2) which was developed by means of Haskell programming language.*

**Keywords:** natural language processing, semantic parsing of instructions, semantic representation, theory of K-representations, SK-language, dictionary of lexical frames, linguistic database, graph of semantic-syntactic ties, natural language control of file system, Natural Language Commander

### References

1. **Lei Y., Uren V., Motta E.** SemSearch: A Search Engine for the Semantic Web. S. Staab and V. Svatek (Eds.). *EKAW 2006 — Proc. 15th Intern. Conf. on Knowledge Engineering and Knowledge management, 2—6 Oct. 2006, Podebrady, Czech Republic. Lecture Notes in Artificial Intelligence*. Berlin, Heidelberg: Springer, 2006, vol. 4248, pp. 238—245.

2. **Fomichov V. A., Kirillov A. V.** A Formal Model for Constructing Semantic Expansions of the Search Requests about the Achievements and Failures. *Ramsay A., G. Agre G. (Eds.). Artificial Intelligence: Methodology, Systems, and Applications. Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer, 2012, vol. 7557, pp. 296—304.

3. **Aggarwal N., Polajnar T., Buitelaar P.** Cross-Lingual Natural Language Querying over the Web of Data. Metais E., Meziane F., Sarace M., Sugumaran V., Vadera S. (Eds.), *Natural Language Processing and Information Systems. 18<sup>th</sup> Intern. Conference on Applications of Natural Language to Information Systems, NLDB 2013, Salford, UK, June 2013, Proceedings. Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer, 2013, vol. 7934, pp. 152—163.

4. **Buitelaar P., Choi K.-S., Cimiano P., Hovy E. H. (Eds.)** *Report from Dagstuhl Seminar 12362 "The Multilingual Semantic Web" (2—9 September, 2012)*. Schloss Dagstuhl: Leibniz-Zentrum fuer Informatik, 2012. 94 p.

5. **Fomichov V. A.** Theory of K-representations as a Comprehensive Formal Framework for Developing a Multilingual Semantic Web, *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia), 2010, vol. 34, no. 3, pp. 387—396.

6. **Dahab M. Y., Hassan H. A., Rafea A.** TextOntoEx: Automatic ontology construction from natural English text, *Expert Systems with Applications*, 2008, vol. 34, no. 2, pp. 1474—1480.

7. **Wang C., Xiong M., Zhou Q., Yu Y.** PANTO: a portable NL-interface to ontologies, *4<sup>th</sup> European Semantic Web Conference Proceedings*. Springer, 2007, pp. 473—487.

8. **Cimiano P., Haase P. et al.** Towards Portable Natural Language Interfaces to Knowledge Bases — the Case of the ORAKEL System, *Data and Knowledge Engineering*, 2008, vol. 65, no. 2, pp. 325—354.

9. **Li W., Yang C.** A semantic search engine for spatial web portals, *IGARSS 2008 — 2008 IEEE Symposium on International Geo-*

science and Remote Sensing, Boston, MA, 7–11 July 2008, vol. 2, pp. 1278–1281.

10. **Erozal G., Cicekli N. K., Cicekli I.** Natural language querying for video databases, *Information Sciences*, 2008, vol. 178, no. 12, pp. 2534–2552.

11. **Bastianelli E., Castellucci C., Croce D., Basili R.** Textual Inference and Meaning Representation in Human Robot Interaction, *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*, 2013, pp. 65–69. URL: <https://aclweb.org/anthology/W/W13/W13-3820v2.pdf>, retrieved 9.03.2016.

12. **Razorenov A. A., Fomichov V. A.** Kompaktnaya formalizatsiya vhodnykh i promezhutochnykh dannykh algoritmov semanticheskogo analiza predpisaniy, *Informatsionnye Tekhnologii*, 2016, vol. 22, no. 12 (In Russian)

13. **Luger G. F.** *Iskusstvennyi intellekt. Strategii i metody resheniya slozhnykh problem*. 4<sup>th</sup> edition. Translated from English. Moscow, Publishing House "Williams", 2003, 864 p. (In Russian)

14. **Fomichov V. A.** Novyi metod preobrazovaniya estestvenno-yazykovykh tekstov v semanticheskie predstavleniya, *Informatsionnye Tekhnologii*, 2005, no. 10, pp. 25–35 (In Russian)

15. **Fomichov V. A.** *Formalizatsiya proektirovaniya lingvisticheskikh processorov*. Moscow: MAKS Press, 2005. 368 p. (In Russian)

16. **Fomichov V. A.** *Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms*. New York, Dordrecht, Heidelberg, London: Springer, 2010. 352 p.

17. **Fomichov V. A.** *Representing Information by Means of K-Calculuses: Textbook*. Moscow, Moscow Institute of Electronic Engineering, 1988 (In Russian).

18. **Fomichov V. A.** K-yazyki i razrabotka novykh informatsionnykh tekhnologii. Bolotova L. S. (Eds.) *Novye informatsionnye tekhnologii v sistemotekhnike*, Moscow, Radio i Svyaz, 1990, pp. 53–62 (In Russian).

19. **Fomichov V. A.** Mathematical Models of Natural-Language-Processing Systems as Cybernetic Models of a New Kind, *Cybernetica. Quarterly Review of the International Association for Cybernetics* (Belgium, Namur), 1992, vol. 35, no. 1, pp. 63–91.

20. **Fomichov V. A.** Towards a Mathematical Theory of Natural Language Communication, *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia), 1993, vol. 17, no. 1, pp. 21–34.

21. **Fomichov V. A.** K-Calculuses and K-Languages as Powerful Formal Means to Design Intelligent Systems Processing Medical Texts, *Cybernetica. Quarterly Review of the International Association for Cybernetics* (Belgium, Namur), 1993, vol. 36, no. 2, pp. 161–182.

22. **Fomichov V. A.** Integral Formal Semantics and the Design of Legal Full-Text Databases. *Cybernetica. Quarterly Review of the International Association for Cybernetics* (Belgium, Namur), 1994, vol. 37, no. 2, pp. 145–177.

23. **Fomichov V. A.** A Mathematical Model for Describing Structured Items of Conceptual Level, *Informatica. An Intern. Journal of Computing and Informatics* (Slovenia), 1996, vol. 20, no. 1, pp. 5–32.

24. **Fomichov V. A.** An Ontological Mathematical Framework for Electronic Commerce and Semantically-structured Web. Y. Zhang, V. A. Fomichov, A. P. Zeleznikar (Eds.), *Special Issue on Database, Web, and Cooperative Systems. Informatica. An Intern. Journal of Computing and Informatics* (Slovenia), 2000, vol. 24, no. 1, pp. 39–49.

25. **Fomichov V. A.** Matematicheskie osnovy predstavleniya smysla tekstov dlya razrabotki lingvisticheskikh informatsionnykh tekhnologii. Part I, *Informatsionnye Tekhnologii*, 2002, no. 10, pp. 16–25 (in Russian).

26. **Fomichov V. A.** Matematicheskie osnovy predstavleniya smysla tekstov dlya razrabotki lingvisticheskikh informatsionnykh tekhnologii. Part II, *Informatsionnye Tekhnologii*, 2002, no. 11, pp. 34–45 (in Russian).

27. **Fomichov V. A.** Standartnye K-yazyki kak universalnyi i gibkii instrument formirovaniya kontraktov i protokolov peregovorov v oblasti elektronnoy kommertsii, *Informatsionnye Tekhnologii*, 2005, no. 3, pp. 26–29 (In Russian).

28. **Fomichov V. A.** Klass formalnykh yazykov i algoritm dlya postroeniya semanticheskikh annotatsii Web-dokumentov, *Vestnik MG-TU imeni N. E. Baumana. Ser. "Priborostroenie"*, 2005, no. 3 (60), pp. 73–86 (In Russian).

29. **Fomichov V. A.** Standard K-languages as a Powerful and Flexible Tool for Building Contracts and Representing Contents of Arbitrary E-Negotiations. K. Bauknecht, B. Proell (Eds.), *E-Commerce and Web Technologies. 6<sup>th</sup> International Conference, EC-Web 2005, Copenhagen, Denmark, August 2005, Proceedings. Lecture Notes in Computer Science*. Springer, Heidelberg, 2005, vol. 3590, pp. 138–147.

30. **Fomichov V. A.** *Matematicheskie osnovy predstavleniya sod-erzhaniya poslanii kompyuternykh intellektualnykh agentov*. Moscow: State University — Higher School of Economics, Publishing House "TEIS", 2007, 176 p. (In Russian).

31. **Fomichov V. A.** A Broadly Applicable and Flexible Conceptual Metagrammar as a Basic Tool for Developing a Multilingual Semantic Web. Metais E., Meziane F., Saraee M., Sugumaran V., Vadera S. (Eds.), *Natural Language Processing and Information Systems. 18th Intern. Conf. on Applications of Natural Language to Information Systems, NLDB 2013. Salford, UK, June 2013, Proceedings. Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer, 2013, vol. 7934, pp. 249–259.

32. **Zakhlebin I. V., Fomichov V. A.** Razrabotka metoda semanticheskogo poiska spetsialistov v korporativnoy baze dannykh po estestvenno-yazykovym zaprosam, *Informatsionnye Tekhnologii*, 2015, no. 5, pp. 323–331 (in Russian).

33. **Fomichov V. A., Razorenov A. A.** Znachenie teorii K-predstavlenii dlya issledovaniy po avtomaticheskomu vyavleniyu semanticheskikh roley, *Informatsionnye Tekhnologii*, 2015, no. 6, pp. 403–411 (in Russian).

34. **Fomichov V. A., Razorenov A. A.** A New Method of Extracting Structured Meanings from Natural Language Texts and Its Application. E. Metais, M. Roche, M. Teisseire (Eds.), *Natural Language Processing and Information Systems. 19<sup>th</sup> International Conference on Applications of Natural Language to Information Systems, NLDB 2014, Montpellier, France, June 18–20, 2014, Proceedings*. Cham: Springer International Publishing Switzerland, 2014, pp. 81–84.

35. **Razorenov A. A., Fomichov V. A.** The Design of a Natural Language Interface for File System Operations on the Basis of a Structured Meanings Model. *Procedia Computer Science*. Elsevier, 2014, vol. 31, pp. 1005–1011; open access, URL: <http://authors.elsevier.com/sd/article/S1877050914005304>.

36. **Razorenov A. A.** Postroenie estestvenno-yazykovogo interfeisa upravleniya failovoy sistemoy na osnove teorii K-predstavlenii, *Informatsionnye sredstva i tekhnologii. Trudy XXII Mezhdunarodnoy nauchno-tekhnicheskoy konferentsii. V 3 t. Vol. 3*. Moscow, Izdatel'skii dom MEI, 2014, pp. 59–65 (In Russian).

37. **Razorenov A. A.** Razrabotka metoda upravleniya programmami na estestvennom yazyke, Pod obschey red.: A. N. Tikhonov, V. N. Azarov, U. V. Aristova, M. V. Karasev, V. P. Kulagin, Y. L. Leokhin, B. G. L'vov, N. S. Titkova, S. U. Uvaysov, *Nauchno-tekhnicheskaya konferentsiya studentov, aspirantov i molodykh spetsialistov HSE imeni E. V. Armenskogo. Materialy konferentsii*. Moscow, MIEM NRU HSE, 2015, pp. 59–60 (In Russian).

38. **Web-resource:** Ninety-Nine Haskell Problems: Construct all spanning trees. URL: [https://wiki.haskell.org/99\\_questions/Solutions/83](https://wiki.haskell.org/99_questions/Solutions/83) — Electronic resource (retrieved 21.05.2015).

39. **Web-resource:** Siri. URL: <http://www.apple.com/ios/siri/> — Electronic resource (retrieved 10.02.2014).

40. **Web-resource:** Sobesednik HD, URL: <http://sobesednik.idevelops.com/> — Electronic resource (retrieved 10.02.2014).

41. **Web-resource:** Braina Project, URL: <http://www.brainasoft.com/braina/#overview> — Electronic resource (retrieved 10.02.2014).

42. **Web-resource:** NLUI Server, URL: [http://www.linguasys.net/web\\_production/server-item/NLUI%20Server](http://www.linguasys.net/web_production/server-item/NLUI%20Server) — Electronic resource (retrieved 11.02.2014).

**В. В. Наумова**<sup>1</sup>, д-р геомин. наук, зав. лаб., e-mail: naumova@fegi.ru,

**С. Е. Дьяков**<sup>2</sup>, науч. сотр., e-mail: sergdkv@gmail.com,

**К. А. Платонов**<sup>1</sup>, мл. науч. сотр., e-mail: platonov@fegi.ru

<sup>1</sup>ФГБУН Дальневосточный геологический институт ДВО РАН, г. Владивосток

<sup>2</sup>ФГБУН Институт автоматизации и процессов управления ДВО РАН, г. Владивосток

## Технологии поиска и представления информации в научной инфраструктуре "Геология Дальнего Востока России"

*В основе подхода для реализации инфраструктуры "Геология Дальнего Востока России" лежит слабосвязанная блочная инфраструктура, основанная на различиях в типах геологических данных: пространственных, количественных, библиографических и основанных на экспертных знаниях. В каждом отдельном информационном блоке инфраструктуры для интеграции, хранения и поиска данных применяются различные подходы и технологические решения. Концепция организации единой точки доступа к данной инфраструктуре заключается в том, чтобы предоставить пользователю всю возможную информацию о геологических объектах Дальнего Востока России и сервисы ее обработки одновременно, в одном месте, в единой точке доступа. В статье описаны технологии поиска и представления информации для данной инфраструктуры.*

**Ключевые слова:** информационная инфраструктура, интеграция неоднородных источников информации, геологическая информация, геология Дальнего Востока России

### Введение

Наличие развитой информационной инфраструктуры является необходимым условием обеспечения эффективности научных исследований. Интеграция информационных и вычислительных ресурсов в единую среду и организация доступа к ним является одним из важнейших направлений развития современных информационных технологий. Стремительное развитие глобальных компьютерных сетей ведет к изменению фундаментальных парадигм обработки данных вследствие необходимости поддержки и развития распределенных информационно-вычислительных ресурсов [1].

Необходимость создания целостного интегрального информационного поля пользователя, которое состоит из совокупности инструментальных средств, аналитических методов, описаний и геоданных, необходимых для проведения прикладных и фундаментальных исследований в науках о Земле, отмечалась в статье Н. П. Лаврова с соавторами [2]: "Подход к созданию целостного интегрального информационного поля пользователя в области наук о Земле реализует архитектуру системы взаимосвязанных порталов, каждый из которых отвечает либо за определенную широкую предметную область, либо за отдельные аспекты технологического функционирования всей системы. Система порталов связана гиперссылками и представляет собой единую распределенную структуру, взаимодействие в которой основано на эффективном обмене метаинформацией и ресурсами".

В последние годы созданы и развиваются крупные зарубежные информационные системы, доступ к которым возможен через Интернет: PANGAEA, GeoRock, GeoRef и др.

Среди российских разработок необходимо отметить систему "Геомета" [3], информационно-вычислительную систему сбора, хранения и обработки данных дистанционного зондирования Земли для регионов Сибири и Дальнего Востока [4], информационно-аналитическую систему поддержки и сопровождения научных исследований "Природные ресурсы Карелии" [5–6] и др.

В. В. Наумовой с соавторами в 2011 г. предложено решение для интеграции неоднородных геологических порталов между собой, основанное на технологиях web-интеграции [7].

Задача web-интеграции — объединить разнородные web-приложения и системы в единую среду на базе web.

Типы интеграции:

- **Интеграция на уровне представления.** Уровень представления — web-базируемый пользовательский интерфейс, платформо-зависимый графический пользовательский интерфейс (GUI) или консоль терминала. Данный уровень позволяет пользователю взаимодействовать с приложением. Интеграция на уровне представления дает доступ к пользовательскому интерфейсу удаленных приложений.
- **Интеграция на уровне функциональности.** Данная интеграция подразумевает обеспечение прямого доступа к логике приложений. Это достигается непосредственным взаимодействием приложений с API (программному интерфейсу приложений) или же взаимодействием посредством web-сервисов.
- **Интеграция на уровне данных.** В данном случае предполагается доступ к одной или нескольким базам данных, используемых удаленным приложением.

- *Комплексная интеграция.* Современные решения по web-интеграции, как правило, включают все три типа интеграции.

### **Информационная инфраструктура поддержки и сопровождения научных геологических исследований на Дальнем Востоке России**

Создаваемая в Дальневосточном геологическом институте ДВО РАН (ДВГИ ДВО РАН) информационная инфраструктура для поддержки и сопровождения научных геологических исследований на Дальнем Востоке России представляет собой блочную платформу для интеграции неоднородных источников геологической информации и сервисов их обработки. Инфраструктура имеет единую точку входа WWW-портал <http://fareastgeology.ru> (рис. 1, четвертая сторона обложки), которая предоставляет пользователям следующие возможности:

- доступ к разнотипным научным геологическим ресурсам Дальнего Востока России через Интернет: научным публикациям, картам, спутниковым данным, количественным данным, фото- и видеоинформации; информации о научных конференциях и др.;
- легкость нахождения специализированных данных с использованием тематических запросов, а также сервисов их обработки;
- визуализацию результатов поиска, в том числе и на картографической основе;
- удобство в распространении геологических данных и сервисов на уровне данных (метаданных), соответствующих международным стандартам и протоколам.

Проектирование и реализация инфраструктуры проведены на основе анализа мирового и отечественного опыта создания и развития территориально распределенных систем и электронных каталогов.

Территориальные границы инфраструктуры: территория Дальневосточного федерального округа РФ: 42° с.ш., 105° в.д. — 77° с.ш., 170° з.д.

Источники информации — территориально распределенные Интернет-ресурсы, информация в которых основана на стандартизованных метаданных и программные решения которых допускают применение стандартизованных протоколов для ее автоматической интеграции в создаваемую инфраструктуру, а также научные материалы научных организаций, библиотек, центров данных и личные материалы сотрудников ДВГИ ДВО РАН.

Основные типы информации:

- научные публикации: статьи, монографии, рукописи, авторефераты диссертаций и диссертации, учебные пособия и другие материалы;
- геологическая картографическая информация (источник государственной геологической информации — ВСЕГЕИ; топографические карты (источник информации — Государственная геодезическая сеть); материалы международных

проектов, опубликованные в открытых отчетах, таких как USGS Open-File Report;

- электронная информация Росгеолфонда: Государственные геологические отчеты и Кадастр месторождений полезных ископаемых;
- спутниковая информация из открытых источников — центров спутникового мониторинга, предоставляющих доступную и достоверную информацию (спутниковая топография; данные ИК каналов радиометров TIRS, AVHRR, MODIS; данные измерений видимых каналов спутников Aqua, Terra, Suomi NPP, спутников серии Landsat);
- экспертные знания по геологии Дальнего Востока (wikipedia);
- научная количественная информация;
- информационные материалы о геологах-дальневосточниках.

Основные методические и технологические решения, использованные при проектировании и реализации инфраструктуры, описаны Наумовой В. В. с соавторами [8].

Инфраструктура строится на основе нескольких слабосвязанных подсистем, обеспечивающих интеграцию территориально распределенных данных и информационных систем: ГИС, картографической информации, спутниковых данных, научных публикаций, количественных данных и экспертных знаний (wikipedia) из различных источников, а также доступа к этим данным на основании тематических запросов.

### **Основные блоки инфраструктуры**

#### *Сервис доступа к картографической информации*

Модуль доступа к картографической информации предназначен для предоставления государственных геологических и топографических карт масштаба 1М, 200М по всей территории Дальнего Востока. Модуль снабжен системой поиска картографической информации, системой просмотра списков карт. Источниками данных для данного модуля служат GIS-сервисы ВСЕГЕИ. Модуль реализован в программной среде Geonetwork 3.0.4.0.

#### *Сервис доступа к электронной информации Росгеолфонда*

Авторский модуль доступа к государственным геологическим отчетам и каталогам месторождений Росгеолфонда снабжен системой контекстного и географического поиска, использующей однотабличную базу данных. Данные регулярно собираются с сервера Росгеолфонда и заносятся в базу данных первичной информации. После занесения массив данных может дополняться. Далее строится полнотекстовый индекс для обеспечения возможности полнотекстового поиска, создаются индексы административных единиц. Выдача информации различна для геологических отчетов и месторождений, что связано с различиями в организации данных в базе данных Росгеолфонда.



### Сервис доступа к спутниковым данным

Модуль доступа к спутниковым данным строится на основе интеграции со спутниковыми сервисами Центра спутникового мониторинга Института автоматизации и процессов управления ДВО РАН (ИАПУ ДВО РАН), Геопортала Тихоокеанского института географии ДВО РАН (ТИГ ДВО РАН), сервиса доступа к спутниковым данным Геологической службы США (USGS). Поставляемые спутниковые данные проходят географическую привязку и атмосферную коррекцию (ПК SeaDAS и др.), что облегчает использование данных.

В настоящее время модуль использует разные методы для доступа к данным различных спутников. Так, для поиска данных радиометра MODIS используется сервис OBPG центра космических полетов Годдарда (NASA, США). Данные извлекаются из хранилища Регионального центра спутникового мониторинга ДВО РАН, обработка данных выполняется ПК SeaDAS 7.1 и набором утилит, управляемых специально созданными сценариями обработки. Предоставление данных пользователю осуществляется через специальную систему заказов. Обработка данных спутников LANDSAT (LANDSAT-7, LANDSAT-8) организована на вычислительной площадке ИАПУ ДВО РАН, на основе программного обеспечения Gdal и набора утилит, управляемых сценариями обработки. Источником данных является ГИС-портал ТИГ ДВО РАН. В отличие от других служб доступа к спутниковым данным, поставляемые спутниковые данные проходят географическую привязку и атмосферную коррекцию (ПК SeaDAS и др.). Поставка осуществляется в нескольких форматах одновременно, что обеспечивает подготовку иллюстраций, визуальный анализ и использование в ГИС-системах [9].

### Электронная библиотека

В данном модуле осуществляется интеграция научных публикаций по геологии Дальнего Востока России из разнородных территориально распределенных источников. Интеграция данных осуществляется с использованием наиболее распространенных протоколов и интерфейсов доступа, а также стандартов метаданных научных публикаций [10]. Предложены подходы и технологические решения на основе совместного применения УДК и тематических тезаурусов для решения задачи географической фильтрации научных публикаций. Модуль реализован в программной среде Dspace с добавлением авторских программ.

### Центр количественных данных

В рамках работ по созданию информационной инфраструктуры для поддержки и сопровождения научных геологических исследований на Дальнем Востоке России разрабатывается Центр количественных данных. Основные источники данных для этого блока инфраструктуры: таблицы количественных данных, извлекаемые из научных публика-

ций, хранящихся в электронной библиотеке (ЭБ), входящей в состав инфраструктуры; авторские базы геологических данных; мировые центры геологических данных.

Предложен общий подход обработки информации, хранящейся в научных публикациях по геологии, а также метод и алгоритм автоматического извлечения количественной информации из таблиц, входящих в научные публикации [11]. В автоматическом режиме из ЭБ по протоколу OAI-PMH собираются карточки метаданных, а по протоколу HTTP загружается полный текст статьи в формате PDF.

Рабочий прототип системы доступен через Интернет. Он разработан в виде модуля открытой программной платформы CMF Drupal, реализованной на скриптовом языке общего назначения PHP. Архитектура Drupal позволяет применять его для построения информационных систем любой сложности. Имеющуюся по умолчанию функциональность можно увеличивать подключением дополнительных расширений.

В состав инфраструктуры входят также информационные ресурсы:

- *wikipedia*;
- *медиа-сервер и фотоархив*;
- *сайт "Геологи-дальневосточники"*.

Каждый блок инфраструктуры имеет собственные программные среды сбора, хранения, поиска и предоставления информации и может работать в автономном режиме.

Инфраструктура построена на основе ряда слабовязанных подсистем, обеспечивающих интеграцию территориально распределенных разнотипных данных и информационных систем: ГИС, картографической информации, спутниковых данных, научных публикаций, количественных данных и экспертных знаний из различных источников, а также доступа к этим данным на основании поисковых запросов (рис. 2).



Рис. 2. Обобщенная схема информационной инфраструктуры

## Единая точка доступа к инфраструктуре

В основе подхода для реализации инфраструктуры лежит слабосвязанная блочная инфраструктура, основанная на различии в типах геологических данных: пространственных, количественных, библиографических и основанных на экспертных знаниях. В каждом отдельном информационном блоке инфраструктуры для интеграции, хранения и поиска данных применяются различные подходы и технологические решения.

Концепция организации единой точки доступа к данной инфраструктуре заключается в том, чтобы предоставить пользователю всю возможную информацию о геологических объектах Дальнего Востока России и сервисы ее обработки одновременно, в одном месте, в единой точке доступа.

Таким образом, возникает необходимость разработки подхода для организации сервиса поиска по всем технологически независимым ресурсам инфраструктуры. В настоящее время существует ряд способов организации подобных систем поиска, в том числе:

- 1) использование систем поиска общего назначения (google, yandex, bing и т. п.);
- 2) непосредственный поиск в базе данных отдельных программных блоков;
- 3) модернизация исходного кода независимых программных блоков: встраивание в них плагинов для поиска;
- 4) использование собственных и внешних систем поиска с последующей фильтрацией результатов.

Подход 1 является максимально простым в использовании, но не является общим: существуют ресурсы, для которых полная индексация невозможна в принципе; подходы 2 и 3 работоспособны, но снижают степень ортогональности программной системы и могут использоваться лишь с ограничениями; подход 4 является достаточно общим, ортогональным и устойчивым к изменениям (робастным).

Система поиска в инфраструктуре основана на подходе 4 и включает в себя следующие модули:

- модуль интерфейса пользователя;
- преобразователь запросов (преобразование регионов, заданных координатами, в набор географических названий);
- модуль-супервизор, отвечающий за взаимодействие с сайтами, выполняющий параллельные запросы к сайтам с помощью набора микромодулей;
- набор микромодулей, отвечающих за взаимодействие с конкретными сайтами;
- модуль дополнительной фильтрации полученных результатов.

Проектом предусматривается поиск информации по следующим категориям: что, где, кто и когда.

Поисковые запросы строятся последовательно. На первом этапе организуется поиск информации в категориях *где* и *когда*:

- географический поиск;
- поиск по времени (стратиграфическому возрасту; абсолютному возрасту; дате документа).

Сервис географического поиска (рис. 3, см. четвертую сторону обложки) организован в виде внешнего модуля для программных блоков инфраструктуры, реализованным на различных технологических платформах, и он фактически преобразует поисковый запрос пользователя в поисковые запросы к различным программным системам. Модуль выполняет поиск автоматически, в последующем преобразуя результаты поиска в единый формат, и выдает результаты пользователю. Данный метод позволяет использовать все возможности поиска источников данных, но ограничен их технологическими возможностями.

Схема модуля поиска представлена на рис. 4.

Система поиска реализует интерфейс запросов, и взаимодействует с другими модулями с использованием технологий AJAX и формата представления данных JSON.

Транслятор запросов преобразует тематический запрос в набор запросов для каждого модуля, определяет число необходимых запросов.

Исполнитель запросов непосредственно выполняет поисковый запрос к внешнему источнику данных. Результат запроса преобразуется в формат поискового модуля и предоставляется пользователю. Несколько исполнителей запросов выполняются одновременно.

В том случае если внешний источник данных не поддерживает запросы на поиск, осуществляется обращение к промежуточному исполнителю запросов. Промежуточный исполнитель может разрешить это затруднение с помощью обращения к поисковой системе Google, Yandex и т. п.

Данная схема достаточно гибкая и расширяемая, но не является универсальной. Добавление

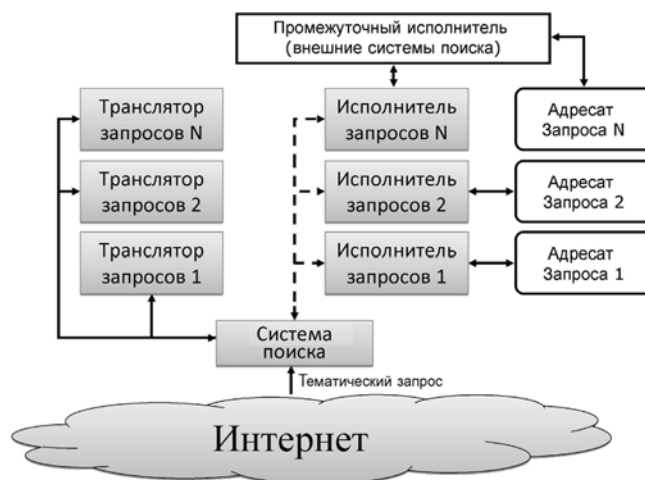


Рис. 4. Обобщенная схема модуля поиска

каждого источника поиска и типа вывода требует модификации подмодулей или увеличения их числа.

### *Географический поиск*

Для построения географического поиска нами использованы подходы и технологии, обеспечивающие обработку географического аспекта информации.

В статье [12] обсуждаются вопросы, связанные с построением интероперабельного тезауруса географических наименований, включающего геометрические данные географических объектов, в том числе и ретроспективные. Определяются основные требования к подобному тезаурусу, выполняется обзор существующих решений исходя из описанных требований, формулируются основные позиции соответствующего профиля для организации доступа к тезаурусу, приводится реляционная схема, предназначенная для хранения данных тезауруса.

Геологическая изученность любой территории является наиболее важным показателем, определяющим представление о ее геологическом строении и минерально-сырьевом потенциале. На основе визуального географического запроса пользователя (в абсолютных координатах) инфраструктура предоставляет пользователю информацию из всех блоков: государственные геологические карты всех масштабов на выбранную территорию, геологические отчеты, информацию из Государственного кадастра месторождений, научные публикации, количественные данные, медиа-информацию, а также данные энциклопедии. Предложены правила, определяющие пространственную вложенность ответов инфраструктуры в выбранную территорию. Поскольку текущий модуль доступа к научным публикациям не позволяет выполнять поиск по географическим координатам, а использует только географические названия, то необходимо реализовать прямое и обратное преобразования от координат к названиям географических объектов.

Преобразование выбранных пользователями регионов, заданных абсолютными координатами, в набор географических названий осуществляется на основе открытых сервисов *OpenStreetMap* и Службы геокодирования API сервиса *Яндекс.Карты*, которые позволяют определить координаты объекта, а также найти адрес, наиболее близкий к указанным координатам. Координаты объектов указаны в виде точек. Переход от выбранного региона поиска к набору координат осуществляется с помощью регулярной сетки, шаг которой адаптивно выбирается на основе масштаба.

### *Поиск по времени*

Осуществляется по трем параметрам:

- стратиграфическому возрасту;
- абсолютному возрасту;
- дате документа.

### *Уточнение результатов поиска*

Проектом предусматривается реализация уточнения результатов поиска. Получив при первичном поиске большое количество результатов, мы должны его уточнить. Уточнение результатов поиска в настоящее время осуществляется, как правило, методами фасетного поиска. Найденные результаты сразу распределяются на группы (фасеты) по заранее заданным признакам. Нередко их число в каждом фасете подсчитывается, и пользователь сразу понимает, в каком направлении следует сужать поиск. Разнообразие здесь состоит в наборе фасетов. Уточнение поиска происходит по категориям: *автор* и *что*.

Окончательные результаты поиска распределяются по четырем категориям: *что, где, кто* и *когда*, в которые соответственно названиям выносятся предметные рубрики, географический охват, автор и временные рамки.

Эффективность поиска информации характеризуется тремя основными показателями: полнота, точность, затраты времени на поиск [5]. Полнота поиска означает, что найдены все релевантные запросу данные в заданном массиве. Однако сплошной просмотр всех данных в массиве приводит к существенному увеличению времени поиска. Для уменьшения последнего целесообразно систематизировать массив данных так, чтобы поиск выполнялся только в определенной части массива. Точность поиска (пертинентность) означает, что в отклике на запрос присутствуют именно те данные, которые соответствуют информационным потребностям пользователя. Одним из главных средств "борьбы" за точность поиска также является систематизация данных. Кроме того, повышению полноты и точности поиска способствует технология построения запросов, основанная на соответствующей систематизации предметных областей.

Информационная инфраструктура поддержки и сопровождения научных геологических исследований на Дальнем Востоке России (<http://fareast-geology.ru>) расположена на виртуальных серверах в Центре обработки данных Института автоматизации и процессов управления ДВО РАН. Диски объединены в дисковый массив RAID-6. Серверы включены в Академсеть г. Владивостока (10 Гбит/с) и выходят на соединения: ТрансТелеком (180 Мбит/с), Beeline (100 Мбит/с), РосТелеком (10 Мбит/с), а также в Единую научно-образовательную сеть г. Владивостока.

*Работа выполняется при финансовой поддержке Гранта РФФИ № 14-07-00068.*

### **Список литературы**

1. Шокин Ю. И., Федотов А. М. К вопросу о развитии информационной инфраструктуры Сибирского отделения РАН // Вычислительные технологии, 2009. Т. 14, № 6. С. 127–137.
2. Лаверов Н. П., Арский Ю. М., Савин Г. И., Жижченко А. Б. Интегральное информационное поле в науках о Земле // Вестник РАН, 2008. Т. 78, № 10. С. 875–879.

3. Вершинин А. В., Серебряков В. А., Ряховский В. М., Дьяконов И. А., Динь ле Дат, Шкотин А. В., Шульга Н. Ю. Создание среды интеграции распределенных источников пространственных данных и приложений // Открытое образование, 2008. № 4. С. 9—16.
4. Добрецов Н. Н., Потатуркин О. И., Чубаров Л. Б., Шокин Ю. И. О проекте распределенной информационно-вычислительной системы сбора, хранения и обработки данных дистанционного зондирования Земли для регионов Сибири и Дальнего Востока // Вестник КАЗНУ им. Аль-Фараби. Сер. математика, механика, информатика, Алматы—Новосибирск, 2008, № 3 (58). Ч. I. С. 492—496.
5. Вдовицын В. Т., Лебедев В. А. Технологии систематизации и поиска электронной научной информации с применением онтологий // Информационные ресурсы России. 2010. № 5. С. 6—10.
6. Вдовицын В. Т., Крижановская Н. Б., Старкова В. Б. Онтологически-ориентированный подход для построения систем полнотекстового информационного поиска электронных документов // Информационные ресурсы России. 2014. № 5. С. 33—40.
7. Наумова В. В., Горячев И. Н., Платонов К. А. Web-интеграция неоднородных научных данных и сервисов по геоло-

гии Дальнего Востока России на основе порталного решения // Геоинформатика. 2011. Т. 21, № 1. С. 56—62.

8. Наумова В. В., Горячев И. Н., Дьяков С. Е., Белоусов А. В., Платонов К. А. Современные технологии формирования информационной инфраструктуры для поддержки и сопровождения научных геологических исследований на Дальнем Востоке России // Информационные технологии, 2015. № 7. С. 551—559.

9. Наумова В. В., Дьяков С. Е. Организация доступа и анализ данных дистанционного зондирования для научных исследований в геологии на Дальнем Востоке России // Вестник КРАУНЦ, Науки о Земле. 2015. № 1. Вып. 25. С. 60—65.

10. Naumova V. V., Belousov A. V. Digital repository "Geology of the Russian Far East" — an open access to the spatially distributed online scientific publications // Russian Journal of Earth Sciences. 2014. Vol. 14, N. 1, P. 1—8.

11. Platonov K. A. Methods and technologies for creation of the information processing system applied to publications on geology of the Russian Far East // Russian Journal of Earth Sciences. 2015. Vol. 15, ES4005, doi: 10.2205/2015ES000560.

12. Скачков Д. М., Жижимов О. Л. Об интеграции географических метаданных посредством ретроспективного тезауруса // Информатика и ее применения, 2012. Т. 6. Вып. 3. Р. 43—51.

V. V. Naumova<sup>1</sup>, Professor, Head of Laboratory, e-mail: naumova@fegi.ru,

S. E. Dyakov<sup>2</sup>, Scientific Researcher, e-mail: sergdkv@gmail.com,

K. A. Platonov<sup>1</sup>, Junior Researcher, e-mail: platonov@fegi.ru

<sup>1</sup>Far East Geological Institute FEBRAS, Vladivostok

<sup>2</sup>Institute Automation and Control Processes FEBRAS, Vladivostok

## Technologies of Search and Submission of Information in the Scientific Infrastructure "Geology of the Russian Far East"

*The block scheme based on distinction in types of geological data: spatial, quantitative, bibliographic and expert knowledge, is an approach basis for realization of Infrastructure. In each separate information block of Infrastructure various approaches and technological decisions are applied to integration, storage and search of the data. The concept of the organization of a uniform point of access to this Infrastructure is in providing to the user all possible information on geological objects of the Far East of Russia and services of its processing at a time, in one place, in a uniform point of access. Technologies of search and submission of information for this Infrastructure are described in article.*

**Keywords:** information infrastructure, integration of non-uniform resources of information, geological information, geology of the Russian Far East

### References

1. Shokin Ju. I., Fedotov A. M. K voprosu o razvitii informacionnoj infrastruktury Sibirskogo otdelenija RAN, *Vychislitel'nye tehnologii*, 2009, vol. 14, no. 6, pp. 127—137.
2. Laverov N. P., Arskij Ju. M., Savin G. I., Zhizhchenko A. B. Integral'noe informacionnoe pole v naukah o Zemle, *Vestnik RAN*, 2008, vol. 78, no. 10, pp. 875—879.
3. Vershinin A. V., Serebrjakov V. A., Rjahovskij V. M., D'jakonov I. A., Din' le Dat, Shkotin A. V., Shul'ga N. Ju. Sozdanie sredy integracii raspredelennyh istochnikov prostranstvennyh dannyh i prilozhenij, *Otkrytoe obrazovanie*, 2008, no. 4, pp. 9—16.
4. Dobrecov N. N., Potaturkin O. I., Chubarov L. B., Shokin Ju. I. O proekte raspredelennoj informacionno-vychislitel'noj sistemy sbora, hranenija i obrabotki dannyh distancionnogo zondirovanija Zemli dlja regionov Sibiri i Dal'nego Vostoka, *Vestnik KAZNU im. Al'-Farabi*. Ser. matematika, mehanika, informatika № 3 (58), Almaty-Novosibirsk, 2008, ch. I. S. 492—496.
5. Vdovicyn V. T., Lebedev V. A. Tehnologii sistemacizacii i poiska jelektronnoj nauchnoj informacii s primeneniem ontologij, *Informacionnye resursy Rossii*, 2010, no. 5, pp. 6—10.
6. Vdovicyn V. T., Krizhanovskaja N. B., Starkova V. B. Ontologicheski-orientirovannyj podhod dlja postroenija sistem polnotekstovogo informacionnogo poiska jelektronnyh dokumentov, *Informacionnye resursy Rossii*, 2014, no. 5, pp. 33—40.

7. Naumova V. V., Gorjachev I. N., Platonov K. A. Web-integracija neodnorodnyh nauchnyh dannyh i servisov po geologii Dal'nego Vostoka Rossii na osnove portal'nogo reshenija, *Geoinformatika*, 2011, no. 1, pp. 56—62.

8. Naumova V. V., Gorjachev I. N., D'jakov S. E., Belousov A. V., Platonov K. A. Sovremennye tehnologii formirovanija informacionnoj infrastruktury dlja podderzki i soprovozhdenija nauchnyh geologicheskijh issledovanij na Dal'nem Vostoke Rossii, *Informacionnye tehnologii*, 2015, vol. 21, no. 7, pp. 551—559.

9. Naumova V. V., D'jakov S. E. Organizacija dostupa i analiz dannyh distancionnogo zondirovanija dlja nauchnyh issledovanij v geologii na Dal'nem Vostoke Rossii, *Vestnik KRAUNC, Nauki o Zemle*, 2015, no. 1, is. 25, pp. 60—65.

10. Naumova V. V., Belousov A. V. Digital repository "Geology of the Russian Far East" — an open access to the spatially distributed online scientific publications, *Russian Journal of Earth Sciences*, 2014, vol. 14, no. 1, pp. 1—8.

11. Platonov K. A. Methods and technologies for creation of the information processing system applied to publications on geology of the Russian Far East, *Russian Journal of Earth Sciences*, 2015, vol. 15, ES4005, doi: 10.2205/2015ES000560.

12. Skachkov D. M., Zhizhimov O. L. Ob integracii geograficheskijh metadannyh posredstvom retrospektivnogo tezaurusa, *Informatika i ejo primenenija*, 2012, vol. 6, is. 3, pp. 43—51.

# МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 539.3

DOI:10.17587/it.23.21-27

Г. Г. Булычев, д-р физ.-мат. наук, проф.,  
Московский технологический университет (МИРЭА)

## Метод пространственных характеристик в задачах механики деформируемого твердого тела. Часть 1

*Анализируется метод пространственных характеристик и изучаются его возможности для исследования волновых процессов в деформируемых твердых телах различной структуры и реологии. Рассматриваются достоинства и недостатки метода, а также определяется круг задач, которые необходимо решить для построения характеристической формы математической модели динамики и динамического разрушения исследуемых тел и использования ее в численной реализации. Приводится исходная и характеристическая форма уравнений динамики анизотропных симметрично упругих тел. С помощью рассмотренного метода численно на ПЭВМ решается задача исследования разрушения двухэтажного дома при сейсмической нагрузке.*

**Ключевые слова:** динамические процессы, метод пространственных характеристик, численное моделирование, динамика и разрушение строительных сооружений и конструкций

### Введение

Метод пространственных характеристик (МПХ) предполагает построение и дальнейшее использование характеристической формы исходных уравнений динамики деформируемых твердых тел, т. е. такой формы, при которой исходные уравнения преобразуются к виду, содержащему уравнения для волн, распространяющихся в теле, и вырожденных волн, характеризующих нераспространяющиеся возмущения (они называются неподвижными разрывами). Первичная словесная формулировка такого представления была дана Х. Гюйгенсом ("Трактат о свете", 1678 г.) и звучала примерно так: "Освещенность тела в точке определяется суммой волн, приходящих в эту точку, и источниками, находящимися в этой точке". Математическая формулировка этого утверждения была получена спустя 70 лет Д'Аламбером как общее решение волнового уравнения (формула Д'Аламбера) и получила название характеристической формы, а метод, использующий эту форму представления, стал называться методом характеристик (для одномерных динамических задач) и, позднее, методом пространственных характеристик (МПХ) для многомерных задач. МПХ с успехом использовали для решения линейных и нелинейных задач динамики газов и жидкостей. Начиная с 50-х годов XX века этим методом стали исследовать динамику деформируемых твердых тел.

В 1990 г. автором был построен математический аппарат [1], позволивший в дальнейшем получить характеристическую форму анизотропных упруго-вязкопластических тел с симметричной матрицей жесткости (то есть для безмоментных сред) [2], и, наконец, после усовершенствования указанного аппарата [3] для сред с произвольной (но не вырожденной и положительно определенной) матрицей жесткости [4]. Таким образом, характеристическое представление для любых деформируемых твердых тел в рамках малых деформаций (дифференциальная часть исходных уравнений должна быть линейной) было получено.

Любопытными оказались два момента: 1) примененный аппарат может быть использован для построения характеристической формы уравнений динамики любых сред (газа, жидкости, плазмы и твердого тела) в том случае, когда дифференциальные части их уравнений описываются дифференциальными инвариантами первого порядка; 2) для решения этой задачи не понадобилось ни тензорное исчисление, ни определение собственных значений и собственных векторов матриц, порядок которых превышал бы третий. Таким образом, на этом этапе удалось обойтись аналитическими, причем тождественными и без дополнительных предположений, преобразованиями. Оказалось даже, что для несимметрично упругих сред указанные преобразования были более простыми, чем для симметрично упругих.

Применительно к такой (характеристической) форме представления исходных уравнений динамики автором был разработан численный метод, используемый для решения задач строительной механики, одна из которых в качестве примера рассмотрена в настоящей работе.

### Построение характеристической формы уравнений динамики упруговязкопластических тел. Математический аппарат

Построим матричную форму дифференциальных инвариантов первого порядка, для чего введем в рассмотрение скаляр  $\varphi$ , вектор  $V$ , симметричный тензор  $\hat{\sigma}$  и несимметричный тензор  $\hat{\rho}$ . Введем для простоты декартову систему координат  $\{x_i\}$  и представим вектор и тензоры в виде матриц-строк:

$$V = \|V_1, V_2, V_3\|, \sigma = \|\Sigma_{11}, \Sigma_{22}, \Sigma_{33}, \Sigma_{12}, \Sigma_{13}, \Sigma_{23}\|; \quad (1)$$

$$p = \|P_{11}, P_{22}, P_{33}, P_{12}, P_{13}, P_{21}, P_{23}, P_{31}, P_{32}\|.$$

Для выбранных объектов составим дифференциальные инварианты, разбивая их на три группы: уменьшающие размерность объектов

$$\text{div}V = (e_i \partial_i)V^T; \quad \text{div}\hat{\sigma} = (R_i \partial_i)\Sigma^T; \quad \text{div}\hat{\rho} = (Q_i \partial_i)P^T; \quad (2)$$

сохраняющие размерность объектов

$$\text{rot}V = (S_i \partial_i)V^T; \quad \text{rot}\hat{\sigma} = (\partial_{ijk} R_i^T R_k \partial_j)\Sigma^T;$$

$$\text{rot}\hat{\rho} = (\partial_{ijk} Q_i^T Q_k \partial_j)P^T; \quad (3)$$

увеличивающие размерность объектов

$$\text{grad}\varphi = (e_i^T \partial_i)\varphi; \quad \text{def}V = (\Pi R_i^T \partial_i)V^T;$$

$$\text{grad}V = (Q_i^T \partial_i)V^T, \quad (4)$$

где  $\partial_i = \partial/\partial x_i$ ,  $e_i = \|\delta_{1i}, \delta_{2i}, \delta_{3i}\|$ ,  $S_i = \partial_{ijk} e_j^T e_k$ ,  $\delta_{ij}$  — единичный тензор, равный 1 при  $i = j$  и 0 при  $i \neq j$ ,  $\partial_{ijk}$  — тензор круговой перестановки Леви—Чевита; по повторяющимся латинским индексам здесь и далее проводится суммирование,

$$R_i = \left\| \begin{array}{cccccc} \delta_{1i} & 0 & 0 & \delta_{2i} & \delta_{3i} & 0 \\ 0 & \delta_{2i} & 0 & \delta_{1i} & 0 & \delta_{3i} \\ 0 & 0 & \delta_{3i} & 0 & \delta_{1i} & \delta_{2i} \end{array} \right\|,$$

$$Q_i = \left\| \begin{array}{cccccc} \delta_{1i} & 0 & 0 & 0 & 0 & \delta_{2i} & 0 & \delta_{3i} & 0 \\ 0 & \delta_{2i} & 0 & \delta_{1i} & 0 & 0 & 0 & 0 & \delta_{3i} \\ 0 & 0 & \delta_{3i} & 0 & \delta_{1i} & 0 & \delta_{2i} & 0 & 0 \end{array} \right\|;$$

$\Pi = \text{diag}(1, 1, 1, 1/2, 1/2, 1/2)$ , верхний индекс "Т" обозначает транспонирование.

Легко заметить, что матрицы  $e_i$ ,  $S_i$ ,  $R_i$ ,  $Q_i$  имеют весьма простой вид, но именно они определяют топологию волновых процессов, как циклического

порождения одних величин другими, например в твердом теле: напряжения—перемещения—деформации—напряжения. Любопытно, что переход к ортогональной криволинейной системе координат не увеличивает число этих матриц и также позволяет проводить указанные ниже аналитические преобразования [2].

Используем соотношения (1), (2) и (4) для построения характеристической формы уравнений динамики симметрично-упругого упруговязкопластического тела. Исходную систему уравнений запишем в виде уравнений движения, условий аддитивности деформаций, закона Гука для упругих деформаций, соотношений Коши для упругих деформаций и закона пластического течения

$$\text{div}\hat{\sigma} + \mathbf{f} = \rho \partial_t \mathbf{v},$$

$$\hat{\varepsilon} = \hat{\varepsilon}^e + \hat{\varepsilon}^{vp}, \quad \hat{\sigma} = \hat{c} \cdot \hat{\varepsilon}^e, \quad \partial_t \hat{\varepsilon} = \text{def} \mathbf{v}, \quad (5)$$

$$\partial_t \hat{\varepsilon}^{vp} = \hat{\Phi}(\hat{\varepsilon}, \hat{\varepsilon}^e, \hat{\varepsilon}^{vp}, \hat{\sigma}, \mathbf{v}, \eta).$$

Здесь  $\mathbf{f}$  — вектор внутренних сил;  $\rho$  — плотность материала тела;  $\hat{\sigma}$  — тензор напряжений;  $\mathbf{v}$  — вектор скоростей частиц тела;  $\hat{\varepsilon}$ ,  $\hat{\varepsilon}^e$  и  $\hat{\varepsilon}^{vp}$  — полная, упругая и вязкопластическая деформации;  $\hat{c}$  — тензор жесткости;  $\hat{\Phi}$  — тензор вязкопластических деформаций (существенно, что он не зависит от производных  $\hat{\sigma}$  и  $\mathbf{v}$ ),  $\partial_t = \partial/\partial t$ .

Видно, что в системе (5) два дифференциальных инварианта — дивергенция ( $\text{div}$ ) и деформация ( $\text{def}$ ), топология которых определяется матрицами  $R_i$ .

Относительно матричных переменных  $V$  и  $\Sigma$  в системе координат  $\{x_i\}$  систему (5) можно привести в двум группам уравнений: уравнения движения и определяющие уравнения:

$$R_i \partial_i \Sigma^T - \rho \partial_t V^T = F^T; \quad \partial_t \Sigma^T = C(\Pi R_i^T \partial_i V^T - \Phi^T), \quad (6)$$

где  $C$  — матрица жесткостей;  $F$  и  $\Phi$  — матрицы-строки внутренних сил и скоростей вязкопластических деформаций.

Выберем направление  $x_\alpha$ , совпадающее с одной из осей координатной системы  $\{x_i\}$ , и построим характеристические уравнения для волн, распространяющихся в обе стороны вдоль этого направления. Остальные оси назовем, соответственно,  $x_\beta$  и  $x_\gamma$ ; напряжения, действующие на площадках с нормальми, соответствующими этим осям  $R_\alpha \Sigma^T$ ,  $R_\beta \Sigma^T$ ,  $R_\gamma \Sigma^T$ , обозначим, соответственно,  $\Sigma_\alpha^T$ ,  $\Sigma_\beta^T$  и  $\Sigma_\gamma^T$ . Матрицу

СП обозначим  $\tilde{C}$  и перепишем (6) в виде

$$\partial_\alpha \Sigma_\alpha^T - \rho \partial_t V^T = F^T - \partial_\beta \Sigma_\beta^T - \partial_\gamma \Sigma_\gamma^T;$$

$$\partial_t \Sigma^T = \tilde{C} R_i^T \partial_i V^T - C \Phi^T. \quad (7)$$

Умножим второе уравнение системы (7) слева на  $R_\alpha$  и введем обозначения  $\tilde{C}_{\alpha\alpha} = R_\alpha \tilde{C} R_\alpha^T$ ;  $\tilde{C}_{\alpha\beta} = R_\alpha \tilde{C} R_\beta^T$ ;  $\tilde{C}_{\alpha\gamma} = R_\alpha \tilde{C} R_\gamma^T$  и  $R_\alpha \tilde{C} \tilde{\Phi}^T = \Psi_\alpha^T$ , при этом второе уравнение (7) приводится к виду

$$\partial_t \Sigma_\alpha^T - \tilde{C}_{\alpha\alpha} \partial_\alpha V^T = \tilde{C}_{\alpha\beta} \partial_\beta V^T + \tilde{C}_{\alpha\gamma} \partial_\gamma V^T - \Psi_\alpha^T. \quad (8)$$

Рассмотрим матрицу  $\tilde{C}_{\alpha\alpha}$ , которая является главным минором третьего порядка матрицы  $\tilde{C}$ . Матрица  $\tilde{C}$  всегда невырождена, положительно определена и симметрична по построению, следовательно, матрица  $\tilde{C}_{\alpha\alpha}$  имеет диагональное разложение:  $\tilde{C}_{\alpha\alpha} = O^T \Lambda O$ , где  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$  — диагональная матрица с положительными собственными числами (пусть  $\lambda_1 \geq \lambda_2 \geq \lambda_3$ , что всегда возможно, если все элементы  $\tilde{C}_{\alpha\alpha}$  положительны), а матрицы собственных векторов  $O$  — ортонормированны и являются матрицами поворота в координатной системе  $\{x_j\}$ .

Примем  $\Lambda = \rho D^2$ , откуда  $D = \pm \sqrt{\Lambda/\rho}$ , имеет размерность скорости и, как будет видно из дальнейшего, является матрицей продольных и поперечных волн, распространяющихся в обоих направлениях вдоль оси  $x_\alpha$ .

Умножим первое уравнение (7) слева на  $|D|O$ , уравнение (8) слева на  $O$ , сложим эти уравнения и заменим  $|D|$  на  $\pm D$ . После группировки получим матричные характеристические уравнения

$$(\partial_t \pm D \partial_\alpha)(O \Sigma_\alpha^T \mp \rho D O V^T) = \partial_\beta (O \tilde{C}_{\alpha\beta} V^T \mp D O \Sigma_\beta^T) + \partial_\gamma (O \tilde{C}_{\alpha\gamma} V^T \mp D O \Sigma_\gamma^T) \pm D O F^T - O \Psi_\alpha^T. \quad (9)$$

Пусть численные расчеты уравнений (9) приводят к следующим формулам:

$$O \Sigma_\alpha^T - \rho D O V^T = \Omega_1, \quad O \Sigma_\alpha^T + \rho D O V^T = \Omega_2, \quad (10)$$

тогда

$$\Sigma_\alpha^T = O^T (\Omega_1 + \Omega_2) / 2, \quad V^T = O^T D^{-1} (\Omega_2 - \Omega_1) / (2\rho)$$

и шесть из девяти определяемых величин оказываются найденными (численно).

Для определения остальных напряжений введем матрицу  $N_\alpha$ , полученную вычеркиванием из единичной матрицы шестого порядка строк, соответствующих матрице  $R_\alpha$ . Тогда, по построению,  $\Sigma_N^T = N_\alpha \Sigma^T$  — те напряжения, которые осталось вычислить.

Умножим второе уравнение (7) слева на  $N_\alpha$  и выразим  $\partial_\alpha V^T$  из (8); в результате после группировки получим матричное вырожденное характеристиче-

ское уравнение, определяющее последние три напряжения  $\Sigma_N^T$ :

$$\begin{aligned} \partial_t (\Sigma_N^T - \tilde{C}_{N\alpha} \tilde{C}_{\alpha\alpha}^{-1} \Sigma_\alpha^T) = \\ = (\tilde{C}_{N\beta} - \tilde{C}_{N\alpha} \tilde{C}_{\alpha\alpha}^{-1} \tilde{C}_{\alpha\beta}) \partial_\beta V^T + (\tilde{C}_{N\gamma} - \\ - \tilde{C}_{N\alpha} \tilde{C}_{\alpha\alpha}^{-1} \tilde{C}_{\alpha\gamma}) \partial_\gamma V^T + \tilde{C}_{N\alpha} \tilde{C}_{\alpha\alpha}^{-1} \Psi_\alpha^T - \Sigma_N^T. \end{aligned} \quad (11)$$

Здесь обозначено:  $\tilde{C}_{N\alpha} = N_\alpha \tilde{C} R_\alpha^T$ ,  $\tilde{C}_{N\beta} = N_\alpha \tilde{C} R_\beta^T$ ,  $\tilde{C}_{N\gamma} = N_\alpha \tilde{C} R_\gamma^T$ ,  $\Sigma_N^T = N_\alpha \tilde{C} \tilde{\Phi}^T$ .

Большинство материалов, используемых в промышленности и строительстве, являются изотропными, трансверсально-изотропными или ортотропными. Для таких материалов матрица  $\tilde{C}$  имеет вид

$$\tilde{C} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{12} & c_{22} & c_{23} & 0 & 0 & 0 \\ c_{13} & c_{23} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66} \end{pmatrix} \quad (12)$$

и  $\tilde{C}_{11} = \text{diag}(c_{11}, c_{44}, c_{55})$ ,  $\tilde{C}_{22} = \text{diag}(c_{44}, c_{22}, c_{66})$ ,  $\tilde{C}_{33} = \text{diag}(c_{55}, c_{66}, c_{33})$ , а следовательно, собственными векторами являются единичные матрицы третьего порядка и уравнения (9) и (11) распадаются на девять скалярных уравнений:

$$\begin{aligned} (\partial_\alpha \mp c_1^{-1} \partial_t)(\Sigma_{\alpha\alpha} \pm \rho c_1 V_\alpha) + \partial_j (\Sigma_{aj} \pm v_1 \rho c_1 V_j) + \\ + \partial_k (\Sigma_{ak} \pm v_2 \rho c_1 V_k) = \\ = \tilde{F}_\alpha \pm \rho c_1 (\Phi_{\alpha\alpha} + v_1 \Phi_{jj} + v_2 \Phi_{kk}); \end{aligned} \quad (13)$$

$$\begin{aligned} (\partial_\alpha \mp c_2^{-1} \partial_t)(\Sigma_{\alpha j} \pm \rho c_2 V_j) + \partial_j (\Sigma_{jj} \pm \rho c_2 V_\alpha) + \partial_k \Sigma_{jk} = \\ = F_j \pm 2\rho c_2 \Phi_{\alpha j}; \end{aligned}$$

$$\begin{aligned} (\partial_\alpha \mp c_3^{-1} \partial_t)(\Sigma_{\alpha k} \pm \rho c_3 V_k) + \partial_k (\Sigma_{kk} \pm \rho c_3 V_\alpha) + \partial_j \Sigma_{jk} = \\ = F_k \pm \rho c_3 \Phi_{\alpha k}; \end{aligned}$$

$$\partial_t (\Sigma_{jj} - v_1 \Sigma_{\alpha\alpha}) = \eta_1 (\partial_j V_j - \Phi_{jj}) + \eta_2 (\partial_k V_k - \Phi_{kk});$$

$$\partial_t (\Sigma_{kk} - v_2 \Sigma_{\alpha\alpha}) = \eta_2 (\partial_j V_j - \Phi_{jj}) + \eta_3 (\partial_k V_k - \Phi_{kk});$$

$$\partial_t \Sigma_{jk} = c_{\chi\chi} (\partial_j V_k + \partial_k V_j - \Phi_{jk}), \quad \alpha, j, k = 1, 2, 3;$$

где  $j \neq \alpha$ ,  $k \neq \alpha$ ,  $j < k$ ,  $\beta = \alpha + j + 1$ ,  $\gamma = \alpha + k + 1$ ,  $\chi = j + k + 1$ ,  $c_1 = \sqrt{c_{\alpha\alpha}/\rho}$ ,  $c_2 = \sqrt{c_{\beta\beta}/\rho}$ ,  $c_3 = \sqrt{c_{\gamma\gamma}/\rho}$ ,  $v_1 = c_{\alpha j}/c_{\alpha\alpha}$ ,  $v_2 = c_{\alpha k}/c_{\alpha\alpha}$ ,  $\eta_1 = c_{jj} - v_1^2 c_{\alpha\alpha}$ ,  $\eta_2 = c_{kj} - v_1 v_2 c_{\alpha\alpha}$ ,  $\eta_3 = c_{kk} - v_2^2 c_{\alpha\alpha}$ ,  $F_i$  и  $\Phi_{ij}$  — физические компоненты вектора внутренних сил и аналогич-

ные компоненты тензора вязкопластических деформаций.

Первые три пары уравнений (13) описывают волновые процессы на движущихся разрывах, т. е. для волн, распространяющихся в направлении  $x_\alpha$  декартовой системы координат  $\{x_j\}$ ; в каждой паре вдоль бихарактеристик изменяются по две определяемые переменные и, следовательно, эти переменные могут быть определены независимо от остальных.

Три последних уравнения системы (13) независимо друг от друга определяют три напряжения  $\Sigma_{jj}$ ,  $\Sigma_{kk}$  и  $\Sigma_{jk}$  по определенному из первой пары уравнений напряжению  $\Sigma_{\alpha\alpha}$ . Таким образом, характеристическая система уравнений (13) позволяет определить все напряжения и скорости частиц во внутренней точке рассматриваемой среды для волн, распространяющихся в обе стороны вдоль оси  $x_\alpha$ .

В случае если матрица  $\tilde{C}_{\alpha\alpha}$  несимметрична (что бывает в случае несимметрично-упругих тел), следует использовать треугольное разложение  $\tilde{C}_{\alpha\alpha} = O^T \Delta O$ , в котором  $O$  — ортонормированные матрицы собственных векторов, а матрица собственных чисел  $\Delta$  имеет вид

$$\Delta = \begin{pmatrix} \lambda_1 & 0 & 0 \\ \omega_1 & \lambda_2 & 0 \\ \omega_2 & \omega_3 & \lambda_3 \end{pmatrix},$$

где коэффициенты  $\lambda_1, \lambda_2, \lambda_3$  — по-прежнему определяют скорости продольной и поперечных волн, а коэффициенты  $\omega_1, \omega_2, \omega_3$  — характеризуют взаимодействие различных типов волн внутри тела. В случае таких матриц  $\tilde{C}_{\alpha\alpha}$  также возможно аналитическое построение матричных характеристических уравнений [4]. Подобные матрицы характерны для описания динамики кристаллов и могут быть использованы для аналитического исследования их разнообразных свойств.

### Достоинства и недостатки характеристической формы представления уравнений динамики деформируемых твердых тел

Математические аспекты МПХ разрабатывали К. М. Магомедов и А. С. Холодов для уравнений типа переноса [5]. На основе сравнительного анализа различных разностных схем ими было показано, что область зависимости конечноразностных характеристических схем наиболее близка к области зависимости исходных дифференциальных уравнений, т. е. точность вычислений с использованием характеристических схем выше, чем с использованием сеточных. Кроме того оказалось, что при первом порядке аппроксимации (односторонняя разность по времени и симметричная разность по координате) характеристические конечнораз-

ностные схемы имеют остаточный член параболического (а не гиперболического!) типа, что делает их устойчивыми при моделировании как гладких, так и разрывных возмущений. Оказалось также, что разностно-характеристические схемы первого порядка не создают вычислительную "рябь", которые приводят сеточные схемы в тех же случаях к осцилляторной неустойчивости.

Являясь обобщением метода характеристик, МПХ обладает всеми его достоинствами: 1) в характеристических уравнениях, по сравнению с исходными на единицу уменьшено число независимых переменных; 2) вдоль характеристики (за исключением кристаллов сложного строения, но там также возможны упрощения) изменяются только две переменные — напряжение и вызванная им скорость частиц, т. е. уравнения оказываются связанными попарно для волн, распространяющихся вдоль одной оси, но в противоположных направлениях; 3) уравнения имеют дивергентный вид, что позволяет при составлении алгоритмов использовать интегральные теоремы и сохранять свойство консервативности в конечно-разностных схемах.

С точки зрения численной реализации эти достоинства позволяют:

- существенно сблизить области зависимости исходных дифференциальных уравнений и их конечно-разностных аналогов, тем самым резко повышая точность вычислений (особенно при моделировании волновой динамики и волнового разрушения);
- резко сократить время счета и необходимый объем оперативной памяти;
- точно передавать скачки решений и отслеживать изменение условий на разрывах (трещинах, неоднородностях, границах раздела сред и т. д.) вплоть до полного разрушения элемента или конструкции;
- сохранять консервативность незатухающих процессов;
- не бояться неустойчивостей счета при использовании схем первого порядка аппроксимации.

### Недостатки МПХ

1. Жесткая связь между шагом по времени и шагом по координатам, что накладывает ограничения на пространственную сетку. Например, для кусочно-однородных сред эта сетка также должна быть кусочно-постоянной, со строго согласованным шагом.

2. Избыточность характеристических соотношений при рассмотрении волн в ограниченных телах. Так, для каждого направления можно построить столько же характеристических соотношений, сколько определяемых переменных. Поэтому для волн, распространяющихся в разных направлениях, получается избыточное число характеристических уравнений. В этом случае выбор необходимых уравнений определяется либо из контекста задачи, либо из дополнительных условий (обычно устойчи-



ности и точности вычислительных схем), либо из сравнения с известными решениями тех же задач.

3. Трудность согласования области распространения волн с криволинейной сеткой при работе в криволинейной ортогональной системе координат.

Однако использование для интегрирования дивергентной формы характеристических уравнений (формула Гаусса—Остроградского) в значительной мере устраняет эту проблему.

### Этапы разработки вычислительного эксперимента

Для применения МПХ к моделированию динамики и динамического разрушения тел сложной структуры, реологии и геометрии разработчику необходимо решить пять взаимосвязанных задач:

1. Построить дифференциальную характеристическую форму уравнений динамики исследуемой среды.

2. Выбрать необходимые уравнения и дополнительные условия во всех характерных точках этой среды.

3. При моделировании динамики разрушения выбрать критерии разрушения (расслоения) и последовательность условий на трещинах при различных напряжениях на их берегах и положении самих берегов.

4. Разработать схему проекта с учетом пп. 2 и 3.

5. Построить характеристические конечно-разностные схемы во всех характерных точках среды и при всех возможных состояниях среды в этих точках; составить подпрограммы, организующие ввод данных, расчет напряженно-деформированного состояния во всех точках пространственной сетки в последовательные моменты времени и вывод полученных результатов в требуемой форме. Объединить весь этот алгоритмический материал (как набор подпрограмм) в единую программу и протестировать ее на задачах с известными или интуитивно известными решениями.

К настоящему времени первая задача решена в рамках малых деформаций для всех видов упруго-вязкопластических деформируемых твердых тел [2, 4].

Если в рассматриваемой точке распространяются волны нескольких направлений или должны выполняться какие-нибудь дополнительные условия, система (13) оказывается переопределенной и выбор тех или иных из входящих в нее уравнений может быть сделан только после дополнительного анализа процессов, происходящих в этой точке. Таким образом, разработчик метода сталкивается со второй из указанных задач. Для ее решения необходимо привлечь условия достижения устойчивости и максимальной точности характеристических вычислительных схем в типичных точках рассматриваемого тела. Этим вопросам посвящено несколько публикаций [5—7]. Так, в работе [5] было показано, что необходимым условием устойчивости разностных характеристических схем во внутренней

точке среды является использование всех независимых уравнений для продольных волн, распространяющихся в этой точке. В работе [6] была предложена характеристическая вычислительная схема в граничной точке, показано, что для построения этой схемы следует использовать те уравнения (13), для которых  $x_\alpha$  совпадает с внутренней нормалью к границе; при этом характеристические уравнения следует дополнить граничными условиями

$$\zeta_\delta \Sigma_{\alpha\delta} + \omega_\delta V_\delta = \psi_\delta, \delta = 1, 2, 3, \quad (14)$$

в которых  $\zeta_\delta$  и  $\omega_\delta$  могут быть константами, функциями координат и времени или операторами времени, а  $\psi_\delta$  — константой или функцией.

В работе [7], в соответствии с указаниями, приведенными в работах [5] и [6], для плоских и осесимметричных тел были построены характеристические вычислительные схемы во внутренних и граничных точках. Там же было показано, что в точках на оси симметрии использование правила Лопиталья к неопределенностям типа  $V_R/R$ ,  $\Sigma_{RZ}/R$  и  $(\Sigma_{RR} - \Sigma_{\Theta\Theta})/R$  приводит к тому, что некоторые характеристические уравнения совпадают друг с другом, в силу чего три из них могут быть исключены и заменены условиями  $V_R = 0$ ;  $\Sigma_{RZ} = 0$  и  $\Sigma_{RR} = \Sigma_{\Theta\Theta}$ .

В работе [8] для плоских и осесимметричных кусочно-однородных тел была предложена вычислительная схема в точке контакта слоев. Согласно этой схеме вычислительный шаблон в точке контакта слоев представлялся в виде двух схем граничных точек, расположенных в одном месте в смежных слоях и соединенных условиями контакта:

$$\Sigma_{\alpha i}^+ = \Sigma_{\alpha i}^-, V_i^+ = V_i^-, i = \alpha, \beta. \quad (15)$$

Впоследствии эта схема была усовершенствована и приспособлена к задачам динамического разрушения [9]: в нее были введены критерии разрушения (на отрыв и на сдвиг) и условия автоматического выбора последовательности условий на контакте; также она была обобщена на случай трехмерных задач. При расслоении в зависимости от типа разрушения эта последовательность условий задается с помощью одной из таблиц (табл. 1 или табл. 2).

Таблица 1

#### Трещина отрыва

Сжатие: $\Delta u_\alpha < 0$		Растяжение: $\Delta u_\alpha \geq 0$
$\Sigma_{\alpha\alpha} < -\Sigma^*$	$-\Sigma^* \leq \Sigma_{\alpha\alpha} \leq 0$	
$\Sigma_{\alpha\alpha}^+ = \Sigma_{\alpha\alpha}^-$	$\Sigma_{\alpha\alpha}^+ = \Sigma_{\alpha\alpha}^-$	$\Sigma_{\alpha\alpha}^+ = 0$
$\Sigma_{\alpha\beta}^+ = \tau^*$	$\Sigma_{\alpha\beta}^+ = \kappa \Sigma_{\alpha\alpha}^+$	$\Sigma_{\alpha\alpha}^- = 0$
$\Sigma_{\alpha\beta}^- = \tau^*$	$\Sigma_{\alpha\beta}^- = \kappa \Sigma_{\alpha\alpha}^-$	$\Sigma_{\alpha\beta}^+ = 0$
$V_\alpha^+ = V_\alpha^-$	$V_\alpha^+ = V_\alpha^-$	$\Sigma_{\alpha\beta}^- = 0$

Таблица 2

## Трещина сдвига

Сжатие: $\Delta u_\alpha < 0$	Растяжение: $\Delta u_\alpha \geq 0$
$\Sigma_{\alpha\alpha}^+ = \Sigma_{\alpha\alpha}^-$	$\Sigma_{\alpha\alpha}^+ = 0$
$\Sigma_{\alpha\beta}^+ = -\gamma  \Sigma_{\alpha\beta}^+  \operatorname{sgn}(V_\beta^+ - V_\beta^-)$	$\Sigma_{\alpha\alpha}^- = 0$
$\Sigma_{\alpha\beta}^- = -\gamma  \Sigma_{\alpha\beta}^-  \operatorname{sgn}(V_\beta^+ - V_\beta^-)$	$\Sigma_{\alpha\beta}^+ = 0$
$V_\alpha^+ = V_\alpha^-$	$\Sigma_{\alpha\beta}^- = 0$

В таблицах обозначено:  $\Sigma_{\alpha\alpha}^+$ ,  $\Sigma_{\alpha\alpha}^-$ ,  $\Sigma_{\alpha\beta}^+$ ,  $\Sigma_{\alpha\beta}^-$  — нормальные и касательные напряжения с обеих сторон трещины отрыва, распространяющейся вдоль оси  $x_\beta$ ;  $u_\alpha^+$ ,  $u_\alpha^-$ ,  $u_\beta^+$ ,  $u_\beta^-$  — перемещения вдоль осей  $x_\alpha$  и  $x_\beta$  с обеих сторон трещины;  $\Sigma^*$ ,  $\kappa$  и  $\gamma$  — константы, полученные опытным путем,  $\tau^* = -\kappa \Sigma^*$ ,  $\Delta u_\alpha = u_\alpha^+ - u_\alpha^-$ ,  $V_\beta^+$  и  $V_\beta^-$  — скорости частиц вдоль трещины.

Аналогично вышеуказанным схемам были построены вычислительные схемы и в некоторых других типичных точках тел, однако следует заметить, что в общем случае эта проблема не решена и требует дополнительного анализа процессов в рассматриваемых точках с привлечением тестовых задач, желательно имеющих аналитическое решение, или с использованием других численных методов.

Задача выбора и стыковки пространственных сеток связана с тем, что в МПХ шаг по времени ( $h_t$ ) и шаг по координате  $x_\alpha$  ( $h_\alpha$ ) связаны соотношением  $h_\alpha = c_\alpha \cdot h_t$ , где  $c_\alpha$  — скорость продольных волн в направлении  $x_\alpha$ .

Поэтому выбор единого шага по времени для неоднородного или анизотропного тела требует различных шагов по координатам как для различных направлений (ввиду анизотропии), так, возможно, и при переходе от точки к точке (ввиду неоднородности).

При стыковке пространственных сеток на границах раздела сред с различными физико-механическими свойствами возникает несовпадение узлов сеток с разных сторон границы, вследствие чего приходится применять процедуру интерполяции между узлами, что приводит к уменьшению точности расчетов в точках на границе раздела сред.

Жесткая связь между шагами сетки по времени и по координатам оказывается неудобной также для тел с криволинейными границами, поскольку для таких тел координатная сетка неравномерная. Для таких тел также приходится применять интерполяцию узлов расчетной сетки между узлами координатной сетки, причем во всем теле.

В части 2 данной статьи будет рассмотрен пример использования МПХ при моделировании динамического разрушения сложного объекта.

## Заключение

В заключение еще раз подчеркнем те свойства характеристической формы уравнений динамики деформируемых твердых тел, которые делают ее привлекательной для численного исследования.

1. *Уменьшенное на единицу число независимых переменных*; при этом резко снижается количество вычислений и объем требуемой памяти. Кроме того, максимально сближены области зависимости дифференциальных уравнений и их конечно-разностных аналогов, что позволяет увеличить точность расчетов при моделировании динамики. Точно передаются скачки возмущений. При первом порядке аппроксимации вычислительные схемы дивергентны, имеют положительную аппроксимационную вязкость, устойчивы и консервативны.

2. *Блочный вид дифференциальных уравнений*; их дифференциальная часть записывается в сопутствующей декартовой системе координат, правая часть состоит из отдельных блоков: блока массовых сил, блока связанного с реологией и блока геометрических членов (в криволинейной системе координат). Такое строение алгоритмизируемых уравнений позволяет создавать программы, состоящие из отдельных функциональных модулей, и легко перестраиваемые при изменении реологии, массовых сил и кривизны координатной системы.

3. *Распад характеристических уравнений на независимые пары и отдельные уравнения*. Для изотропных, трансверсально-изотропных и ортотропных сред система характеристических уравнений распадается на ряд независимых скалярных пар уравнений для соответствующих пар неизвестных напряжений и скоростей частиц. Такими парами уравнений являются уравнения на движущихся разрывах, а переменными (для волн, распространяющихся в направлении  $x_\alpha$ ) —  $\Sigma_{\alpha j}$  и  $V_j$ ,  $j = 1, 2, 3$ . При этом еще более упрощается вид уравнений и сокращается объем вычислений, а для граничных точек и точек, в которых возможно разрушение, появляется возможность вводить внутрь вычислительных схем дополнительные условия (граничные условия или условия на трещине), в которых определяются те же переменные. Оставшиеся уравнения являются отдельными уравнениями для определения напряжений  $\Sigma_{jj}$ ,  $\Sigma_{kk}$  и  $\Sigma_{jk}$ , причем в каждое уравнение входит только одна из этих переменных.

4. При исследовании процессов разрушения или расслоения характеристическая форма представления исходных уравнений позволяет использовать двойную пространственную сетку и проводить моделирование при различных, меняющихся в процессе счета, условиях на границах между ячейками. Таким образом может быть смоделировано не

только образование и движение трещины, но и множественное разрушение тел.

Пример моделирования разрушения двухэтажного дома будет показан во 2-й части статьи.

#### Список литературы

1. Бульчев Г. Г. Характеристическая форма уравнений динамики анизотропных упруговязкопластических сред // Доклады АН СССР. Серия Математика и механика. 1990. Т. 314, № 2.
2. Бульчев Г. Г. Построение матричной характеристической формы уравнений динамики анизотропных упруговязкопластических сред // Известия РАН. Механика твердого тела. 1995. № 1. С. 91–95.
3. Бульчев Г. Г. Матричная форма представления дифференциальных инвариантов в задачах динамики сплошных сред // Строительная механика и расчет сооружений. 2009. № 3. С. 14–19.
4. Бульчев Г. Г. Характеристическая форма представления уравнений динамики несимметрично упругих упруговязкопла-

стических тел // Строительная механика и расчет сооружений. 2010. № 2. С. 52–56.

5. Магомедов К. М., Холодов А. С. Сеточно-характеристические численные методы. М.: Наука, 1988. 288 с.

6. Courant R., Friedrichs K. O., Lewy H. Uber die partiellen differenzengleichungen der mathematischen // Physic. Math. Ann. 1928. Vol. 100. N. 32. P. 32, русский перевод: Курант Г., Фридрихс К., Леви Г. О разностных уравнениях математической физики // УМН. 1940. Вып. 8. С. 125.

7. Butler D. S. The numerical solution of hyperbolic systems of partial differential equations in three independent variables // Proc. Roy. Soc., London, A255, 1962. P. 319–343.

8. Бульчев Г. Г. Численное исследование динамики и динамической прочности кусочно-однородных упругопластических тел // Строительная механика и расчет сооружений. 1987. № 5. С. 56–59.

9. Бульчев Г. Г., Кукуджанов В. Н. Динамическое разрушение предварительно напряженного волокнистого композита, вызванное обрывом волокна // Известия РАН. Механика твердого тела. 1993. № 3. С. 207–214.

G. G. Bulychev, DSc., Professor,  
Moscow Technological University, Moscow

## Method of Spatial Characteristics in Problems of a Mechanics of a Deformable Solid Body. Part 1

*In this article the method of spatial characteristics is considered and analyzed and its possibilities for studying of wave processes in deformable solid bodies of various structure and a rheology are investigated. Merits and demerits of a method are considered, and also the circle of tasks which it is necessary to solve for building of the characteristic form of mathematical model of dynamics and dynamic destruction of researched bodies and for its usage in numerical simulation. The initial and characteristic form of dynamical equations of the anisotropic symmetrically elastic bodies is shown. By means of the considered method numerically on PC the research problem of destruction of the two-storey house is solved at an earthquake load.*

**Keywords:** dynamic processes, a method of spatial characteristics, numerical simulation, dynamics and destruction of buildings and constructions

#### References

1. Bulychev G. G. Kharakteristicheskaya forma uravneniy dinamiki anizotropnykh uprugoviyazkoplasticheskikh sred. Doklady AN SSSR, s. *Matematika i mekhanika*, 1990, vol. 314, no. 2.
2. Bulychev G. G. Postroenie matrichnoy kharakteristicheskoy formy uravneniy dinamiki anizotropnykh uprugoviyazkoplasticheskikh sred // *Izvestiya AN Rossii, Mekhanika tverdogo tela*, 1995, no. 1, pp. 91–95.
3. Bulychev G. G. Matrichnaya forma predstavleniya differentsialnykh invariantov v zadachakh dinamiki sploshnykh sred, *Stroitel'naya mekhanika i raschet sooruzheniy*, 2009, no. 3, pp. 14–19.
4. Bulychev G. G. Kharakteristicheskaya forma predstavleniya uravneniy dinamiki nesimmetrichno uprugikh uprugoviyazkoplasticheskikh tel, *Stroitel'naya mekhanika i raschet sooruzheniy*, 2010, no. 2, pp. 52–56.

5. Magomedov K. M., Kholodov A. S. *Setochno-kharakteristicheskie chislenniye metodih*, Moscow, Nauka, 1988, 288 p.

6. Courant R., Friedrichs K. O., Lewy H. Uber die partiellen differenzengleichungen der mathematischen, *Physic. Math. Ann.* 1928, vol. 100, no. 32, p. 32.

7. Butler D. S. The numerical solution of hyperbolic systems of partial differential equations in three independent variables, *Proc. Roy. Soc., London*, A255, 1962, pp. 319–343.

8. Bulychev G. G. Chislennoe issledovanie dinamiki i dinamicheskoy prochnosti kusochno-odnorodnykh uprugoplasticheskikh tel, *Stroitel'naya mekhanika i raschet sooruzheniy*, 1987, no. 5, pp. 56–59.

9. Bulychev G. G., Kukudzhanov V. N. Dinamicheskoe razruhenie predvaritel'no napryazhennogo voloknistogo kompozita, vyzvanoe obryvom volokna. *Izvestiya RAN, MTT*, 1993, no. 3, pp. 207–214.

М. А. Стюгин, канд. техн. наук, науч. сотр., e-mail: styugin@gmail.com,  
Сибирский государственный аэрокосмический университет  
имени академика М. Ф. Решетнева, г. Красноярск

### Метод построения неразличимого исполняемого кода\*

*Предложена задача обфускации, в которой неразличимость рассматривается относительно программ различной функциональности. Доказывается теоретическая возможность построения такого обфускатора и существование алгоритмов полиномиальной сложности для его получения. Использование подобного обфускатора позволяет хранить исполняемый код в недоверенной вычислительной среде, а также затрудняет атаки инъекций и анализа распределенных вычислений.*

**Ключевые слова:** обфускация, неразличимость, недоверенные вычисления, обфускация с ключом, вычислительная сложность, эффективные алгоритмы

#### Введение

Проблема защиты кода от исследования существует достаточно давно. Последние несколько лет данное направление информационной безопасности претерпело множество изменений. Если раньше обфускация кода представляла собой в некотором роде искусство, то примерно с 2000-х годов обфускация все чаще формулируется с использованием требований безопасности, используемых в криптографии.

Формально обфускатор  $O$  представляет собой компилятор, основанный на эффективных (имеющих полиномиальную сложность) алгоритмах, который получает на вход программу (*circuit*)  $P$  и продуцирует новую программу  $O(P)$ , которая имеет функциональность, аналогичную  $P$ , но при этом является в некотором роде неразборчивой (*unintelligible*) [1]. С этой точки зрения код  $O(P)$  представляет собой черный ящик, для которого нельзя получить никакой иной информации, кроме той, что можно получить на основе итеративного взаимодействия (*oracle access*) с  $P$ .

Такой подход к обфускации просуществовал достаточно долго. На этой стадии "неразборчивость" программного кода определялась скорее интуитивными представлениями о сложности его исследования. Но в 2001 г. группа исследователей доказала [2], что обфускация в виде получения подобного "черного ящика" является недостижимой задачей. Для доказательства этого утверждения они определили класс эффективно вычисляемых программ  $\mathbf{P}$ , представляющих однонаправленные хеш-функции, которые являются необфусцируемыми в связи с тем, что:

— при полученной эффективно вычисляемой программе  $P'$ , имеющей аналогичную функциональность, что и программа  $P \in \mathbf{P}$ , код программы  $P$  может быть эффективно реконструирован за квадратичное время от числа базовых операций (в оригинале *the running time* [2])  $P'$ .

— при этом имея доступ к итеративному взаимодействию (*oracle access*) с  $P \in \mathbf{P}$ , мы не имеем эффективного алгоритма, реконструирующего  $P$  (даже если мы можем отличить отдельные биты на выходе от случайных) с поправкой на пренебрежимо малую величину.

Таким образом, для данного класса программ  $\mathbf{P}$  имеем недостижимость получения "черного ящика"  $O(P)$ , о котором нет никакой информации за исключением той, что можно эффективно получить на основе итеративного взаимодействия с  $P$ .

Для того чтобы сохранить криптографическую проблему обфускации, в работе [2] предложено более слабое, но выполнимое определение обфускатора. Задача обфускатора в данном случае сводилась к тому, что если имеются два различных алгоритма равной функциональности, то мы не можем различить, какой из них был обфусцирован с точностью до пренебрежимо малой функции.

Неразличимый обфускатор (*indistinguishable obfuscator*) определяется также как программный обфускатор за исключением того, что свойства "черного ящика" на выходе определены следующим образом.

*Неразличимость (Indistinguishability).* Для любого вероятностно-полиномиального алгоритма  $A$  существует пренебрежимо малая функция  $\varepsilon$  такая, что для любых программ  $C_1$  и  $C_2$  равной функциональности и равного размера  $k$  выполняется условие:

$$|\Pr(A(O(C_1))) - \Pr(A(O(C_2)))| \leq \varepsilon(k).$$

\* Работа поддержана грантом РФФИ, проект 16-29-09456 конкурс офи\_м.

То есть ни один полиномиальный алгоритм не может различить, что было на входе у неразличимого обфускатора с точностью до пренебрежимо малой величины. Постановка такой задачи аналогична постановке задачи неразличимости шифров в классической криптографии [3]. В работе [2] было доказано, что существуют алгоритмы с неполиномиальной сложностью для построения неразличимых обфускаторов, т. е. задача защиты от исследования программного кода в такой постановке является разрешимой, хотя долгое время не было известно существуют ли для этого эффективные алгоритмы.

В 2013 г. в работе [4] впервые было представлено решение задачи построения обфускатора, неразличимого для любых алгоритмов и имеющего полиномиальную сложность в зависимости от времени выполнения программы. Данное решение было названо важнейшим открытием в области криптографии за последнее десятилетие. В частности, оно позволяло решать неразрешимую ранее проблему функциональной криптографии [5]. Некоторые из таких решений приведены в работах [6] и [7]. Идея обфускатора основана на использовании сложных математических задач, таких как логарифмирование в конечном поле. Техническое решение было названо "многолинейными пазлами" (*Multilinear Jigsaw Puzzles*). Эффективность его против алгебраических атак показана в работе [8].

Специфика многолинейных пазлов заключалась в том, что размер обфусцированной программы быстро растет от времени выполнения программы (то есть от числа базовых операторов). Таким образом, даже простая программа превращается в огромный непрактичный код. Впоследствии появились и некоторые решения этой проблемы, как например обфускатор, построенный для операторов машины Тьюринга, код на выходе которого пропорционален числу команд входа [9]. На сегодняшний день неразличимая обфускация кода до сих пор остается больше теоретической задачей, редко реализуемой на практике. В данной работе будет предложена иная постановка задачи обфускации, которая позволяет получить неразличимость на более простых реализациях.

### Постановка задачи обфускации с использованием ключа

Результат, который мы планируем получить, не укладывается в формулировку задачи как классической [1], так и неразличимой обфускации [2]. Мы будем предполагать, что обфускатор  $ikO(P, k)$  получает на вход помимо программы  $P$  еще и некоторый ключ  $k$ . Далее мы будем называть его *ключевой обфускатор*. Ключ является обязательным параметром приложения на выходе. Если мы знаем ключ обфускатора, то задача сводится либо к обыч-

ному обфускатору  $O(P)$ , либо к неразличимому обфускатору  $ikO(P)$ .

Обфусцированное приложение  $ikO(P, k)$  является неразличимым относительно исходного алгоритма  $P$ . Условие неразличимости можно сформулировать следующим образом.

Возьмем два эффективно вычисляемых алгоритма  $P_0$  и  $P_1$  и атакующий алгоритм (алгоритм различения)  $A$ . Введем определение вероятностного эксперимента  $Priv_{A, ikO}(n)$ . Для любого случайно выбранного  $i \in \{0, 1\}$  и случайного  $k$  побитовой длины  $n$  генерируем программу  $ikO(P_i, k)$ . Программу  $ikO(P_i, k)$  подаем на вход  $A$ . Атакующий  $A$  является вероятностно-полиномиальным алгоритмом. На выходе он генерирует значение  $i' \in \{0, 1\}$ . Если  $i = i'$ , то эксперимент оценивается как удачный,  $Priv_{A, ikO}(n) = 1$ , в противном случае  $Priv_{A, ikO}(n) = 0$ .

Тогда условие неразличимости для ключевого обфускатора можно сформулировать следующим образом.

*Неразличимость (Indistinguishability)*. Для любых двух эффективных алгоритмов  $P_0$  и  $P_1$  равной длины, но *различной* функциональности и любого вероятностно-полиномиального алгоритма  $A$  выполняется

$$\Pr(Priv_{A, ikO}(n)) \leq \frac{1}{2} + \varepsilon(n),$$

где  $\varepsilon(n)$  — пренебрежимо малая величина (*negligible function*).

Сделаем небольшие пояснения. Мы изменили постановку задачи таким образом, что после обфускации остается неясным, какой алгоритм выполняет программа. Допустим, до обфускации была некоторая программа  $P$ . После применения ключевого обфускатора мы получили программу  $P' \leftarrow ikO(P, k)$ . При этом если исходная программа  $P$  имела множество входных параметров  $\mathbf{par}$ , т. е.  $P(\mathbf{par})$ , то программа на выходе  $P'$  будет иметь еще обязательным параметром ключ  $k$ , т. е.  $P'(k, \mathbf{par})$ . При различных ключах  $k$  на входе мы будем получать различный алгоритм —  $P'_k(\mathbf{par})$ . Обозначим как  $\mathbf{P}_k$  множество всех алгоритмов  $P'_k(\mathbf{par})$  при всех возможных значениях ключа  $k$ . Если обфускатор эффективно использует ключ, то

$$|\mathbf{P}_k| \approx |\mathbf{k}| = 2^{|\mathbf{k}|},$$

где двойная скобка обозначает побитовую длину.

Легко доказать, что если пространство алгоритмов  $\mathbf{P}_k$  значительно меньше пространства ключей  $2^{|\mathbf{k}|}$ , то условие неразличимости не выполняется. В результате, если ключ имеет длину 128 бит, то на выходе обфускатора  $ikO(P, k)$  мы получим программу, выполняющую примерно  $2^{128}$  алгоритмов при различном ключе на входе.

Из условий неразличимости можно сделать еще один важный вывод:

**Следствие 1.** Для любого вероятностно-полиномиального алгоритма (атакующего)  $A$  задача опре-

деления, выполняет ли программа  $P' \leftarrow ikO(\dots, k)$  определенный алгоритм  $P$ , не может быть решена на основе эффективных алгоритмов. То есть сложность таких алгоритмов проверки будет расти неполиномиально от  $\|k\|$ .

Дальнейшее изложение материала будет строиться вокруг двух важных вопросов: "Существует ли ключевой обфускатор?" и "Существуют ли эффективные алгоритмы получения ключевого обфускатора?".

Если ключевой обфускатор существует, и мы находим эффективные алгоритмы для его получения, то, тем самым, мы можем решить множество прикладных задач.

- Хранение кода приложения на недоверенных ресурсах. Поскольку непосредственный анализ кода обфусцированной программы не дает нам информации относительно тех алгоритмов, которые он может выполнять, мы можем не защищать его от анализа. Это принципиально отличает ключевой обфускатор от классического, так как в последнем случае функциональность обфусцированной программы всегда известна.

- Нет необходимости в шифровании кода. Шифрование кода иногда используется для защиты его от атак инъекций. Однако для того чтобы запустить код, нам надо его дешифровать для получения исполняемых файлов, которые, в свою очередь, могут быть скомпрометированы. Программа на выходе ключевого обфускатора уже является исполняемой, нет необходимости в ее дешифровке. Анализ ее работы возможен во время исполнения, но это представляется уже более сложной задачей, что будет описано ниже.

- Программа может быть разделена на множество частей, часть из которых будут избыточны (не задействованы в исходном алгоритме, либо задействованы в других обфусцированных алгоритмах). Но при этом для атакующего путем анализа кода программы нет никакой возможности определить, какие ее части являются задействованными в выполнении тех или иных алгоритмов. Если мы можем физически разделить такие составляющие приложения, то значительно усложним для злоумышленника задачу взлома.

Далее мы ответим на вопрос о существовании ключевого обфускатора.

### Неэффективные алгоритмы для получения ключевого обфускатора

Для доказательства существования ключевого обфускатора, рассмотрим следующую функцию:

```
Func(k, par)
x = hash(k)
if (x == (x0)): P0(par)
if (x == (x1)): P1(par)
...
if (x == (x2n-1)): P2n-1(par)
end;
```

Здесь функция  $hash(\cdot)$  обозначает однонаправленную хеш-функцию.

**Теорема 1.** Обфускатор  $ikO(P(\mathbf{par}), k) \rightarrow Func(k, \mathbf{par})$  является ключевым обфускатором.

Допустим, на вход обфускатора мы подали одну из двух программ  $P_{t_0}, P_{t_1} \in \{P_0, P_1, \dots, P_{2^n-1}\}$ .

Чтобы определить, какая из двух программ  $P_t$  (где  $t \in \{t_0, t_1\}$ ) была обфусцирована, необходимо найти и проверить существование соответствующего ключа, для которого выполняется  $hash(k) = x_t$ . Поскольку функция  $hash$  является однонаправленной хеш-функцией, то в соответствии с определением [3] для любого вероятностно-полиномиального алгоритма инверсии  $A$  существует пренебрежимо малая величина  $negl(n)$  такая что:

$$\Pr[A(f(x)) \in f^{-1}(f(x))] \leqslant negl(n).$$

В результате успешность эксперимента по различению программ на входе обфускатора будет оцениваться величиной:

$$\Pr(Priv_{A, ikO}(n)) \leqslant \frac{1}{2} + \Pr[A(f(x)) \in f^{-1}(f(x))]q(n),$$

где  $q(n)$  — число выполнений алгоритма. В соответствии со свойством пренебрежимо малых функций  $negl(n) \cdot poly(n) = negl(n)$ . Таким образом, получаем вероятность успешного эксперимента как

$$\Pr(Priv_{A, ikO}(n)) \leqslant \frac{1}{2} + negl(n),$$

что удовлетворяет условию неразличимости.

Приведенный пример доказывает существование ключевых обфускаторов, однако программный код на выходе обфускатора растет неполиномиально от длины ключа. Следовательно, и сам алгоритм является неэффективным.

Не претендуя на всеобщность выводов, предположим, что программа на выходе ключевого обфускатора может не расти экспоненциально от длины ключа, если мы используем функцию рекурсивно.

Рассмотрим следующий пример.

```
Func_rb(k, par)
if (hash(k) == x)
    result = par
else
    func_rb(k | c(hash(k) mod t), f(hash(k) mod t)(par))
end;
```

Здесь используется множество коэффициентов  $\{c_0, c_1, \dots, c_{t-1}\}$  и множество функций  $\{f_0, f_1, \dots, f_{t-1}\}$ . Таким образом, размер программы на выходе обфускатора определяется опциональной величиной  $t$ . Операция "||" обозначает конкатенацию двух строк.

В качестве примера: если коэффициент  $t = 4$ , т. е. функция рекурсивно будет выполняться 64 цикла, то мы получим  $2^{128}$  различных алгоритмов функционирования с различными значениями на выходе.

**Теорема 2.** Обфускатор  $ikO(P(\mathbf{par}), k) \rightarrow Func\_rb(k, \mathbf{par})$  является ключевым обфускатором.

Для доказательства этого утверждения обратим внимание на тот факт, что выполнение рекурсивной функции может быть либо бесконечным, либо заканчиваться на условии ( $hash(k) = x$ ). Таким образом, мы должны определить алгоритм, который мог бы найти  $k$ , при котором программа сможет закончить свою работу. Для этого надо найти решения уравнения

$$k \| c_{i_1} \| c_{i_2} \| \dots \| c_{i_n} = hash^{-1}(x),$$

$$\begin{cases} k \| c_{i_1} \| c_{i_2} \| \dots \| c_{i_n} = hash^{-1}(x) \\ hash(k) \bmod t = i_1 \\ hash(k \| c_{i_1}) \bmod t = i_2 \\ hash(k \| c_{i_1} \| c_{i_2} \| \dots \| c_{i_{n-1}}) \bmod t = i_n. \end{cases}$$

Здесь  $i_1, \dots, i_n \in \{0, \dots, t-1\}$ . В случае если необходимо проверить выполнение конкретного алгоритма  $P(\mathbf{par})$ , то необходимо решить систему уравнений с дополнительным условием:

$$P(\mathbf{par}) = f_{i_n}(f_{i_{n-1}}(\dots f_{i_1}(\mathbf{par}) \dots)).$$

Очевидно, что обязательной составляющей решения любой из этих задач является вычисление значения  $k \| c_{i_1} \| c_{i_2} \| \dots \| c_{i_n} = hash^{-1}(x)$ , что как и ранее предполагает поиск полиномиального алгоритма для инверсного вычисления хеш-функции. Как уже было ранее определено, для любых вероятностно полиномиальных алгоритмов, вероятность его вычисления будет меньше либо равна пренебрежимо малой величине  $negl(n)$ . Таким образом, вероятность осуществления успешного эксперимента  $Priv_{A, ikO}(n)$  будет

$$\Pr(Priv_{A, ikO}(n)) \leq \frac{1}{2} + negl(n),$$

что удовлетворяет условию неразличимости.

Рекурсивная функция  $Func\_rb(k, \mathbf{par})$  удовлетворяет нескольким важным условиям:

- является эффективно вычислимой;
- имеет линейный размер, пропорциональный переменной  $t$ ;
- может заключать в себе любой программный алгоритм, представленный в виде  $P(\mathbf{par}) = f_{i_n}(f_{i_{n-1}}(\dots f_{i_1}(\mathbf{par}) \dots))$ ;
- на основе анализа кода функции представляется невозможным определить (не существует полиномиальных алгоритмов такого определения), останавливаются ли вычисления при каком-либо значении ключа;
- на основе анализа кода функции представляется невозможным определить (не существует полиномиальных алгоритмов такого определения), может ли программа выполнить алгоритм  $P(\mathbf{par})$  при каком-либо значении ключа.

Эти результаты представляются интересными с точки зрения проблем обфускации, обозначенных

в начале статьи. Следовательно, важным является ответ на вопрос: можем ли мы получить эффективный алгоритм работы обфускатора  $ikO(P(\mathbf{par}), k) \rightarrow Func\_rb(k, \mathbf{par})$ , работающий за полиномиальное время от длины ключа  $n$ .

### Эффективные алгоритмы для получения ключевого обфускатора

Рассмотрим возможные пути получения ключевого обфускатора  $ikO(P(\mathbf{par}), k) \rightarrow Func\_rb(k, \mathbf{par})$ . Допустим, мы определили несколько базовых функций  $\{f_0 \dots f_{t-1}\}$ , использующихся рекурсивной функцией  $func\_rb$  для воспроизводства обфусцируемого алгоритма  $P(\mathbf{par}) = f_{i_n}(f_{i_{n-1}}(\dots f_{i_1}(\mathbf{par}) \dots))$ . Число вложений (рекурсивных вызовов функцией себя) в данном случае должно быть таким, чтобы пространство функций было не меньше пространства ключей. Обозначим число вложений как  $h$ . Тогда минимальное значение  $h = 2n/t$ .

**Теорема 3.** Сложность алгоритма  $ikO(P(\mathbf{par}), k) \rightarrow Func\_rb(k, \mathbf{par})$  при фиксированном значении  $t$  и заданном разложении функции  $P(\mathbf{par}) = f_{i_n}(f_{i_{n-1}}(\dots f_{i_1}(\mathbf{par}) \dots))$ , где  $i_1, \dots, i_n \in \{0, \dots, t-1\}$ , растет неполиномиально от длины ключа  $n$ .

Для доказательства данного утверждения приведем логику работы алгоритма. Допустим в разложении  $P(\mathbf{par}) = f_{i_n}(f_{i_{n-1}}(\dots f_{i_1}(\mathbf{par}) \dots))$  первой на выполнение стоит функция  $f_{i_1}(\mathbf{par})$ . В результате мы должны задать такой исходный ключ  $k$ , чтобы выполнялось

$$hash(k) \bmod t = i_1.$$

Поскольку хеш-функция является однонаправленной, то для вычисления инверсного значения нам необходимо перебирать параметры функции. Для каждой попытки перебора

$$\Pr[A(hash(k)) \in hash^{-1}(hash(k))] = \frac{1}{t},$$

значит в среднем на выполнение этой операции нам необходимо  $t$  циклов работы программы. Аналогично, если далее на выполнение стоит функция  $f_{i_2}(\mathbf{par})$ , вычисляем

$$hash(k \| c_{i_1}) \bmod t = i_2.$$

Здесь также получаем среднее число циклов на выполнение (подбор коэффициента  $c_{i_1}$ ) —  $t$ . Поскольку операции выполняются последовательно, то суммарная сложность этих двух операций  $O(\cdot) = 2t$ . Сложность алгоритма резко растет, если мы встречаем дважды одну и ту же базовую функцию  $P(\mathbf{par}) = f_{i_n}(f_{i_{n-1}}(\dots f_{i_1}(\mathbf{par}) \dots))$ , и, соответственно, дважды встречаем один и тот же коэффициент в исходной строке для хеш-функции  $c_{i_e} = c_{i_l}$ . Здесь уже на каждую операцию перебора надо перебирать все ранее полученные значения. Число таких дублирующих коэффициентов будет не менее, чем  $(h - t)$ .

В результате сложность всего алгоритма можно оценить по формуле

$$\begin{aligned} O[ikO(P(\mathbf{par}), k) \rightarrow \text{func\_rb}(k, \mathbf{par})] &= \\ &= \underbrace{(t + t + \dots + t)}_t \cdot \underbrace{t \cdot t \cdot \dots \cdot t}_{h-t} = \\ &= t^2 + (h-t) = t^{\frac{2n}{t} - t + 2} \end{aligned}$$

Таким образом, мы определили, что сложность алгоритма обфускации для фиксированного  $t$  будет расти неполиномиально от  $n$ .

В случае, когда имеется всего четыре базовых функции и 128-битный ключ, число итераций алгоритма будет не менее чем  $4^{62}$ , что нереализуемо на современных вычислительных ресурсах.

Однако это совсем не значит, что мы не можем использовать обфускатор с таким алгоритмом. Для сокращения операций можно увеличивать  $t$  и сокращать число итераций алгоритма для фиксированного  $n$ . Например, возьмем 16 базовых операций для 128-битного ключа. В этом случае среднее число проходов алгоритма обфускатора будет всего 256, т. е. алгоритм при таких параметрах является достаточно простым.

Из этого следует важный вывод.

**Следствие 2.** Алгоритм  $ikO(P(\mathbf{par}), k) \rightarrow \text{Func\_rb}(k, \mathbf{par})$  для заданного разложения функции  $P(\mathbf{par}) = f_{i_n}(f_{i_{n-1}}(\dots f_{i_1}(\mathbf{par}) \dots))$ , где  $i_1, \dots, i_n \in \{0, \dots, t-1\}$  будет эффективен при выборе коэффициента  $t$  в зависимости от длины ключа  $n$ .

Таким образом, с учетом подбора коэффициента  $t$  нам удалось получить эффективный алгоритм ключевого обфускатора.

Однако увеличение числа  $t$  базовых функций рекурсивной функции  $\text{func\_rb}(k, \mathbf{par})$  не всегда удобно. Актуальной задачей остается получение эффективных алгоритмов ключевого обфускатора с маленькими значениями  $t$  на больших длинах ключа. Это может оказаться возможным, если мы переформулируем исходную постановку задачи.

Теперь мы не определяем формат представления алгоритма  $P(\mathbf{par}) = f_{i_n}(f_{i_{n-1}}(\dots f_{i_1}(\mathbf{par}) \dots))$ . В новой постановке задачи определяем набор коэффициентов  $\{c_0, c_1, \dots, c_{t-1}\}$  и исходный ключ  $k$ . В результате, вычисляя хэши, жестко определяем последовательность выполнения базовых функций  $\{f_0 \dots f_{t-1}\}$ . Таким образом, зная последовательность рекурсивного выполнения базовых функций, необходимо найти такие базовые функции  $\{f_0 \dots f_{t-1}\}$ , чтобы в результате выполнилось равенство  $P(\mathbf{par}) = f_{i_n}(f_{i_{n-1}}(\dots f_{i_1}(\mathbf{par}) \dots))$ .

Для базовых команд процессора всегда есть алгоритм простого получения развернутой последовательности  $f_{i_n}(f_{i_{n-1}}(\dots f_{i_1}(\mathbf{par}) \dots))$ . Однако работа обфускатора на уровне базовых команд процессора приведет к значительному увеличению и замедленному выполнению программного кода. Для нетривиальных функций (не являющихся базовыми

командами процессора) такая обфускация возможна, но вопрос универсального алгоритма этой обфускации остается открытым.

Допустим, исходная программа выполняет функцию  $P(x) = x^2$ . Рекурсивная цепочка базовых функций  $\text{Func\_rb}(k, x)$  имеет вид  $f_1(f_2(f_2(f_1(x))))$ . Для  $P(x) = f_1(f_2(f_2(f_1(x))))$  мы можем определить

$f_1(x) = x^{1/2}$  и  $f_2(x) = x^{8^{1/2}}$ . Такая рекурсия базовых функций выполнит функцию, аналогичную  $P(x)$ .

Для любых базовых команд процессора можно определить базовые функции, которые при заданной комбинации выполнят исходную команду. Сделать это можно с учетом наличия для любой операции обратной операции, которая в рекурсии могла бы привести к нужному результату. Однако вводить рекурсивную функцию для каждой базовой команды слишком затратно и приведет к значительному росту исполняемого кода приложения. Для этого нам необходимо определить, важно ли, чтобы программа оставалась неразличимой, если злоумышленник может анализировать ее в процессе выполнения, т. е. видит все ключи на входе функций. В результате программа может оставаться неразличимой в значении определения, введенного в работе [2], либо оставаться только неразборчивой (*unintelligible*). В последнем случае мы можем не вводить рекурсивную функцию для каждой базовой команды, а использовать совокупности команд.

Для совокупности команд или более сложных алгоритмов поиск базовых функций уже не является столь простой задачей. Частично эта проблема решена в обфускаторе RKB (*recursive key blur*).

### Реализация ключевого обфускатора на примере обфускатора RKB

Обфускатор RKB реализует алгоритм ключевого обфускатора, сопоставляя команды высокоуровневого языка программирования с рекурсивными функциями типа  $\text{func\_rb}$ .

Рассмотрим пример простой программы

```
For (i = 0; i < 10; i++)
    Print("hello world")
end;
```

Данная программа выполняет цикл и выводит строку "hello world" 10 раз.

Аналогичная ей рекурсивная функция будет записана в виде.

```
func_rb(k, i)
    if (hash(k) = x)
        result = i
    else
        print("hello world")
        func_rb(k || c, i = i + 1)
end;
```

Здесь значения  $k$ ,  $c$  и  $x$  будут подобраны таким образом, чтобы рекурсивная функция выполнялась



ровно 10 раз. В результате получим функционал, идентичный программе выше.

Аналогичные паттерны построения рекурсивных функций можно определить для всех высокоуровневых команд.

### Заключение

В данной работе удалось сформулировать требования к ключевому обфускатору, доказать его существование и найти эффективные алгоритмы его работы. Данные алгоритмы легли в основу разработки обфускатора RKB, выполняющего роль ключевого обфускатора на уровне высокоуровневого языка программирования.

С прикладной точки зрения использование ключевых обфускаторов позволяет решить такие проблемы, как хранение кода приложения на недоверенных ресурсах и разделение логики программы с невозможностью восстановления без знания ключевой информации. Поскольку непосредственный анализ кода обфусцированной программы не дает информации относительно тех алгоритмов, которые он может выполнять, мы можем не защищать его от анализа потенциальным злоумышленником.

Важным направлением дальнейшей разработки является получение алгоритмов для приведения к

рекурсивным функциям любого программного кода, а также применение одноразовых ключей в рекурсивных функциях для того, чтобы затруднить их анализ в процессе выполнения.

### Список литературы

1. **Hada S.** Zero-knowledge and code obfuscation // *Advances in Cryptology — ASIACRYPT*, 2000. P. 443–457.
2. **Barak B., Goldreich O., Impagliazzo R.** On the (Im)possibility of obfuscating programs // *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2001. N. 2139. P. 1–18.
3. **Katz J., Lindell Y.** *Introduction to Modern Cryptography*. CRC PRESS, 2007. 512 p.
4. **Garg S., Raykova M., Sahai A.** Candidate indistinguishability obfuscation and functional encryption for all circuits // *Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*. 2013. P. 23–41.
5. **Alwen J., Barbosa M., Farshim P.** On the Relationship between Functional Encryption, Obfuscation, and Fully Homomorphic Encryption // *Lecture Notes in Computer Science, Cryptography and Coding*. 2014. N. 8308. P. 65–84.
6. **Sahai A., Waters B.** How to use indistinguishability obfuscation: deniable encryption, and more // *Proc. of the 46th Annual ACM Symposium on Theory of Computing*. 2014. P. 475–484.
7. **Garg S., Gentry C., Halevi S.** Candidate Multilinear Maps from Ideal Lattices // *Lecture Notes in Computer Science (EUROCRYPT 2013)*. 2013. N. 7881. P. 1–17.
8. **Barak B., Garg S., Kalai Y.** Protecting Obfuscation Against Algebraic Attacks // *EUROCRYPT 2014*. 2014. P. 221–238.
9. **Koppula V., Bishop A., Waters B.** Indistinguishability Obfuscation for Turing Machines with Unbounded Memory // *Proc. of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*. 2015. P. 419–428.

M. A. Styugin, PhD, Researcher, e-mail: styugin@gmail.com,  
Siberian State Aerospace University (SibSAU), Krasnoyarsk

## Indistinguishable Executable Code Generation Method

*The classical objective of obfuscation considers indistinguishability of the obtained code in relation to original programs of equal functionality. The present paper reviews another objective of obfuscation, when indistinguishability is considered relatively to programs of different functionality. In this case, an obfuscator is provided with a key at the input. It is nearly impossible to discover the program's functionality without having the key, therefore that obfuscator type is named a key obfuscator. The paper proves existence of a key obfuscator and existence of efficient algorithms for its operation demonstrated by recursive functions. The practical relevance of the result obtained by the research is the possibility to store an executable code in an untrusted computational environment and to obstruct injection attacks and distributed computing analysis.*

**Keywords:** obfuscation, indistinguishability, untrusted computations, key-based obfuscation, computational complexity, polynomial time algorithm

### References

1. **Hada S.** Zero-knowledge and code obfuscation, *Advances in Cryptology — ASIACRYPT*, 2000, pp. 443–457.
2. **Barak B., Goldreich O., Impagliazzo R.** On the (Im)possibility of obfuscating programs, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2001, no. 2139, pp. 1–18.
3. **Katz J., Lindell Y.** *Introduction to Modern Cryptography*, CRC PRESS, 2007, 512 p.
4. **Garg S., Raykova M., Sahai A.** Candidate indistinguishability obfuscation and functional encryption for all circuits, *Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, 2013, pp. 23–41.
5. **Alwen J., Barbosa M., Farshim P.** On the Relationship between Functional Encryption, Obfuscation, and Fully Homomorphic

Encryption, *Lecture Notes in Computer Science, Cryptography and Coding*, 2014, no. 8308, pp. 65–84.

6. **Sahai A., Waters B.** How to use indistinguishability obfuscation: deniable encryption, and more, *Proc. of the 46th Annual ACM Symposium on Theory of Computing*, 2014, pp. 475–484.
7. **Garg S., Gentry C., Halevi S.** Candidate Multilinear Maps from Ideal Lattices, *Lecture Notes in Computer Science (EUROCRYPT 2013)*, 2013, no. 7881, pp. 1–17.
8. **Barak B., Garg S., Kalai Y.** Protecting Obfuscation Against Algebraic Attacks, *EUROCRYPT 2014*, 2014, pp. 221–238.
9. **Koppula V., Bishop A., Waters B.** Indistinguishability Obfuscation for Turing Machines with Unbounded Memory, *Proc. of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, 2015, pp. 419–428.

К. А. Щеглов, аспирант, А. Ю. Щеглов, д-р техн. наук, проф., e-mail: info@npp-itb.sbp.ru,  
Университет ИТМО, Санкт-Петербург, Россия

## Эксплуатационная безопасность. Моделирование реализации угроз атак потенциальным нарушителем

*Исследованы вопросы моделирования эксплуатационных параметров и характеристик угрозы атаки, отличающихся от надежностных моделированием не процесса возникновения и устранения в информационной системе реальных угроз атак, а процесса их реализации потенциальным нарушителем. Предложен и обоснован подход к моделированию угрозы атаки марковскими моделями с дискретными состояниями и непрерывным временем с потерями, где поток случайных событий потерь составляют возникающие в системе реальные угрозы уязвимостей реализации, не используемых по различным причинам потенциальным нарушителем для осуществления атаки. Рассмотрены особенности моделирования угрозы безопасности информационной системы и угрозы безопасности обрабатываемой в ней информации, определяемые целями реализации реальных угроз атак потенциальным нарушителем — нарушение конфиденциальности, целостности и доступности информации.*

**Ключевые слова:** угроза атаки, угроза уязвимостей, математическое моделирование, марковская модель, модель с потерями входных случайных событий, цель атаки, угроза безопасности информационной системы, угроза безопасности информации

### Введение

В работах [1, 2] изложен подход к моделированию угрозы атаки марковской моделью с дискретными состояниями и непрерывным временем с последующим построением для нее модели вероятного разрежения входных потоков случайных событий, основанный на введенной в [3] интерпретации угрозы атаки схемой параллельного резервирования создающих ее угроз уязвимостей. В результате реализации данного подхода к моделированию могут быть рассчитаны надежностные параметры и характеристики угрозы атаки, поскольку при этом моделируется не атака как процесс последовательного деструктивного воздействия нарушителем на систему [4–7], а именно угроза атаки как процесс возникновения и устранения в системе отказов информационной безопасности — реальных угроз атак (все создающие угрозу атаки уязвимости присутствуют в системе), что позволило ввести в [8] понятие надежности информационной безопасности. В [8] сформулированы требования к корректности построения для решаемых задач марковских моделей с дискретными состояниями и непрерывным временем, предложен подход к объединению состояний на графе системы состояний случайного процесса, существенно расширяющий возможности моделирования в части определения различных надежностных характеристик угроз атак.

Однако при проектировании защищенных информационных систем необходимо не только оценивать параметры и характеристики возникновения и устранения в системе реальных угроз атак, но и готовность потенциального нарушителя реализовать подобную реальную угрозу — осуществить атаку, т. е. необходимо строить модели реализации угрозы атаки.

В данной работе речь пойдет о моделировании эксплуатационных параметров и характеристик угрозы атаки — рассмотрим вопросы моделирования реализации угроз атак, отличающихся от надежностных возможностью учета заинтересованности и готовности потенциального нарушителя реализовать реальную угрозу атаки, а также учета возможных целей реализации атак потенциальным нарушителем.

Под эксплуатационной информационной безопасностью будем понимать свойство информационной системы сохранять во времени в установленных пределах значения всех характеристик безопасности информации, определяющих способность выполнять информационной системой требуемые функции обработки информации в безопасном режиме, в условиях реализации атак на информационную систему потенциальным нарушителем в процессе ее эксплуатации.

## Подход к моделированию эксплуатационных параметров и характеристик угрозы атаки

Поскольку у события реализации возникающей в системе реальной угрозы уязвимостей<sup>1</sup> два исхода с соответствующими вероятностями — атака реализована либо нет, вместо каждого одного состояния графа системы состояний случайного процесса, отражающего событие возникновения в системе соответствующей реальной угрозы уязвимостей, в данном случае при моделировании уже требуется использовать два состояния — возникшая реальная угроза уязвимостей реализована потенциальным нарушителем либо нет, при задании же интенсивностей переходов между состояниями требуется учитывать вероятности реализации потенциальным нарушителем соответствующих возникающих реальных угроз уязвимостей.

Подход к моделированию эксплуатационных характеристик безопасности информации, предполагающий соответствующее преобразование графа состояний случайного процесса марковской модели угрозы атаки в марковскую модель реализации угрозы атаки, проиллюстрирован на рис. 1 (отображаем на модели только интересующий нас входной поток случайных событий).

На рис. 1, а приведен фрагмент графа состояний случайного процесса, моделирующего надежные параметры и характеристики безопасности информации. Переход из состояния  $S_i$  в состояние  $S_j$  связан с возникновением в системе  $j$ -й реальной угрозы (типа угрозы [8]) уязвимостей, интенсивность возникновения которой в системе  $\lambda_j$ .

Переход к графу состояний случайного процесса, моделирующего эксплуатационные характеристики безопасности информации, состоит во включении в него вместо состояния  $S_j$  двух состояний —  $S_{j1}$  и  $S_{j0}$ , соответствующим образом интерпретируемых — реализована угроза уязвимостей реализации либо нет (видим, что здесь иная интерпретация состояний на графе — не возникновение/устранение угрозы уязвимостей, а реализация либо нет успешной атаки на возникшую (реальную) угрозу уязвимостей).

С использованием вероятности  $P_j$  между этими состояниями разрежается входной в состояние  $S_j$  поток случайных событий, имеющий интенсивность  $\lambda_j$ . Физический смысл вероятностного разрежения входного потока случайных событий в данном случае следующий: только доля входного потока  $P_j \lambda_j$  возникающих реальных угроз уязвимостей будет реализована потенциальным нарушителем в целях осуществления атак на информационную систему.

Обоснуем корректность данного подхода к моделированию. При этом вероятность реализации

<sup>1</sup> Под угрозой уязвимостей понимаем угрозу возникновения (присутствия) свойства информационной системы, обуславливающего возможность реализации на нее атаки потенциальным нарушителем.

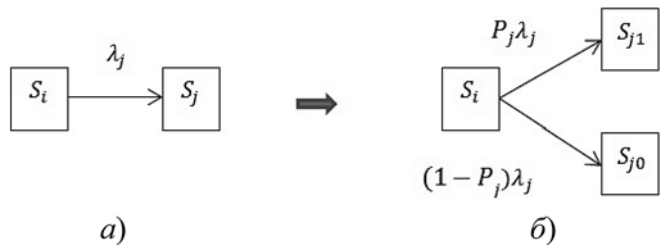


Рис. 1. Иллюстрация подхода к моделированию эксплуатационных характеристик безопасности информации:

а — фрагмент графа для определения надежных характеристик безопасности; б — преобразованный фрагмент графа для определения эксплуатационных характеристик безопасности

потенциальным нарушителем реальной угрозы уязвимостей  $P_{a,y}$  может интерпретироваться как эксплуатационная характеристика опасности потенциального нарушителя для информационной системы.

Изменение характеристики  $P_{a,y}$  во времени, в предположении, что реальные угрозы уязвимостей возникают в системе в среднем через некоторые интервалы времени  $T_{0y}$  может быть описано геометрическим распределением (естественно, предполагаем, что реальные угрозы уязвимостей взаимно не зависимы, поскольку каждая из них уникальна, также предполагаем, что значение  $P_{a,y}$  неизменно во времени) [9] (см. таблицу).

Изменение характеристики  $P_{a,y}$  во времени

$t$	$T_{0y}$	$2T_{0y}$	$3T_{0y}$	...	$kT_{0y}$	...
$P_{a,y}$	$P_{a,y}$	$P_{a,y}(1 - P_{a,y})$	$P_{a,y}(1 - P_{a,y})^2$	...	$P_{a,y}(1 - P_{a,y})^{k-1}$	...

Для геометрического распределения может быть определено соответствующее математическое ожидание — среднее число событий до первого успеха, в нашем случае — среднее число возникающих в системе реальных угроз уязвимостей до осуществления успешной атаки на угрозу уязвимостей  $M_{P_{a,y}}$ :

$$M_{P_{a,y}} = 1/P_{a,y}$$

С учетом этого интенсивность входного потока случайных событий, приводящего к реализации реальной угрозы уязвимостей, обозначим  $\lambda_j^*$ :

$$\lambda_j^* = P_{a,y} \lambda_j = \lambda_j / M_{P_{a,y}}$$

Таким образом, при данном способе вероятностного разрежения входного потока случайных событий в состояние, характеризующее реализацию успешной атаки на реальную угрозу уязвимостей, поступает входной поток только тех событий, которые предполагают реализацию успешной атаки. При вероятностном разрежении простейшего потока случайных событий разреженный поток также будет простейшим [10].

*Замечание.* В данной работе ввиду ограничений на объем статьи не будем останавливаться на воп-

росе, как определять эксплуатационную характеристику  $P_{а.у}$ . Это отдельный крайне важный вопрос — вопрос построения математической модели потенциального нарушителя, которому будет посвящена следующая работа авторов. Здесь же будем исходить из того, что мы знаем, как определять данную характеристику.

Другое принципиальное отличие подхода к моделированию эксплуатационных характеристик безопасности от надежностных состоит в следующем. Потенциальному нарушителю незачем одновременно осуществлять успешные атаки на несколько одновременно (не одновременно — входные потоки случайных событий ординарны) возникающих (точнее, присутствующих) в системе реальных угроз уязвимостей одного типа, предполагающих единообразное последующее развитие атаки — достаточно реализации успешной атаки на одну реальную угрозу уязвимостей одного типа.

Как следствие, любая вторая, третья и т. д. одновременно возникающие в системе реальные угрозы уязвимостей одного типа не будут реализованы потенциальным нарушителем в рамках реализации соответствующей атаки (для него они будут потерями, поскольку не будут использоваться для реализации атаки), что для модели угрозы реализации атаки можно интерпретировать как потери соответствующих случайных событий. Как следствие, можем сделать вывод о том, что марковская модель реализации угрозы атаки — это модель с потерями.

*Замечание.* Как видим, применительно к модели реализации угрозы атаки имеем принципиально новую интерпретацию "потери": событие будет потеряно не потому, что не может быть обслужено, а потому, что в его использовании (в обслуживании) потенциальным нарушителем нет необходимости.

Сформулируем и обоснуем требование к построению корректной марковской модели с дискретными состояниями и непрерывным временем с потерями входных событий. Как отмечали в работе [8], требованием к корректности марковской модели будет выполнение следующего условия: в корректной модели должен корректно вероятностно

разрезаться между состояниями систем входной поток случайных событий.

Пусть мы имеем некую марковскую модель, представленную на рис. 2, а. Модель с потерями получается в результате преобразования, состоящего во включении для состояний системы соответствующих переходов из состояния в себя же, что интерпретируется как потери входных событий (рис. 2, б).

Проблема построения корректной модели в данном случае обуславливается тем, что включение данных переходов в модель не сказывается на исходной системе линейных уравнений. Это объясняется тем, что включаемые подобным образом переходы не влияют на интенсивности переходов между состояниями.

Данное противоречие разрешается тем, что в системе с потерями в качестве входного потока случайных событий должен подаваться уже вероятностно разреженный поток — поток только тех случайных событий, которые переводят систему из одного состояния в другое, разрежение же входного потока состоит в исключении из него потоков теряемых системой событий (потока событий потерь). Это можно рассматривать в качестве требования к построению корректной марковской модели с потерями.

Для модели с потерями, приведенной на рис. 2, б, исключаемые потоки событий потерь  $\lambda_i^п$  определяются следующим образом:

$$\lambda_1^п (P_{10} + P_{11})\lambda_1;$$

$$\lambda_2^п = (P_{01} + P_{11})\lambda_2.$$

Получаемая подобным образом корректная марковская модель с потерями из модели, приведенной на рис. 2, б, представлена на рис. 3, а, соответствующая ей модель вероятностного разрежения входных потоков случайных событий — на рис. 3, б.

Оценим для марковской модели, построенной подобным образом, корректность разрежения входных потоков случайных событий по аналогии с тем, как мы это делали в работе [8], для чего обратимся к рис. 3, б. Интенсивность первого потока случайных событий, циркулирующего в модели, обозначим  $\lambda_{п1}$ , для данной модели должна определяться следующим образом:

$$\lambda_{п1} = \lambda_1 - \lambda_1^п = (P_{00} + P_{01})\lambda_1,$$

т. е. должны учитываться переходы первого потока событий только из состояний  $S_{00}$  и  $S_{01}$ , что, как видим, и реализуется моделью, приведенной на рис. 3, б.

Таким образом, входные потоки случайных событий требуется при моделировании дважды по различным причинам вероятностно разрезать. Особенностью построения модели с потерями с кор-

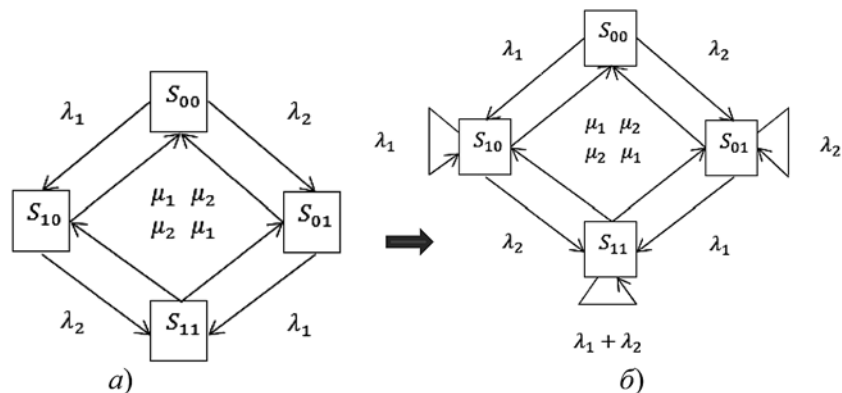


Рис. 2. Иллюстрация построения модели с потерями: а — исходная модель без потерь; б — модель с потерями

ректным вероятностным разрежением входного потока случайных событий является то, что в данном случае вероятностно разрезается (в целях учета потерь событий) уже не исходный поток, поступающий в систему с интенсивностью  $\lambda_j$ , а только часть потока — уже вероятностно разреженного по иным соображениям — для учета вероятности реализации потенциальным нарушителем реальной угрозы уязвимостей, приводящая к потерям случайных событий (рис. 4).

Однако и события — реальные угрозы уязвимостей, которыми не смог воспользоваться потенциальный нарушитель, ввиду своей неготовности к реализации угрозы атаки (состояние на рис. 4, а) также для него могут рассматриваться в качестве потерянных (с точки зрения успешной реализации атаки) событий. Сказанное позволяет исключить из входного потока случайных событий и эти события, которые также, но уже по иным причинам, составят поток потерь. В результате подобного преобразования получаем модель, представленную на рис. 5, а, из которой уже может быть построена корректная марковская модель с потерями (рис. 5, б) и модель вероятностного разрежения входных потоков случайных событий (рис. 5, в).

Покажем, что вероятностное разрежение входных потоков случайных событий для модели, приведенной на рис. 5, а, корректно, т.е. эта марковская модель корректна.

Для модели с потерями, приведенной на рис. 5, б, исключаемый поток случайных событий потерь  $\lambda^\Pi$  определяется следующим образом:

$$\lambda^\Pi = (1 - P)P_0\lambda + P_1\lambda.$$

Исходя из того, что интенсивность вероятностно неразрезанного входного потока

$$\lambda = (P_0 + P_1)\lambda,$$

интенсивность потока случайных событий, циркулирующего в модели  $\lambda_\Pi$ , должна определяться следующим образом:

$$\lambda_\Pi = \lambda - \lambda^\Pi = P_0P\lambda,$$

что, как видим, и реализуется данной моделью (рис. 5, в).

Сформулируем общее правило построения корректной марковской модели с потерями для рассматриваемой задачи моделирования и модели вероятностного разрежения потоков случайных событий.

**Правило.** Если под воздействием входного потока случайных событий, поступающего в систему с интенсивностью  $\lambda$ , система переходит из состояния

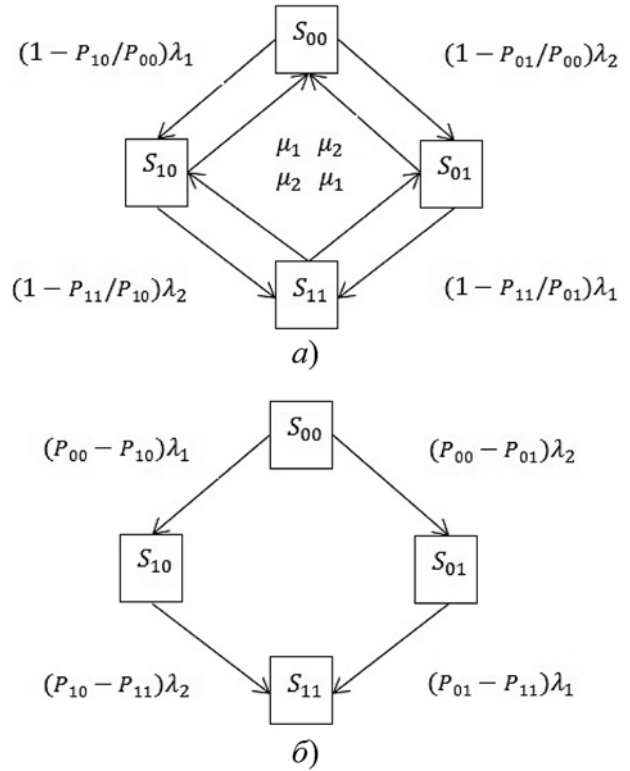


Рис. 3. Размеченные графы системы состояний случайного процесса марковской модели с потерями с исключенным потоком событий потерь: а — корректная марковская модель с потерями; б — модель вероятностного разрежения входных потоков

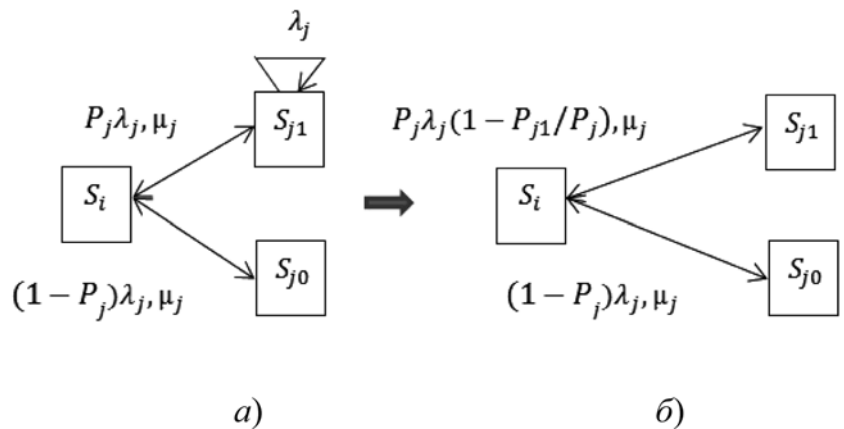


Рис. 4. Иллюстрация построения модели с потерями с корректным вероятностным разрежением входного потока случайных событий: а — фрагмент графа исходной модели с потерями; б — фрагмент графа модели с потерями с корректным вероятностным разрежением входного потока случайных событий

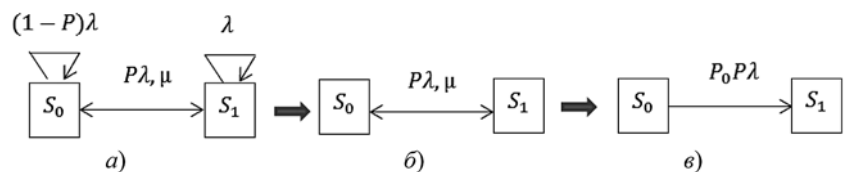


Рис. 5. Иллюстрация построения корректной марковской модели с потерями и модели вероятностного разрежения потока входных случайных событий: а — исходная модель; б — марковская модель с потерями; в — модель вероятностного разрежения потока входных случайных событий

$S_i$  в состояние  $S_j$  с вероятностью  $P$ , то интенсивность входного потока случайных событий в марковской модели с потерями, поступающих из состояния  $S_j$ , определяется, как  $P\lambda$ , а интенсивность потока случайных событий, переводящего систему из состояния  $S_i$  в состояние  $S_j$ , в модели вероятностного разрежения потоков случайных событий определяется как  $P_i P\lambda$ .

Определим эксплуатационные параметры и временные характеристики безопасности для этой модели (рис. 5, 6).

Интенсивность возникновения моделируемого события в системе — успешная реализация атаки потенциальным нарушителем  $\lambda_{pa}$ , определяется следующим образом:

$$\lambda_{pa} = P_0 P\lambda.$$

Интенсивность восстановлений эксплуатационной безопасности информационной системы (при реализации на нее успешных атак) с учетом того, что безопасность системы, нарушаемая с интенсивностью  $\lambda_{pa}$ , восстанавливается за долю времени, равную  $P_1$ , рассчитывается следующим образом:

$$\mu_{pa} = \frac{P_0 P\lambda}{(1 - P_0)}.$$

Соответствующим образом (по аналогии с тем, как это делалось в работе [8] при определении надежностных параметров и характеристик безопасности) определяются временные эксплуатационные характеристики безопасности: среднее время  $T_{mra}$  между отказами эксплуатационной безопасности — успешными реализациями атак, среднее время  $T_{hra}$  наработки на отказ эксплуатационной безопасности — на реализацию успешной атаки, среднее время  $T_{ra}$  восстановления эксплуатационной безопасности — пребывания системы в состоянии реализации на нее успешных атак:

$$T_{mra} = \frac{1}{\lambda_{pa}}; T_{ra} = \frac{1}{\mu_{pa}}; T_{hra} = T_{mra} - T_{ra}.$$

### **Моделирование эксплуатационных параметров и характеристик безопасности информационной системы и безопасности информации**

Атаки потенциальным нарушителем на информационную систему могут осуществляться в целях нарушения конфиденциальности, целостности и доступности обрабатываемой в ней информации.

Сначала рассмотрим угрозы атак, реализуемых в целях нарушения целостности и доступности информации. В этом случае моделируемая информационная система без каких-либо оговорок может рассматриваться как система с отказами и восстановлениями безопасности. Естественно, что угроза подобной атаки связана с соответствующим риском потерь, однако последствия от атаки в дан-

ном случае не фатальны для обрабатываемой в ней информации — после восстановления информационной система может быть вновь введена в эксплуатацию, что позволяет моделировать систему применительно к данным угрозам атак как восстанавливаемую. Восстановление системы состоит в восстановлении целостности информации (например, из резервных копий) или в восстановлении доступности информации, которая может быть нарушена как ее несанкционированным удалением нарушителем, так и выводом из строя системных средств.

Теперь рассмотрим особенности моделирования реализации угроз атак, направленных на нарушение конфиденциальности обрабатываемой информации. Здесь следует рассматривать две в общем случае различные задачи моделирования.

С одной стороны, это моделирование реализации угрозы атаки на информационную систему. С позиций данной задачи моделирования в качестве объекта моделирования имеем собственно информационную систему — это система с отказами и восстановлениями безопасности информации.

Отказ безопасности информации — реализация угрозы атаки, приведшая к нарушению конфиденциальности (хищению) обрабатываемой информации, не приведет к тому, что информационная система будет выведена из эксплуатации. Система будет далее эксплуатироваться. Как следствие, моделирование реализации угрозы атаки применительно к безопасности информационной системы здесь ничем не отличается от рассмотренного ранее.

С другой стороны, применительно к обрабатываемой в информационной системе информации реализация потенциальным нарушителем угрозы атаки может рассматриваться как фатальный отказ безопасности — уже безопасности информации — нарушение конфиденциальности обрабатываемой на момент реализации атаки информации, поскольку в результате подобной атаки защищаемая информация будет похищена. В данном смысле уже нельзя говорить о восстанавливаемости безопасности, так как конфиденциальность похищенных данных не восстанавливается.

Рассмотрим, с учетом сказанного, в чем состоит отличие моделирования эксплуатационных параметров и характеристик безопасности информационной системы и безопасности информации.

Для моделирования эксплуатационных параметров и характеристик безопасности информационной системы реализуется подход, описанный выше. Проиллюстрируем его на примере построения модели реализации угрозы атаки, создаваемой двумя угрозами уязвимостей реализации (рис. 6).

На рис. 6 использованы следующие обозначения. Угрозы уязвимостей, соответственно, первого и второго типов, создающие угрозу атаки, возникают в системе с интенсивностями  $\lambda_1, \lambda_2$ , устраняются

с интенсивностями  $\mu_1, \mu_2$ . Через  $P_1, P_2$  обозначена характеристика вероятности реализации потенциальным нарушителем реальных угроз уязвимостей соответствующих типов,  $S_{00}$  — состояние, характеризующее отсутствие в системе реализации потенциальным нарушителем реальных угроз уязвимостей,  $S_{10}$  — потенциальным нарушителем реализована реальная угроза уязвимостей первого типа,  $S_{01}$  — потенциальным нарушителем реализована реальная угроза уязвимостей второго типа,  $S_{11}$  — потенциальным нарушителем реализованы реальные угрозы уязвимостей обоих типов, создающих угрозу атаки).

По аналогии с тем, как это делалось ранее, для этой модели могут быть соответствующим образом определены эксплуатационные параметры и временные характеристики безопасности — безопасности информационной системы.

Интенсивность возникновения моделируемого события в системе — успешная реализация атаки потенциальным нарушителем — определяется следующим образом:

$$\lambda_{pa} = P_{10}P_2\lambda_2 + P_{01}P_1\lambda_1.$$

Интенсивность восстановлений эксплуатационной безопасности информационной системы (при реализации на нее успешных атак) с учетом того, что безопасность системы, нарушаемая с интенсивностью  $\lambda_{pa}$ , восстанавливается за долю времени, равную  $P_{11}$ , рассчитывается следующим образом:

$$\mu_{pa} = \frac{P_{10}P_2\lambda_2 + P_{01}P_1\lambda_1}{P_{11}}.$$

Соответствующим образом определяются временные эксплуатационные характеристики безопасности.

Моделирование фатального отказа безопасности — отказа безопасности информации, может быть проведено с использованием в марковских моделях так называемых поглощающих состояний — состояний, из которых нет выхода (перехода) [11].

Модель, приведенная на рис. 6, но уже применительно к моделированию безопасности информации, представлена на рис. 7. Включенное в модель поглощающее состояние обозначено через  $S_a$ .

Как видим, моделируемая система — это система с отказами и восстановлениями безопасности информационной системы в отношении не реализуемых потенциальным нарушителем угроз уязвимостей реализации (угроз атак) и одновременно

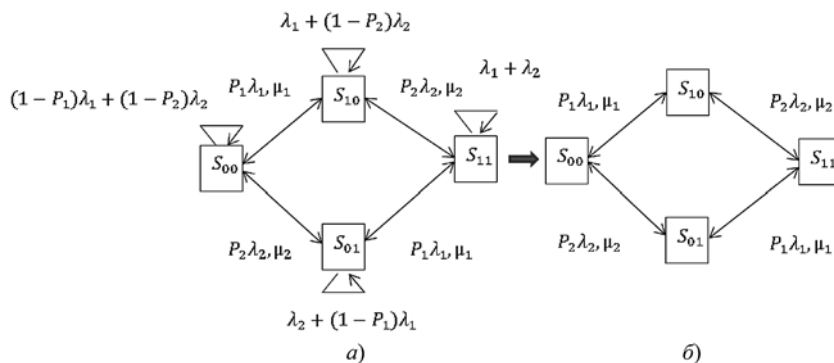


Рис. 6. Размеченные графы системы состояний случайного процесса для модели безопасности информационной системы:

*a* — исходная модель; *б* — марковская модель с потерями

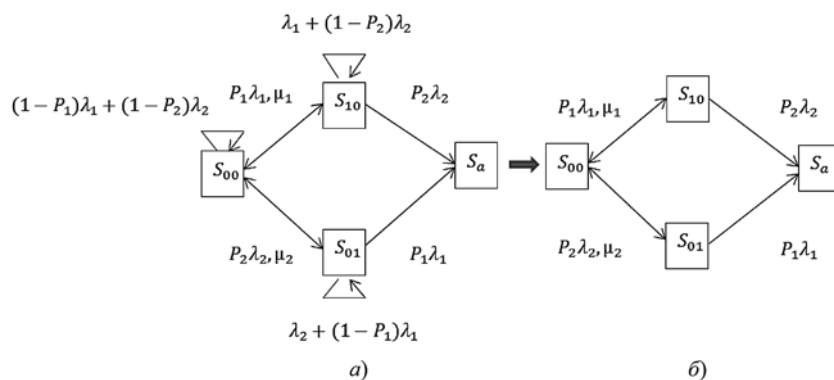


Рис. 7. Размеченные графы системы состояний случайного процесса для модели безопасности информации:

*a* — исходная модель; *б* — марковская модель с потерями

система с отказами (без восстановления) в отношении угрозы нарушения конфиденциальности обрабатываемой информации.

Поскольку стационарный (установившийся) режим функционирования для этой системы отсутствует, могут моделироваться исключительно временные эксплуатационные характеристики безопасности, причем в данном случае не среднее время наработки на отказ безопасности информационной системы, а среднее время наработки до отказа (не восстанавливаемая система) безопасности информации.

При моделировании для вычисления среднего времени  $T_i$  пребывания системы в каждом  $i$ -м состоянии в системе уравнений Колмогорова нужно положить нулю все производные  $P'_i$  ( $P'_i = 0$ ), кроме  $P'_0$  (считаем, что в начальный момент вероятность первого состояния  $P_0 = 1$ ). Тогда на основании теоремы о дифференцировании изображений в преобразовании Лапласа левая часть первого уравнения будет равна  $-1$ . В правых частях уравнений вместо  $P_i$  подставляются  $T_i$ , и относительно них решается система алгебраических уравнений [11].

Рассчитав же значения  $T_i$  и просуммировав их для состояний, не являющихся поглощающими  $S_j$ , можно вычислить среднее время наработки до отказа  $T_{ндa}$  эксплуатационной безопасности информа-

ции (фатальный отказ) информационной системы в отношении угрозы атаки, направленной на нарушение конфиденциальности обрабатываемой информации. Для модели, представленной на рис. 7,

$$T_{нда} = T_{00} + T_{01} + T_{10}.$$

Для наглядности иллюстрации излагаемых подходов к моделированию в работе использовались достаточно простые модели для случаев, когда угроза атаки создается одной или двумя угрозами уязвимостей. Вместе с тем, используя рассмотренные подходы, можно моделировать реализацию угроз атак любой сложности, при этом увеличение числа угроз уязвимостей, создающих угрозу атаки, соответствующим образом скажется на увеличении числа состояний в размеченном графе системы состояний случайного процесса.

### Заключение

Рассмотренные в данной работе подходы к моделированию реализации угрозы атаки позволяют учитывать при моделировании заинтересованность и готовность потенциального нарушителя к осуществлению атаки на информационную систему, обрабатываемая информация в которой для него представляет определенную ценность. С этой целью в математическую модель включена эксплуатационная характеристика безопасности — вероятность реализации потенциальным нарушителем реальной угрозы уязвимостей — интерпретируемая как количественная мера опасности потенциального нарушителя для информационной системы. Вопросам определения данной ключевой характерис-

тики безопасности, предполагающего построение математической модели потенциального нарушителя, будет посвящена следующая работа авторов.

### Список литературы

1. Шеглов К. А., Шеглов А. Ю. Интерпретация и моделирование угрозы атаки на информационную систему. Часть 1. Моделирование угрозы уязвимости и интерпретация угрозы атаки // Информационные технологии. 2015. Т. 21, № 12. С. 930—940.
2. Шеглов К. А., Шеглов А. Ю. Интерпретация и моделирование угрозы атаки на информационную систему. Часть 2. Моделирование угрозы атаки // Информационные технологии. 2016. Т. 22, № 1. С. 54—64.
3. Шеглов К. А., Шеглов А. Ю. Математические модели эксплуатационной информационной безопасности // Вопросы защиты информации. 2014. Вып. 106, № 3. С. 52—65.
4. Котенко Д. И., Котенко И. В., Саенко И. Б. Методы и средства моделирования атак в больших компьютерных сетях: состояние проблемы // Труды СПИИРАН. 2012. Вып. 3 (22). С. 5—30.
5. Zhang S., Song S. Novel Attack Graph Posterior Inference Model Based on Bayesian Network // Journal of Information Security. 2011. N. 2. P. 8—27.
6. Kichkaylo T., Ryutov T., Orosz M. D., Neches R. Planning to Discover and Counteract Attacks // Informatica (Slovenia). 2010. N. 34 (2). P. 159—168.
7. Gamal M. M., Hasan D., Hegazy A. F. A Security Analysis Framework Powered by an Expert System // International Journal of Computer Science and Security. 2011. Vol. 4. P. 505—526.
8. Шеглов К. А., Шеглов А. Ю. Вопросы корректности и универсальности подхода к моделированию угроз атак // Информационные технологии. 2016. Т. 22, № 11. С. 854—861.
9. Шеглов К. А., Шеглов А. Ю. Моделирование угрозы безопасности информационной системы с использованием аппроксимирующих функций // Известия высших учебных заведений. Приборостроение. 2016. Т. 59. № 1. С. 50—59.
10. Алиев Т. И. Основы моделирования дискретных систем. СПб.: Изд. СПбГУ ИТМО, 2009.
11. Вентцель Е. С. Исследование операций. Задачи, принципы, методология. М.: Высшая школа, 2007.

K. A. Shcheglov, Graduate Student, A. Yu. Shcheglov, Professor, e-mail: info@npp-itb.spb.ru, St. Petersburg University of ITMO, Russia

## Operational Safety. Potential Offender Attack Threats Implementation Modeling

*We researched attack threat exploitation parameters and characteristics elaborating modeling problem. Its parameters differ from reliability ones by taking into account potential violator interest and willingness for executing real attack threat and also taking into account attack goals. We suggest an approach for building mathematical model of potential violator, introducing quantitative complexity characteristics (measures) for attack threat implementation and for informational system danger level (from potential violator). We suggest an approach of quantitative estimation of attack threat actuality (to make decision on security implementation while designing informational security system). We research attack threat modeling problems and features while different potential violator goals (like information confidentiality, integrity and availability violation).*

**Keywords:** attack threat, vulnerability threat, mathematical modeling, potential violator model, attack threat implementation complexity, danger from potential violator, attack threat model, attack goal

### References

1. Shcheglov K. A., Shcheglov A. Yu. Interpretaciya i modelirovanie ugrozy ataki na informacionnyuyu sistemu. Chast 1. Modelirovanie ugrozy uyazvimosti i interpretaciya ugrozy ataki, *Informacionnye tehnologii*, 2015, vol. 21, no. 12, pp. 930—940.
2. Shcheglov K. A., Shcheglov A. Yu. Interpretaciya i modelirovanie ugrozy ataki na informacionnyuyu sistemu. Chast 2. Modelirovanie ugrozy uyazvimosti i interpretaciya ugrozy ataki, *Informacionnye tehnologii*, 2016, vol. 22, no. 1, pp. 54—64.

rovaniye ugrozy uyazvimosti i interpretaciya ugrozy ataki, *Informacionnye tehnologii*, 2015, vol. 21, no. 12, pp. 930—940.

2. Shcheglov K. A., Shcheglov A. Yu. Interpretaciya i modelirovanie ugrozy ataki na informacionnyuyu sistemu. Chast 2. Modelirovanie ugrozy uyazvimosti i interpretaciya ugrozy ataki, *Informacionnye tehnologii*, 2016, vol. 22, no. 1, pp. 54—64.



rovanie ugrozy ataki, *Informacionnye tehnologii*, 2016, vol. 22, no. 1, pp. 54–64.

3. **Shcheglov K. A., Shcheglov A. Yu.** Matematicheskie modeli ekspluatatsionnoy informacionnoy bezopasnosti, *Voprosy zaschity informacii*, 2014, vyp. 106, no. 3, pp. 52–65.

4. **Kotenko D. I., Kotenko I. V., Saenko I. B.** Metody i sredstva modelirovaniya atak v bolshih kompyuternyh setyah: sostoyanie problemy, *Trudy SPIIRAN*, 2012, vyp. 3 (22), pp. 5–30.

5. **Zhang S., Song S.** Novel Attack Graph Posterior Inference Model Based on Bayesian Network, *Journal of Information Security*, 2011, no. 2, pp. 8–27.

6. **Kichkaylo T., Ryutov T., Orosz M. D., Neches R.** Planning to Discover and Counteract Attacks, *Informatika (Slovenia)*, 2010, no. 34 (2), pp. 159–168.

7. **Gamal M. M., Hasan D., Hegazy A. F.** A Security Analysis Framework Powered by an Expert System, *International Journal of Computer Science and Security*, 2011, vol. 4, pp. 505–526.

8. **Shcheglov K. A., Shcheglov A. Yu.** Voprosy korrektnosti i universalnosti podhoda k modelirovaniyu ugroz atak, *Informacionnye tehnologii*, 2016, vol. 22, no. 11, pp. 854–861.

9. **Shcheglov K. A., Shcheglov A. Yu.** Modelirovanie ugrozy bezopasnosti informacionnoy sistemy s ispolzovaniem approksimiruyuschih funktsiy, *Izvestiya vysshih uchebnykh zavedenij. Priborostroenie*, 2016, vol. 59, no. 1, pp. 50–59.

10. **Aliev T. I.** *Osnovy modelirovaniya diskretnykh sistem*. SPb: Izd. SPbGU ITMO, 2009.

11. **Ventcel E. S.** *Issledovanie operatsiy. Zadachi, principy, metodologiya*, Moscow, Vysshaya shkola, 2007.

УДК 004.942

DOI:10.17587/it.23.41-49

**В. Н. Костин**, канд. техн. наук, доц., e-mail: vladimirkostin57@mail.ru,

Федеральное государственное бюджетное образовательное учреждение высшего образования "Оренбургский государственный университет"

## Синтез оптимального размещения технических средств систем физической защиты критически важных объектов

Разработан метод оптимального размещения технических средств защиты на критически важных объектах. На основе графа проникновения нарушителя формировались все возможные пути проникновения на объект, представленные в виде матрицы инцидентности. С помощью задачи о покрытии на матрице инцидентности определяются все варианты пресечения проникновения нарушителя (развития опасности). На основе метода динамического программирования для разных по степени важности (риска) критических элементов объекта при возникновении чрезвычайной ситуации (разных требований к величине защищенности) определяется оптимальный план размещения технических средств защиты, удовлетворяющий заданным критериям защищенности объекта (требованиям безопасности).

**Ключевые слова:** технические средства защиты, система физической защиты, матрица инцидентности, план оптимального размещения технических средств защиты, критерий защищенности объекта

### Введение

В условиях возрастающих требований к обеспечению безопасности объектов жизнедеятельности человека решение задачи повышения эффективности и рентабельности проектных решений систем физической защиты (СФЗ) является актуальной. Сложный и дорогостоящий процесс проектирования СФЗ предъявляет высокие требования к проектным решениям, особенно принятым на ранних стадиях разработки проекта. При этом высокая стоимость СФЗ не позволяет провести практическую проверку принимаемых проектных решений. Функциональная модель построения СФЗ в виде последовательных этапов представлена на рис. 1. Завершающим и важным этапом разработки СФЗ является синтез оптимального варианта размещения технических средств защиты (ТСЗ).

При организации СФЗ объекта используется классический принцип последовательных рубежей охраны, при нарушении которых угроза своевременно обнаруживается, и ее продвижению будут препятствовать надежные преграды. Оптимальное расположение рубежей охраны и размещение на них ТСЗ (обнаружения и противодействия) состав-

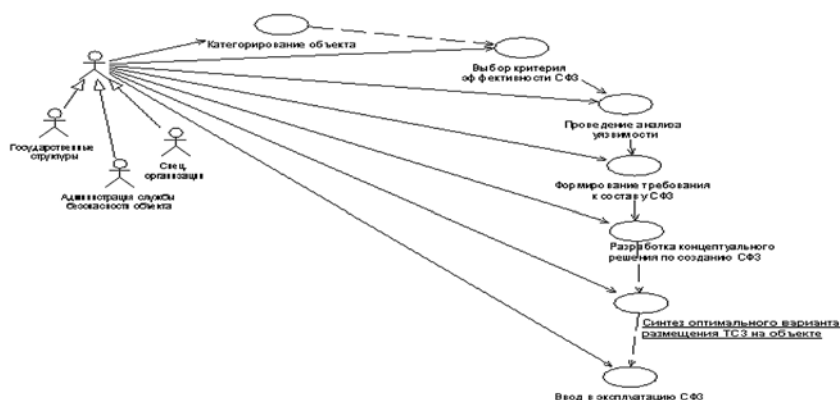


Рис. 1. Функциональная модель создания СФЗ

ляет основу физической защиты любого объекта [1—3]. Разработка метода формирования оптимального варианта размещения ТСЗ на объекте и является целью настоящих исследований.

Вопросы синтеза элементов СФЗ с использованием логико-вероятностных методов (ЛВМ) рассматривались в статьях Панина О. А. [4, 5], делался акцент на большую трудоемкость ЛВМ [5, 6]. Материал данной статьи является продолжением исследований в направлении автоматизации построения функций алгебры логики на основе системного анализа, методических вопросов оптимизации размещения ТСЗ при проектировании СФЗ.

**Постановка задачи.** Цель работы — разработать метод оптимального размещения ТСЗ СФЗ на основе формирования логических функций проникновения нарушителя на объект, которые представлены как функции условий проникновения в виде конъюнкции логических переменных — аргументов. Аргументы функции — ребра графа проникновения, представленные как булевы переменные. Логические функции необходимо сформировать в матрицу инцидентности, на основе которой с помощью задач оптимизации о покрытии и динамического программирования (ДП) провести синтез оптимального размещения ТСЗ, удовлетворяющий заданным требованиям эффективности СФЗ.

В качестве показателя эффективности СФЗ определим вероятность нахождения объекта охраны в безопасном состоянии  $P_{БС}(V)$ , введенный Никитиным В. В. [7]. То есть необходимо добиться заданной вероятности обнаружения нарушителя и вероятности своевременного прибытия сил охраны в точку пресечения при минимальной стоимости на проектирование, монтаж и эксплуатацию ТСЗ СФЗ. Таким образом, вероятность нахождения объекта защиты в безопасном состоянии (вероятность защиты объекта) определим как произведение вероятностей выполнения своей задачи каждой из составляющих СФЗ: системы обнаружения, системы задержки и системы сил охраны:

$$P_{БС}(V) = P_{О}(A) \cdot P_{СП}(T/A) \cdot P_{Н}(V/T), \quad (1)$$

где  $P_{О}(A)$  — вероятность обнаружения нарушителя при проникновении;  $P_{СП}(T/A)$  — вероятность своевременного прибытия сил охраны в точку перехвата при условии обнаружения нарушителя;  $P_{Н}(V/T)$  — вероятность нейтрализации нарушителя при условии своевременного прибытия охраны. Показатель не исследуется — принимаем за единицу.

Критерием эффективности СФЗ будем считать  $P_{О}(A) \geq P_{ОЗ}$  — вероятность обнаружения нарушителя не менее заданной  $P_{ОЗ}$  и  $P_{СП}(T/A) \geq P_{СПЗ}$  — вероятность своевременного прибытия сил охраны в точку перехвата при условии обнаружения не менее заданной  $P_{СПЗ}$ . При этом стоимость затрат на проектирование, монтаж и эксплуатацию ТСЗ стремится к минимуму:  $C_{СТЗ} \rightarrow \min$ . Считаем, что стои-

мость ТСЗ, их монтажа и эксплуатации прямо пропорционально зависят от длины их размещения, т. е. стоимость определяется как  $C_{СТЗ} = K \cdot L$ , где  $L$  — длина размещения ТСЗ на рубежах зон охраны;  $K$  — коэффициент пропорциональности.

Особенностью данной задачи является то, что вследствие разной степени важности критических элементов (КЭ) объекта им задаются разные требования безопасности. Поэтому для КЭ с большей степенью важности необходимо формирование дополнительных покрытий для получения приращения показателя безопасности.

Решение поставленной задачи укладывается в следующую последовательность этапов.

1. На основе графовой модели проникновения нарушителя сформировать все пути достижения цели в виде логических функций, представленных матрицей инцидентности.
2. Сформировать множество вариантов размещения ТСЗ с помощью задачи о покрытии на матрице инцидентности.
3. Провести синтез вариантов оптимального размещения ТСЗ на основе задачи ДП.
4. Сформировать дополнительные варианты покрытий для повышения показателя безопасности более важных КЭ.
5. Провести синтез дополнительных вариантов размещения ТСЗ, обеспечивающих повышение показателя безопасности более важных КЭ.

**Решение задачи** рассмотрим на модельном примере (рис. 2).

Охраняемый объект представляет собой сложную систему, состоящую из множества связанных зон, зданий (элементов) различной природы, назначения и важности. Вся территория объекта имеет ограждение, контрольно-пропускной пункт (КПП), автомобильный КПП (АКПП), железнодорожный КПП (ЖКПП). На объекте имеются три КЭ, подлежащие охране: цех 1 (Y10), цех 2 (Y12) и склад готовой продукции (Y4), а также административное

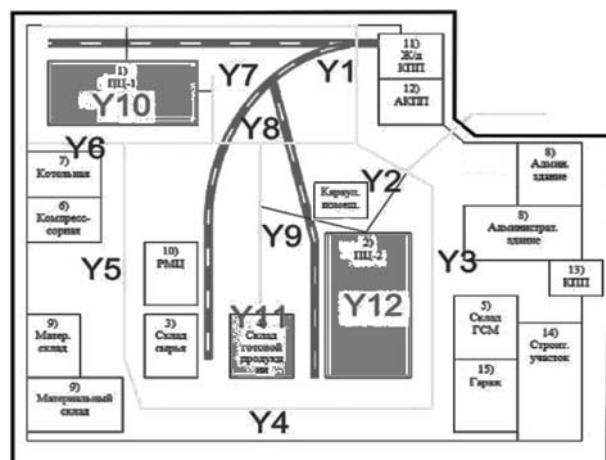


Рис. 2. План объекта

здание. Цель посягательства — проникнуть в цех 1, цех 2 или склад готовой продукции и провести диверсионно-террористическое действие.

*Первый этап.* Представим сценарий проникновения нарушителя в виде разветвленного ориентированного графа (рис. 3). Вершины графа ( $Y_k$ ) — рубежи достижения нарушителем определенного результата на пути к цели, ребра — варианты возможных перемещений между рубежами, представленные как логические аргументы функций проникновения. Ребра обозначим  $X_j$ , где  $j$  — номер ребра в графе,  $X_0 - X_{25}$  — варианты перемещения через зоны между рубежами. Полученный граф назовем моделью достижимости нарушителем цели (модель развития опасности). Граф имеет  $k$  событий, начальное событие  $Y_0$  является инициирующим. Наступление хотя бы одного конечного события  $Y_{10}, Y_{11}, Y_{12}$  означает факт достижения нарушителем цели. Вероятность нахождения событий  $Y_{10}, Y_{11}, Y_{12}$  в безопасном состоянии и будет показателем эффективности СФЗ. Наступление промежуточных событий

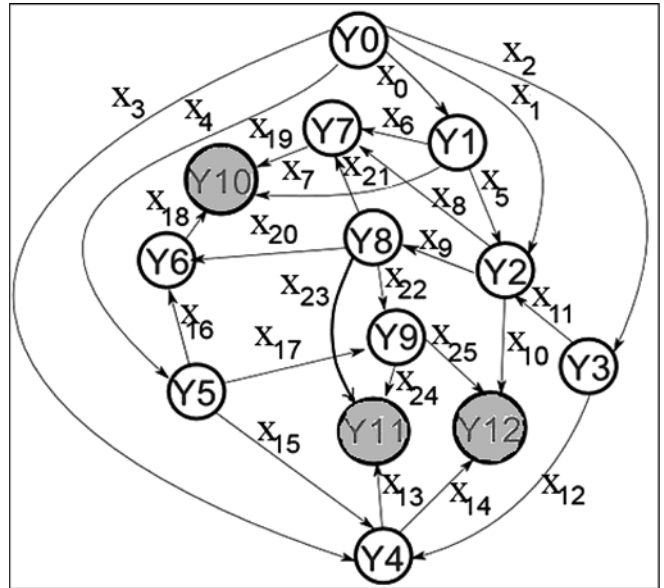


Рис. 3. Граф достижимости нарушителем своей цели

Таблица 1

Матрица инцидентности

№ функции проникновения	Ребра графа																										
	$X_0$	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$	$X_9$	$X_{10}$	$X_{11}$	$X_{12}$	$X_{13}$	$X_{14}$	$X_{15}$	$X_{16}$	$X_{17}$	$X_{18}$	$X_{19}$	$X_{20}$	$X_{21}$	$X_{22}$	$X_{23}$	$X_{24}$	$X_{25}$	
1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
3	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
4	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
7	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
11	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
12	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
13	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
14	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
15	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
16	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
17	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
18	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
19	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
23	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
24	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
26	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
28	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1

является условием логических комбинаций двух и более ребер графа на пути проникновения нарушителя.

Необходимо определить все пути проникновения из начальной вершины  $Y_0$  в конечные вершины графа  $Y_{10}, Y_{11}, Y_{12}$ . Пути определяются с помощью операции композиции матрицы смежности графа. При этом получаем матрицу, в которой будут представлены все пути различного состава ребер из начального события в конечные. Каждый путь проникновения описывается логической функцией, аргументами которой являются множество упорядоченных ребер графа. Аргументы функции — булевы переменные: 1 — если ребро входит в путь проникновения, 0 — если не входит. Всего получаем 32 логические функции проникновения, сведенные в матрицу инцидентности (табл. 1).

Строки в матрице — пути проникновения (номера логических функций), а столбцы — упорядоченные ребра графа. Элементы матрицы в строке связаны конъюнктивно, а множество функций проникновения — дизъюнктивно (совершенная дизъюнктивная нормальная форма). Полученные логические функции позволяют оценить вероятность реализации цели нарушителем на каждом пути проникновения, т. е. оценить эффективность СФЗ. В вероятностном смысле эффективность СФЗ будет определяться вероятностью нереализации ни одной функции проникновения.

*Второй этап.* Число функций проникновения определяет число вариантов доступа нарушителя на объект. Необходимо исключить все пути проникновения. Это задача нахождения минимального сечения на графе. Она решается путем определения минимального покрытия на матрице инцидентности.

Постановка задачи о покрытии: все пути проникновения покрыть минимальным числом ребер:

$$\sum_{j=1}^n X_j \rightarrow \min, \quad (2)$$

где  $x_j = \begin{cases} 1 & \text{если } j\text{-е ребро графа входит в состав} \\ & \text{покрытия;} \\ 0 & \text{в противном случае.} \end{cases}$

При этом избыточность не реализованных возможностей покрывающих ребер графа стремится к минимуму:

$$\sum_{j=1}^n a_{ij} x_j \rightarrow \min, \quad i = \overline{1, m}. \quad (3)$$

При ограничении: каждое ребро покрывает хотя бы один путь проникновения:

$$\sum a_{ij} x_j \geq 1. \quad (4)$$

Исходные данные задаются с помощью матрицы инцидентности:

$$A = \|a_{ij}\|, \quad (5)$$

где  $i = 1 - m$  — номер пути проникновения;

$j = 1 - n$  — номер ребра графа;

$a_{ij} = \begin{cases} 1, & \text{если } j\text{-е ребро входит в } i\text{-й путь проникновения;} \\ 0 & \text{в противном случае.} \end{cases}$

Данная задача решается методом ветвей и границ. Для оценки границ решения необходимо определить мощность каждого ребра:

$$W(j) = E'(j) - S(j), \quad (6)$$

где  $E'(j)$  — потенциал  $j$ -го ребра;

$$E'(j) = \sum_{\forall i \in I} a_{ij}, \quad j \in J, \quad i \in I, \quad (7)$$

$I$  — множество маршрутов, которые не покрыты ребрами;

$$S(j) = \sum_{\forall i \in I'} a_{ij}, \quad j = (J/J_1), \quad (8)$$

$I'$  — множество маршрутов, которые покрыты ребрами;  $S(j)$  — избыточность или не использованные возможности  $j$ -го ребра.

Для того чтобы задача о покрытии быстро сходилась к конечному результату, вводится оценка перспективной мощности  $j$ -го ребра:

$$\tilde{W}(j) = W(i) - S(j), \quad (9)$$

где  $W(i)$  — мощность  $i$ -го ребра, из которого выполняется ветвление;  $S(j)$  — избыточность ребра, претендующего на включение в покрытие;  $\tilde{W}(j)$  — перспективная мощность  $j$ -го ребра.

Решая задачу о покрытии для каждого ребра графа (столбца матрицы) получили 10 покрытий, которые представлены в табл. 2. Причем каждое покрытие позволяет контролировать все пути проникновения при размещении на них ТСЗ, т. е. покрытие может ассоциироваться с каким-то вариантом размещения ТСЗ, который будет характеризоваться вероятностью обнаружения и стоимостью.

С точки зрения системного анализа процесс получения всех путей проникновения и формирование множества вариантов их покрытий является задачей декомпозиции сложной задачи на простые подзадачи. После этой задачи, согласно теории системного анализа, решается задача синтеза покрытий для оптимального размещения ТСЗ СФЗ.

*Третий этап.* Возможны две постановки задачи оптимизации размещения ТСЗ: минимизировать стоимость затрат на реализацию СФЗ при заданной вероятности противодействия нарушителям; максимизировать вероятность защиты объекта от воздействия нарушителей при заданном значении затрат на обеспечение защиты. С учетом выбранного критерия эффективности СФЗ решаем первую задачу. Определим требования к СФЗ: вероятность обнаружения нарушителя  $P_{\text{О}}(A) \geq 0,9$ ; вероятность своевременного прибытия сил охраны для нейтрализации нарушителя  $P_{\text{СП}}(T/A) \geq 0,8$ .

Таблица покрытий

№ покрытия	Ребра покрытий	Длина покрытия, м	Номер протяжен.	Удаление от КЭ, м	Примечание
1	$X_3X_4 X_6X_7 X_8X_9X_{10}X_{12}$	1560	2	190	Близко к КЭ
2	$X_0X_1X_2X_3X_4$	3000	10	460	Подходит
3	$X_1X_2X_3X_4X_5 X_6X_7$	2690	9	190	Близко к КЭ
4	$X_0X_1X_3X_4 X_{11}X_{12}$	2500	6	400	Подходит
5	$X_3X_4 X_7X_9X_{10}X_{12}X_{19}$	1410	1	240	Близко к КЭ
6	$X_3X_4 X_7X_{10}X_{12}X_{19}X_{20}X_{22}X_{23}$	2540	7	240	Близко к КЭ
7	$X_6X_7X_8X_9X_{10}X_{13}X_{14}X_{17}X_{18}$	2300	4	250	Подходит
8	$X_4 X_6X_7X_8X_9X_{10}X_{13}X_{14}$	1860	3	190	Близко к КЭ
9	$X_6X_7X_8X_9X_{10}X_{13}X_{14}X_{16}X_{17}$	2350	5	260	Подходит
10	$X_3X_6X_7X_8X_9X_{10}X_{12}X_{15}X_{16}X_{17}$	2680	8	240	Близко к КЭ

В этой постановке необходимо на множестве комбинаций покрытий сформировать размещение ТСЗ, обеспечивающее заданные критерии эффективности с минимальной стоимостью ТСЗ, их монтаж и эксплуатацию. Стоимость  $C_{ТСЗ}$  размещения ТСЗ на покрытии пропорционально ( $K$ ) зависит от его длины  $L$ . Определив длину (стоимость) каждого покрытия, необходимо сформировать возрастающий ряд. Обычно при проектировании формируют несколько рубежей обнаружения и задержки продвижения нарушителя. В первую очередь в оптимальное множество включают покрытия с минимальной длиной и необходимым удалением от КЭ. Каждое дополнительное покрытие будет повышать вероятность обнаружения и вероятность распознавания поведения нарушителя (конкретный путь движения) для принятия решения его нейтрализации. Формирование множества покрытий заканчивается при достижении заданной вероятности обнаружения. Кроме того, каждое покрытие еще характеризуется удалением от охраняемых КЭ. Этот параметр вводится для оценки вероятности своевременного прибытия сил охраны для нейтрализации нарушителя. Для этого выделяется критическая зона — минимальное расстояние от КЭ, после которой противодействие ввиду недостатка времени невозможно. Покрытия, попавшие в эту зону, исключаются из процесса оптимизации. Оптимизация заключается в минимизации общей длины покрытий (стоимости) при обеспечении заданной вероятности обнаружения нарушителя и своевременного прибытия сил охраны.

Для обеспечения заданного критерия эффективности СФЗ рассмотрим покрытия минимальной длины и с минимальным числом общих ребер. Из десяти результирующих покрытий выбраны четыре покрытия (2, 4, 7, 9), причем покрытия 2 и 4 имеют общие ребра (пересекаются), покрытия 7 и 9 также пересекаются. Остальные покрытия или попадают в критическую зону, или имеют большую длину, а значит стоимость размещения ТСЗ. Таким образом, мы получили два множества непересе-

кающихся покрытий, из которых выбираем наиболее короткие: 4 и 7.

Покрытие 4 короче покрытия 2 на 500 м, но расположение покрытия 2 дополнительно защищает административное здание и склад горюче-смазочных материалов (ГСМ). В этом случае выбор покрытия определяет вышестоящее руководство. На рис. 4 данные покрытия показаны соответственно сплошной, штриховой и штрих-пунктирной линиями.

Для каждого покрытия определим удаление от КЭ объекта и по этому значению определим вероятность своевременного прибытия сил охраны. Удаление КЭ от караула составляет:  $Y_{10}$  — 240 м;  $Y_{11}$  — 200 м;  $Y_{12}$  — 90 м. Удаление покрытий от ближайших КЭ охраны составляет: 7-го — 250 м; 4-го — 400 м; 2-го — 460 м. Для вычисления значения вероятности своевременного прибытия  $P_{СП}(T/A)$  используется модель, отражающая рост вероятности своевременного прибытия сил охраны по мере уве-

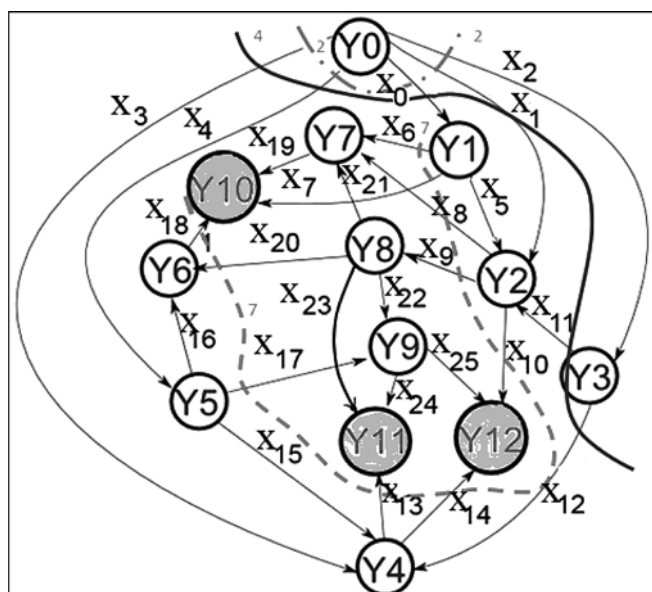


Рис. 4. Результирующие покрытия

Характеристики технических средств защиты

Тип ТСЗ	Стоимость за штуку, руб.	Стоимость покрытия 7, тыс. руб.	Стоимость покрытия 2, тыс. руб.	Вероятность обнаружения, $P_o$	Угол обзора, °	Дальность, м
Первый тип (CNBWFL21S)	4800	184	240	0,60	70—90	60
Второй тип (SCANALL)	7000	203	266	0,80	70—90	80
Третий тип (IWPC22ZW)	6000	198	258	0,70	70—90	70

личения времени задержки нарушителей на физических барьерах (заборы, преграды, замки и т. п.) и по мере сокращения времени перемещения сил охраны на требуемое расстояние. Вероятность определялась по следующей формуле [7, с. 23]:

$$P_{СП}(T/A) = \exp[1,7(t_d - t_f)/\sigma] / [1 + \exp[1,7(t_d - t_f)/\sigma]]. \quad (10)$$

Вероятность  $P_{СП}(T/A)$  своевременного прибытия сил охраны в точку перехвата является функцией

средних значений времени  $t_d$  задержки нарушителей физическими барьерами, времени  $t_f$  занятия позиций силами охраны и средних квадратичных отклонений  $\sigma$ . Время занятия позиций подразделяется на время сборов сил охраны и на время ее перемещения к месту КЭ. При переходе от пространственных характеристик к временным параметрам перемещений для уровня оснащенности нашего объекта физическими барьерами считаем, что скорости движения составляют: сил охраны — 4,4 м/с; нарушителя — 2 м/с; а их средние квадратичные отклонения соизмеримы [7, с. 22]. Покрытия 2, 4, и 7 удовлетворяют требованиям вероятности своевременного прибытия, так как  $P_{СП}(T/A) \geq 0,8$ .

Запишем покрытия в порядке увеличения их длины: покрытие 7 — 2300 м; 4 — 2500 м; 2 — 3000 м. Учитывая, что покрытие 2 обеспечивает охрану склада ГСМ и административного здания, имеет смысл покрытие 4 заменить покрытием 2 (ограждение объекта).

Для покрытий 4 и 7 на основе решения задачи ДП определим наилучший вариант выбора типов ТСЗ. Для формирования исходных данных задачи из большого множества ТСЗ на основе метода анализа иерархий определены наиболее приемлемые типы ТСЗ, характеристики которых представлены в табл. 3.

Для решения задачи ДП используется табличный алгоритм. Данные из файла автоматически считываются, и формируется шапка таблицы исходных данных (рис. 5). Задаются требуемые условия, и нажимается кнопка "выполнить". Оконная форма результатов реализации алгоритма программы для вероятности обнаружения 0,9 представлена на рис. 6.

Вывод: на покрытии 2 располагать ТСЗ первого типа с вероятностью обнаружения 0,6, а на покрытии 7 — ТСЗ второго типа с вероятностью обнаружения 0,8. Такое расположение ТСЗ обеспечит  $P_o(A) = 0,92$  нарушителя на каждом маршруте проникновения при минимуме стоимости — 443 тыс. руб. (рис. 6). Результат размещения ТСЗ представлен на рис. 7.

*Четвертый этап.* Для обеспечения повышения уровня безопасности КЭ объекта рассмотрим пересекающиеся покрытия с выделением путей проникновения нарушителя к КЭ повышенной опасности. Например, для цеха 1 (Y10) необходимо обеспечить

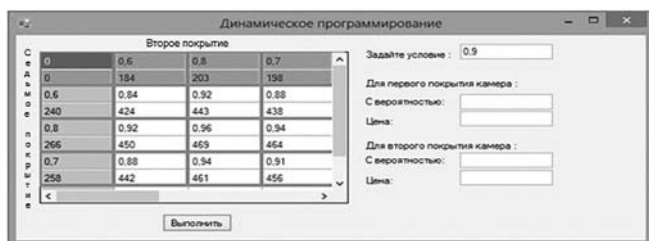


Рис. 5. Исходная оконная форма

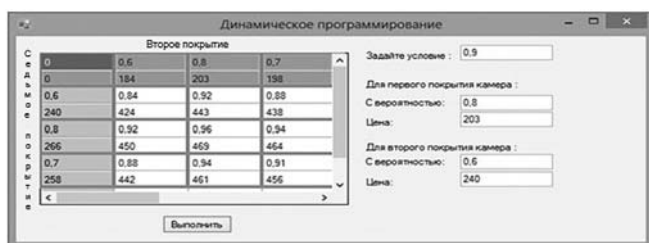


Рис. 6. Результаты выполнения модуля

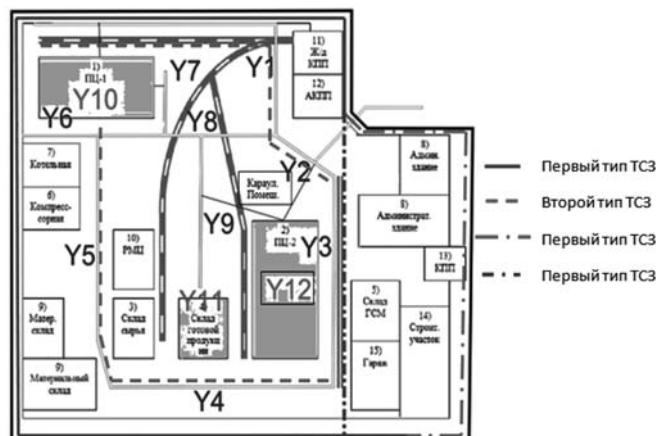


Рис. 7. План размещения камер наблюдения на объекте

вероятность обнаружения не 0,9, а  $P_O(A) \geq 0,95$ . В этом случае на графе проникновения нарушителя (см. рис. 3) с помощью алгоритма обхода графа в ширину определим маршруты (функции) проникновения только на Y10 [8]. Результаты решения сведем в матрицу инцидентности.

Идеология рассуждений такова: необходимо найти дополнительное покрытие, которое не пересекалось с ранее выделенными покрытиями, или выявить такие покрытия, которые пересекались незначительно по протяженности с назначенными покрытиями и имели возможность увеличения вероятности обнаружения. При этом полученные покрытия должны иметь минимальную протяженность (стоимость) и достаточное удаление от Y10 для своевременного реагирования сил охраны на проникновение. То есть ребра, вошедшие в покрытие при решении предыдущей задачи, на которых уже размещены ТСЗ, необходимо не исключать из рассмотрения. В результате решения получаем две-

надцать функций проникновения, представленные в виде матрицы инцидентности (табл. 4).

Решив задачу о покрытии, получаем дополнительное множество покрытий. Из этого множества по изложенной выше методике выберем покрытия, которые минимальны по протяженности и расположение которых обеспечивает своевременное прибытие сил охраны. Характеристики полученных покрытий приведены в табл. 5.

*Анализ содержания покрытий.* Очевидно, что формировать новое покрытие (рубеж обнаружения) для установки ТСЗ будет затратным решением, поэтому необходимо на максимально объединяющихся покрытиях предыдущей задачи обеспечить требуемую вероятность обнаружения для Y10. Для решения этой задачи выбираем покрытия, которые не пересекаются с покрытием 7 предыдущей задачи пересекаются с покрытием 2 и одновременно имеют наименьшую длину.

Таблица 4

Матрица инцидентности

№ функции проникновения	Ребра графа																					
	X <sub>0</sub>	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>	X <sub>15</sub>	X <sub>16</sub>	X <sub>17</sub>	X <sub>18</sub>	X <sub>19</sub>	X <sub>20</sub>	X <sub>21</sub>
1	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
2	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0
3	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1
4	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
5	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
7	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0
8	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1
9	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
10	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	0
11	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1
12	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0

Таблица 5

Таблица покрытий

№ покрытия	Ребра покрытия	Длина покрытия, м	Номер протяженности	Удаление от Y <sub>10</sub> , м	Примечание
1	X <sub>7</sub> X <sub>18</sub> X <sub>19</sub>	640	1	110	Близко к Y <sub>10</sub>
2	X <sub>4</sub> X <sub>6</sub> X <sub>7</sub> X <sub>8</sub> X <sub>9</sub>	1860	9	420	Пересекает
3	X <sub>6</sub> X <sub>7</sub> X <sub>8</sub> X <sub>9</sub> X <sub>16</sub>	1320	5	220	Пересекает
4	X <sub>0</sub> X <sub>1</sub> X <sub>2</sub> X <sub>4</sub>	2600	14	510	Длинное
5	X <sub>0</sub> X <sub>1</sub> X <sub>2</sub> X <sub>16</sub>	2100	11	260	Длинное
6	X <sub>0</sub> X <sub>1</sub> X <sub>4</sub> X <sub>11</sub>	2110	12	250	Длинное
7	X <sub>0</sub> X <sub>1</sub> X <sub>11</sub> X <sub>16</sub>	1600	7	240	Удовлетворяет
8	X <sub>1</sub> X <sub>2</sub> X <sub>4</sub> X <sub>6</sub> X <sub>7</sub>	2340	13	210	Пересекает
9	X <sub>1</sub> X <sub>2</sub> X <sub>5</sub> X <sub>6</sub> X <sub>7</sub> X <sub>16</sub>	1890	10	220	Пересекает
10	X <sub>1</sub> X <sub>4</sub> X <sub>5</sub> X <sub>6</sub> X <sub>7</sub> X <sub>12</sub>	1790	8	230	Пересекает
11	X <sub>1</sub> X <sub>5</sub> X <sub>6</sub> X <sub>7</sub> X <sub>11</sub> X <sub>16</sub>	1390	6	220	Пересекает
12	X <sub>4</sub> X <sub>7</sub> X <sub>19</sub> X <sub>20</sub>	1240	4	150	Близко, пересекает
13	X <sub>7</sub> X <sub>16</sub> X <sub>19</sub> X <sub>20</sub>	730	2	150	Близко, пересекает
14	X <sub>6</sub> X <sub>7</sub> X <sub>8</sub> X <sub>18</sub> X <sub>21</sub>	1210	3	230	Пересекает

Из четырнадцати покрытий выбираем наиболее приемлемые. Для покрытий 1, 12 и 13 не выполняется условие своевременного прибытия (9). Покрытия 2, 3, 8, 9, 10, 11 и 14 частично пересекаются с покрытием 7 предыдущей задачи оптимизации. На этих покрытиях было сделано размещение ТСЗ второго типа с  $P_0 = 0,8$ , т. е. нет возможности наращивать показатель вероятности обнаружения. Поэтому выберем наиболее короткие по протяженности покрытия, которые не имеют общих ребер с покрытием 7 предыдущей задачи. Получаем покрытия 4 ( $X_0X_1X_2X_4$ ), 5 ( $X_0X_1X_2X_{16}$ ), 6 ( $X_0X_1X_4X_{11}$ ), 7 ( $X_0X_1X_{11}X_{16}$ ). Для обеспечения вероятности 0,95 достаточно одного самого короткого по протяженности покрытия — номер 7 ( $X_0X_1X_{11}X_{16}$ ) протяженностью 1600 м.

**Пятый этап.** Решим задачу ДП для покрытий 7 и 14 для обеспечения вероятности обнаружения 0,95

КЭ Y10. В результате решения получили: на оба покрытия необходимо устанавливать ТСЗ второго типа. На покрытие 14 уже установлены ТСЗ второго типа. Чтобы получить вероятность не менее 0,95 надо на покрытие 7 установить ТСЗ второго типа, — на ребрах  $X_0, X_1, X_{11}$  ТСЗ первого типа заменить на ТСЗ второго типа, а на ребре  $X_{16}$  дополнительно установить ТСЗ второго типа, т. е. на покрытие 2 предыдущей задачи частично ТСЗ первого типа заменить вторым. Дополнительное покрытие показано пунктиром на рис. 8 и на плане объекта рис. 9. Получаем избыток вероятности обнаружения в первой задаче по всем путям, проходящим через ребра  $X_0, X_1$  и  $X_{11}$  при движении в направлении КЭ Y11 и Y12. Также на схеме видно, что имеется избыток вероятности обнаружения путей, проходящих через ребро  $X_{16}$ , поэтому на этом ребре можно рекомендовать установить менее дорогие камеры первого типа с  $P_0 = 0,6$  (ТСЗ первого типа на  $X_{16}$ ).

**Заключение.** Разработан метод оптимального размещения ТСЗ. На основе формирования множества логических функций проникновения и решения задач о покрытии и ДП проведен синтез оптимального размещения ТСЗ СФЗ, удовлетворяющий заданным требованиям безопасности разных по степени важности КЭ объекта. Вероятность обнаружения нарушителя на путях проникновения к цеху 2 и складу готовой продукции, которая составила  $P_0(A) \geq 0,9$ , а к цеху 1 —  $P_0(A) \geq 0,95$ , удовлетворяет заданным требованиям при минимальной стоимости размещения ТСЗ. Кроме того, расположение ТСЗ на путях проникновения нарушителя соответствует заданным требованиям вероятности своевременного прибытия сил охраны  $P_{СП}(T/A) \geq 0,8$ . Таким образом, размещение ТСЗ обеспечивает заданные требования безопасности критически важного объекта.

#### Список литературы

1. Мишин Е. Т., Соколов Е. Е. Построение систем физической защиты потенциально опасных объектов. М.: Радио и связь, 2005. 200 с.
2. Загородний В. И. Комплексная защита информации в компьютерных системах: учеб. пособие. М.: Логос, 2001. 264 с.
3. Гарсия М. Проектирование и оценка систем физической защиты: пер. с англ. М.: Мир, 2002. 392 с.
4. Панин О. А. Проблемы оценки эффективности функционирования систем физической защиты объектов // Безопасность, достоверность, информация (БДИ). 2005. № 3. С. 22—27.
5. Панин О. А. Как измерить эффективность // БДИ. 2008. № 2. С. 20—24.
6. Рябинин И. А. Надежность и безопасность структурно-сложных систем. СПб.: Изд-во С.-Петербург. ун-та, 2007. 276 с.
7. Никитин В. В., Цицулин А. К. Телевидение в системах физической защиты. URL: [http://www.security-bridge.com/biblioteka/knigi\\_po\\_bezopasnosti/televidenie/ohrana/](http://www.security-bridge.com/biblioteka/knigi_po_bezopasnosti/televidenie/ohrana/)
8. Андерсон Д. А. Дискретная математика и комбинаторика: пер. с англ. М.: Вильямс, 2004. 960 с.

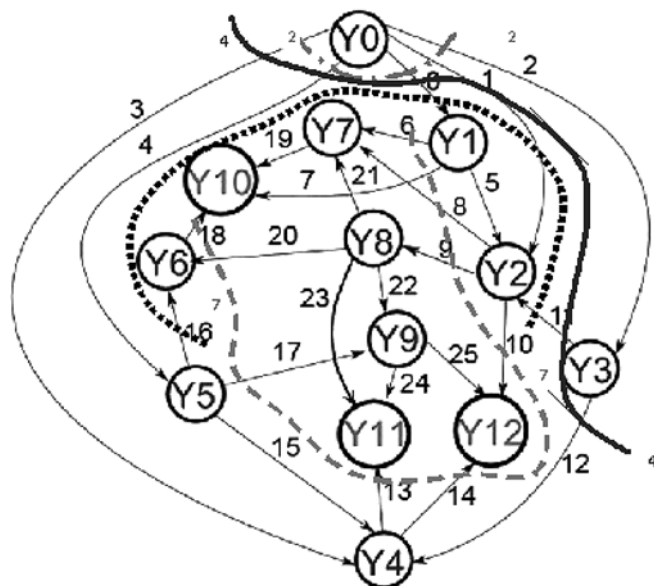


Рис. 8. Результирующие покрытия

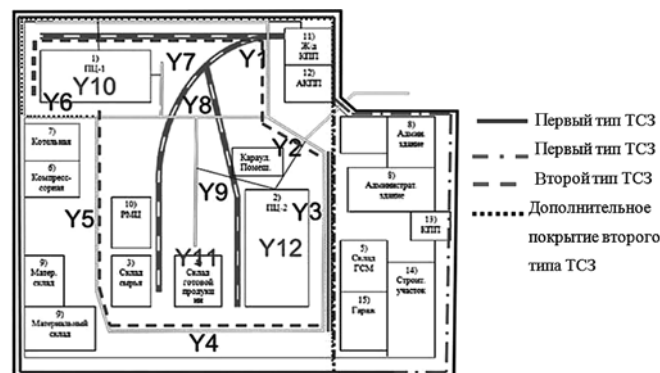


Рис. 9. План размещения ТСЗ на объекте



## Synthesis of Optimal Placement of Technical Means of the Systems of Physical Protection Critical Facilities

*The method of optimum placement of the technical means of protection (TMP) at design of the systems of physical protection (SPP) of crucial objects is developed. Basic data is the plan of object of protection with characteristics of an arrangement of buildings (elements) and the security indicators demanded to them. Feature of a task is the different required level of criterion of security of elements of object of protection which SPP has to provide.*

*On the basis of the graph of penetration of the violator, using operation of composition of a matrix of contiguity of the graph, all possible ways of penetration to object elements were formed. The received results were formed in the form of an incidence matrix: lines — ways of penetration, columns — the graph's edges. The field of a matrix — Boolean variables: one if the edge of the count is included in a movement route, zero — otherwise. It is necessary to provide counteraction to penetration of the violator on all routes of penetration. By means of a task about a covering on a matrix of incidence options of suppression of penetration of the violator (danger development) were defined. Each covering is characterized by two parameters: extent and removal from the protected object element. In the first parameter determine the cost of a covering, and by the second — probability of timely arrival of group of reaction which defines time of reaction to penetration of the violator.*

*The method of the analysis of hierarchies has determined the acceptable TMP set for formation of SPP. On the basis of a method of dynamic programming for various levels of potentially dangerous buildings (requirements to security size) the plan of optimum placement of TMP meeting the set criteria of their security was defined. Method advantage — the important principle of design of SPP — the principle of uniform strength is provided.*

**Keywords:** *the technical means of protection (TMP), system of physical protection, an incidence matrix, a task about a covering, dynamic programming, the plan of optimum placement of TMP, criterion of security of object*

### References

1. **Mishin E. T., Sokolov E. E.** *Postroenie sistem fizicheskoy zashchity potencial'no opasnykh ob'ektov*, Moscow, Radio i svyaz', 2005, 200 p.
2. **Zavgorodnij V. I.** *Kompleksnaya zashchita informacii v kom-p'yuternykh sistemah*, ucheb. posobie, Moscow, Logos, 2001, 264 p.
3. **Garsia M.** *Proektirovanie i ocenka sistem fizicheskoy zashchity* (Design and evaluation of physical protection systems): per. s angl., Moscow, Mir, 2002, 392 p.
4. **Panin O. A.** Problemy ocenki ehffektivnosti funkcionirovaniya sistem fizicheskoy zashchity ob'ektov, *Bezopasnost', dostovernost', informaciya (BDI)*, 2005, no. 3, pp. 22—27.

5. **Panin O. A.** Kak izmerit' ehffektivnost', *BDI*, 2008, no. 2, pp. 20—24.
6. **Rjabinin I. A.** Nadezhnost' i bezopasnost' strukturno-slozhnykh sistem, Sankt-Petersburg, Izd-vo S.-Peterb. Un-ta, 2007, 276 p.
7. **Nikitin V. V., Ciculin A. K.** Televidenie v sistemah fizicheskoy zashchity. URL: [http://www.security-bridge.com/biblioteka/knigi\\_po\\_bezopasnosti/televidenie/ohrana/](http://www.security-bridge.com/biblioteka/knigi_po_bezopasnosti/televidenie/ohrana/)
8. **Anderson D. A.** Diskretnaya matematika i kombinatorika (Discrete mathematics and combination theory.): per. s angl., Moscow, Vil'yams, 2004, 960 p.

### ИНФОРМАЦИЯ

3—4 Марта 2017 г. в отеле Holiday Inn Moscow Vinogradovo состоится

**4-я Международная научно-практическая конференция**

## «Инструменты и методы анализа программ, ТМРА-2017»

На конференции будут представлены приглашенные доклады признанных специалистов в области программной инженерии и конкурсные доклады, прошедшие рецензирование несколькими независимыми экспертами.

Темы, рассматриваемые на конференции, включают (но не ограничиваются):

- автоматизацию тестирования программного обеспечения
- статический анализ программ
- верификацию
- динамические методы анализа программ
- тестирование и анализ параллельных и распределенных систем
- тестирование и анализ высоконагруженных систем и систем высокой доступности
- анализ и верификацию программно-аппаратных систем
- методы создания качественного программного обеспечения
- инструментальные средства анализа, тестирования и верификации

**Подробнее:** <http://tmpaconf.org>

А. В. Скороходов, науч. сотр., e-mail: vazime@yandex.ru,  
Федеральное государственное бюджетное учреждение науки  
Институт оптики атмосферы им. В. Е. Зуева Сибирского отделения РАН, Томск

## Применение метода Виолы — Джонса для обнаружения атмосферных гравитационных волн над водной поверхностью по спутниковым снимкам MODIS

*Разработан алгоритм автоматической идентификации атмосферных гравитационных волн над водной поверхностью по спутниковым снимкам MODIS с пространственным разрешением 1000 м на основе метода Виолы — Джонса. Предложены признаки Хаара для решения задачи обнаружения квазипараллельных полос на изображениях. Обсуждаются результаты идентификации атмосферных гравитационных волн по спутниковым снимкам MODIS и Himawari-8 различных регионов планеты.*

**Ключевые слова:** атмосферные гравитационные волны, классификация, облачность, примитивы Хаара, спутниковые данные

### Введение

Атмосферные гравитационные волны (АГВ), представляющие собой вид волновых движений в устойчиво стратифицированной атмосфере Земли, считаются одним из наименее изученных явлений окружающей среды. Основными причинами этого являются отсутствие информации о периодичности их возникновения над конкретными регионами планеты и малый объем теоретических сведений о механизмах их возникновения и распространения в атмосфере Земли [1, 2]. На данный момент времени актуальными задачами данного направления являются [3, 4]:

- локализация источников генерации АГВ;
- автоматическое обнаружение их сигнатур на спутниковых снимках;
- оценка влияния рельефа подстилающей поверхности и климатических параметров на структуру облачных проявлений атмосферных волн и их динамику.

При этом предполагается, что АГВ могут воздействовать на теплообмен в системе "атмосфера — подстилающая поверхность", распространение акустических сигналов в воздушном пространстве, биологическую продуктивность экосистем и сейсмическую активность [5]. Решение задачи автоматической идентификации облачных проявлений на спутниковых снимках, которой и посвящена данная работа, является одним из начальных этапов при исследованиях АГВ.

Атмосферные волны на спутниковых снимках представляют собой группы упорядоченных квазипараллельных полос протяженностью до нескольких сотен километров, состоящих из облаков различных типов. При этом расстояние между полосами варьируется от 2 до 70 км, а общее число может достигать 50...60 [6]. Анализируя же серии спутниковых снимков GOES для исследуемых явлений, полученных с разницей в 5 мин, можно отметить ограниченность протяженности таких полос. Основными регионами появления сигнатур АГВ являются прибрежные акватории Мирового океана и внутриматериковые водоемы, обладающие значительной площадью (например, Каспийское море и о. Байкал). Однако встречаются случаи наблюдения подобных явлений и над другими типами подстилающей поверхности. В настоящее время основным способом детектирования АГВ на спутниковых снимках является их визуально-ручное дешифрирование, что в первую очередь связано с нерегулярностью их появления, изменчивостью структуры полей облачности и временем существования.

В немногих существующих работах по автоматическому обнаружению проявлений атмосферных волн используются преобразования Хафа, Радона, вейвлет-преобразования, методы подчеркивания контраста и выделения границы [7—9]. Таким образом, в указанных работах сигнатуры АГВ не идентифицируются как объекты с определенными координатами, а выполняется контрастирование

квазипараллельных полос совместно с использованием процедур шумоподавления. При этом рассмотрены только отдельные случаи наблюдения АГВ, что не дает целостного представления об эффективности предложенных алгоритмов для идентификации сигнатур атмосферных волн, имеющих различную структуру. Поэтому применение более традиционных классификаторов, для которых объект исследования характеризуется набором параметров, представляется более эффективным решением задачи обнаружения сигнатур АГВ.

Целью данной работы является разработка алгоритма автоматической идентификации облачных проявлений АГВ с различной структурой над водной поверхностью по спутниковым снимкам MODIS с пространственным разрешением 1000 м на основе применения метода Виолы — Джонса с использованием набора новых примитивов Хаара.

### Исходные данные

В работе используются данные спектрорадиометра MODIS, установленного на борту спутников Terra и Aqua, видимого диапазона спектра (0,62...0,67 мкм) с пространственным разрешением 1000 м, что позволяет надежно детектировать и проводить исследования контуров облачных образований согласно спецификациям NASA [10]. Поиск регионов с наибольшей среднегодовой повторяемостью АГВ осуществлялся путем визуального анализа спутниковых снимков прибрежных районов Мирового океана, полученных в период с 2007 по 2015 г. Общее число найденных космоснимков атмосферных волн составляет 528. При этом наиболее часто АГВ наблюдаются над акваториями Тихого океана в районе Курильских островов, Индийского океана в районах Аравийского полуострова и северо-западного побережья Австралии, а также Мадагаскарского пролива. На рис. 1 показаны изображения атмосферных волн над различными регионами планеты.

Из рис. 1 видно, что проявления АГВ имеют различную структуру, которая зависит от причин их генерации, рельефа подстилающей поверхности и состояния окружающей среды [11]. На рис. 2 показаны изменения сигнатур атмосферных волн в зависимости от размеров островов, встречающих набегающий поток воздушных масс. Примерами такого структурирования проявлений АГВ являются случаи их наблюдения над Курильскими остро-

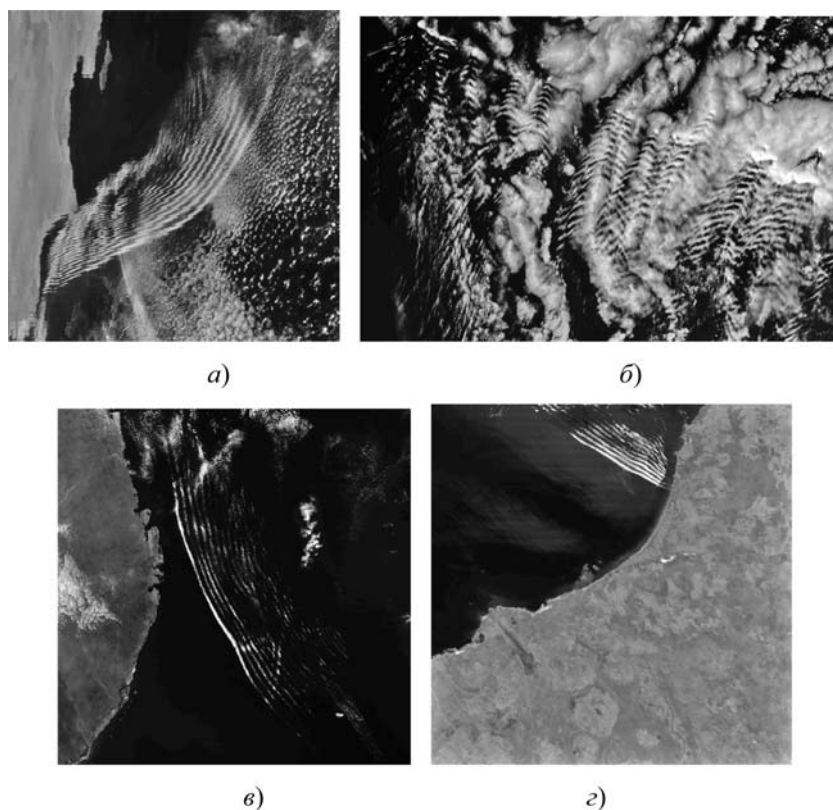


Рис. 1. Атмосферные волны над юго-восточным побережьем Сомали (UTC — 06:20, 27.09.2013 г.) (а); Курильскими островами (UTC — 00:45, 23.04.2015 г.) (б); Мозамбикским проливом (UTC — 07:25, 30.07.2015 г.) (в); северо-западным побережьем Австралии (UTC — 02:25, 17.09.2012 г.) (г)



Рис. 2. Влияние размеров островов на структуру сигнатур атмосферных волн

вами, что видно на рис. 1, б. Кроме этого, встречаются ситуации нахождения нескольких различных ориентированных сигнатур атмосферных волн с отличающимися структурами, что свидетельствует о различии источников и причин их возникновения.

### Алгоритм идентификации

На данный момент метод Виолы — Джонса является одним из самых эффективных алгоритмов бинарной классификации. Основной областью его применения является распознавание лиц на изображениях и в видеопотоке в режиме реального времени. Основными преимуществами данного метода являются его быстроедействие, возможность контролировать ошибки первого и второго рода на этапе обучения классификатора, устойчивость к шуму, поворотам и масштабированию изображений.

Главными же недостатками метода являются длительная процедура его обучения и большой объем генерируемых "слабых" классификаторов [12]. Примерами успешного применения алгоритма Виолы — Джонса являются: локализация номерных и дорожных знаков, идентификация текста, обнаружение различных объектов на спутниковых и аэрофотоснимках и ряд других задач [13—15].

Алгоритм идентификации облачных проявлений АГВ по спутниковым снимкам MODIS с пространственным разрешением 1000 м на основе метода Виолы — Джонса состоит из следующих этапов:

1. Вычисляется интегральное представление исследуемого полноразмерного изображения из космоса размером  $2030 \times 1354$  пикселя.

2. Выполняется сканирование скользящим окном спутникового снимка заданного размера, со смещением в 1 пиксель по горизонтали и вертикали.

3. Для каждого положения скользящего окна рассчитывается набор значений классификационных признаков Хаара.

4. Рассчитанные характеристики предъявляются на вход каскадного классификатора, который принимает решение о наличии в исследуемом фрагменте облачного проявления АГВ.

5. Центральный пиксель правильно проклассифицированного фрагмента исследуемого изображения помечается.

6. После окончания сканирования скользящее окно масштабируется на 10 пикселей по вертикали и горизонтали, и обработка изображения начинается заново со 2-го этапа.

7. Выполняется фильтрация результатов классификации, целью которой является выделение облачных проявлений АГВ на спутниковом снимке рамкой наибольшего размера.

Рассмотрим этапы приведенного выше алгоритма более подробно. Интегральное представление изображения позволяет эффективно вычислять

суммарную яркость прямоугольника произвольного размера

$$D(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j),$$

где  $D$  — значение точки интегрального представления;  $I$  — яркость пикселя исходного изображения. Начальный размер скользящего окна определен экспериментальным путем и равен  $40 \times 40$  пикселей для уменьшения числа ложных срабатываний алгоритма на участках спутниковых снимков, относящихся к слоисто-кучевым волнистообразным облакам, которые имеют сходную текстуру изображений с проявлениями атмосферных волн. Предельный же размер окна ограничен  $200 \times 200$  пикселей в соответствии с максимально возможным числом наблюдаемых квазипараллельных полос АГВ и учетом расстояния между ними.

Классификационными характеристиками в методе Виолы — Джонса являются признаки Хаара — наборы изображений различно ориентированных точек и линий. На рис. 3, *a* показаны классические примитивы Хаара, используемые в методе Виолы — Джонса, например, для обнаружения лиц на изображениях и в видеопотоке. На рис. 3, *б* представлены некоторые классификационные характеристики, предложенные нами в целях повышения эффективности детектирования квазипараллельных полос облачных проявлений атмосферных волн на спутниковых снимках. Использование новых признаков предполагает незначительное увеличение времени работы алгоритма, которое компенсируется применением технологии Nvidia CUDA для параллельной обработки множества положений скользящего окна. Отметим, что общее число предложенных нами классификационных характеристик равно 36. При этом численное выражение признаков Хаара имеет следующий вид

$$f = \sum_{\forall I=0} I - \sum_{\forall I=255} I,$$

где  $\sum_{\forall I=0} I$  — сумма яркостей в области

черных прямоугольников;  $\sum_{\forall I=255} I$  —

в области белых. Интегральное же представление изображений позволяет эффективно вычислять эти суммы. С учетом местоположения признаков в исследуемом фрагменте изображения суммарное число их рассчитанных значений может достигать нескольких тысяч.

В работе используется каскадный классификатор, который также может быть представлен и в виде дерева решений. Элементом каскада являются сильные (результатирующие) классифи-

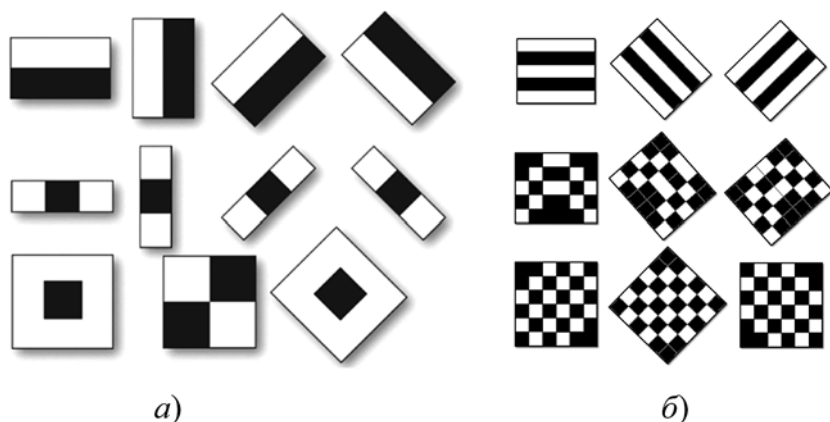


Рис. 3. Классические примитивы Хаара (*a*) и их модифицированные аналоги (*б*)

каторы, представляющие собой композицию наилучших слабых (простых) классификаторов. Решающее правило сильного классификатора задается следующим выражением:

$$H(z) = \begin{cases} 1, & \sum_{t=1}^T a_t h_i^{(t)}(z) \geq \frac{1}{2} \sum_{t=1}^T a_t; \\ 0, & \text{в остальных случаях;} \end{cases}$$

$$a_t = \log \frac{1}{b_t}, \quad b_t = \frac{E_i(t)}{1 - E_i(t)},$$

здесь  $E_i(t)$  — ошибка наилучшего слабого классификатора  $h_i(z)$  на шаге  $t$ , имеющего вид

$$h_i(z) = \begin{cases} 1, & p_i f_i(z) < p_i \theta_i; \\ 0, & \text{в остальных случаях,} \end{cases}$$

где  $p_i$  определяет направление знака неравенства;  $z$  — область (местоположение) расчета признака Хаара  $f_i$  в исследуемом фрагменте изображения, а  $\theta_i$  — пороговое значение признака.

Таким образом, слабый классификатор по своей сути является пороговым решающим правилом. Каскадный классификатор, как правило, состоит из 10...30 сильных классификаторов. В данной работе число элементов каскада равнялось 20, что позволяет избежать переобучения. Фрагмент спутникового снимка считается правильно идентифицированным (является изображением облачного проявления АГВ или его части), если хотя бы один из сильных классификаторов дал положительный ответ. Таким образом, время обработки различных положений скользящего окна может варьироваться.

Результатом работы каскадного классификатора является набор помеченных пикселей, как принадлежащих изображениям облачных проявлений АГВ. Предложенная нами процедура фильтрации результатов классификации заключается в формировании групп (кластеров) близко расположенных друг к другу пикселей на основе евклидова расстояния. Далее находятся центры кластеров, и соответствующая группа пикселей выделяется рамкой. Фрагменты, выделенные рамкой меньшего, чем  $40 \times 40$  пикселей размера, отбраковываются, что позволяет игнорировать ложные срабатывания каскадного классификатора. Таким образом, результатом работы алгоритма идентификации является спутниковый снимок с отмеченными фрагментами изображений, принадлежащих облачным проявлениям АГВ.

Метод Виолы — Джонса относится к классу алгоритмов, обучающихся с "учителем". Соответственно для его настройки требуются обучающая выборка, содержащая наборы образцов двух классов (бинарная классификация). Первый состоит из

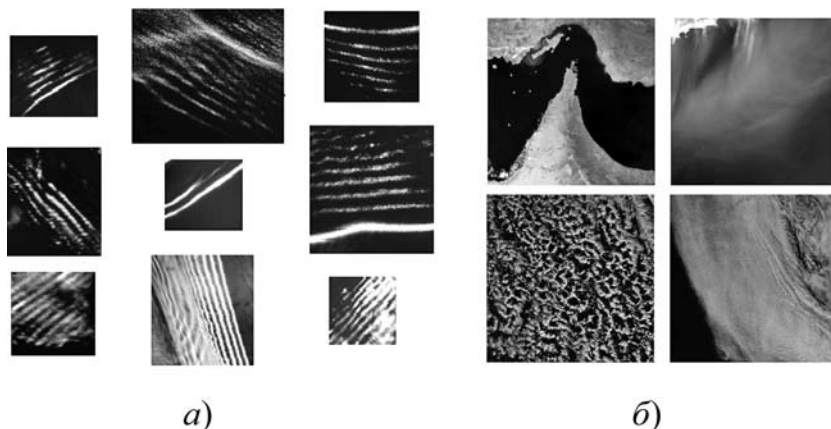


Рис. 4. Сигнатуры облачных проявлений АГВ (а) и фрагменты спутниковых снимков без атмосферных волн (б)

характерных изображений объекта классификации — фрагментов изображений произвольного размера атмосферных волн, показанных на рис. 4, а. Второй же содержит фрагменты спутниковых снимков MODIS размером  $400 \times 400$  пикселей, на которых облачные проявления АГВ отсутствуют (рис. 4, б). Общее число сигнатур атмосферных волн в обучающей выборке равно 912, а произвольных изображений — 403 фрагмента. Каскадный классификатор обучается алгоритмом бустинга, который основан на "усилении" слабых классификаторов. Обучение заключается в поиске слабых классификаторов с наименьшими значениями целевой функции ошибки  $E_i(t)$  и создании на их основе результирующих (сильных) классификаторов, из которых формируется каскад.

### Результаты идентификации атмосферных волн

Разработанный в рамках данной работы алгоритм идентификации был апробирован на основе тестовой выборки, состоящей более чем из 300 спутниковых снимков MODIS с пространственным разрешением 1000 м, полученных в период с 2007 по 2015 г. Поскольку на одном изображении могут находиться несколько несвязанных друг с другом облачных проявлений атмосферных волн, то общее число тестовых участков составило  $N_T = 385$ . Качество обнаружения оценивалось на основе вероятности правильной классификации тестовой выборки

$$E_C = \frac{N_R}{N_T},$$

где  $N_R$  — число правильно идентифицированных АГВ. При этом не учитывались ситуации, когда алгоритм выделял фрагменты проявлений АГВ, неучтенные (пропущенные) при визуальном ручном дешифрировании тестовых снимков. Таким образом, оценка вероятности правильной идентификации  $E_C = 0,87$  при  $N_R = 337$ .

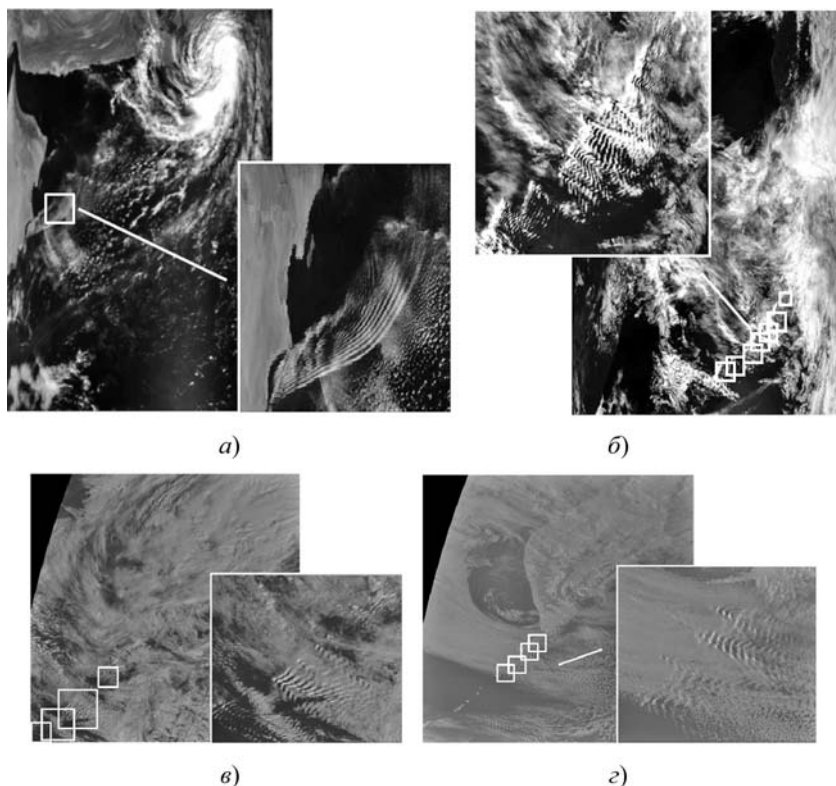


Рис. 5. Результаты идентификации проявлений АГВ над побережьем Сомали (UTC — 06:20, 27.09.2013 г.) (а) и их каскадов над Курильскими островами по снимкам MODIS (UTC — 01:20, 25.06.2011 г.) (б) и Himawari-8 (UTC — 07:37:30, 01.10.2015 г.) (в); (UTC — 15:00:00, 18.03.2016 г.) (г)

На рис. 5, а и б показаны результаты идентификации АГВ над побережьем Сомали от 27.09.2013 г. и их каскадов над Курильскими островами от 25.06.2011 г. соответственно. Следует отметить, что волновой процесс, изображенный на рис. 5, а, хорошо контрастирует с подстилающей поверхностью и слоистообразной облачностью, а на рис. 5, б — сливается с плотной облачностью. Однако на рис. 5, а настройки фильтрации позволили выделить только центральную часть облачного проявления АГВ, а на рис. 5, б — весь каскад атмосферных волн, формируемый группой островов. Отметим, что текстура облачных проявлений АГВ, показанных на рис. 5, а и б, различна. Результаты же классификации всей тестовой выборки подтверждают, что предложенный алгоритм позволяет идентифицировать данные явления с различной структурой, в том числе и на фоне плотной облачности.

Разработанный алгоритм идентификации был апробирован и на других данных дистанционного зондирования Земли из космоса. На рис. 5, в и г показаны результаты обнаружения проявлений АГВ над Курильскими островами на спутниковых снимках Himawari-8 видимого диапазона спектра (0,615...0,675 мкм) с пространственным разрешением 1000 м. Данная система космического базирования является геостационарной и наряду со

спутниками серии GOES и MSG позволяет снимать целиком всю полусферу Земли с разрешением 1000 м и менее в зависимости от спектрального диапазона. Однако в отличие от последних периодичность съемки Himawari-8 составляет 10 мин для полусферы и 2,5 мин для территории Японии, что делает его перспективным инструментом для исследования динамики облачного покрова и его свойств над территориями Сибири и Дальнего Востока России.

## Заключение

Предложенный алгоритм позволяет эффективно идентифицировать облачные проявления АГВ с различной структурой на спутниковых снимках MODIS видимого диапазона спектра с пространственным разрешением 1000 м, в том числе и на фоне оптически плотной облачности. Это подтверждается результатами обнаружения исследуемых явлений на изображениях тестовой выборки, полученных за продолжительный период времени. В отличие от методов контрастирования параллельных прямых, разработанный алгоритм позволяет идентифицировать проявления атмосферных волн как локализованные объекты.

При этом вероятность правильных обнаружений подобных явлений 0,87 на изображениях тестовой выборки свидетельствует о хорошей обобщающей способности предложенного алгоритма. Эффективность детектирования квазипараллельных полос атмосферных волн достигается за счет применения предложенных нами признаков Хаара, примеры которых показаны на рис. 3, б. При этом наблюдается незначительное увеличение длительности процедуры идентификации, которое компенсируется применением технологии Nvidia CUDA. Использование же скользящего окна с размерами  $40 \times 40$  пикселей и применение предложенной нами процедуры фильтрации результатов идентификации позволяет избежать ложных срабатываний метода на участках спутниковых снимков, принадлежащих слоисто-кучевым волнистообразным облакам.

Алгоритм идентификации АГВ успешно апробирован и на данных геостационарного спутника Himawari-8, что делает его применение возможным для оперативного мониторинга данных явлений над целой полусферой Земли благодаря быстрдействию метода Виолы — Джонса и технологии Nvidia CUDA. Перспективными направлениями данной работы можно считать дальнейшее исследование характеристик атмосферных волн в целях поиска источников их генерации, оценки их влияния на

окружающую среду, прогноза их возникновения над определенными территориями.

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 16-37-60019 мол\_а\_дк.

#### Список литературы

1. Zilitinkevich S. S., Elperin T., Kleorin N., L'vov V., Rogachevskii I. Energy and flux-budget turbulence closure model for stably stratified flows. Part II: The role of internal gravity waves // *Boundary—Layer Meteorology*. 2009. Vol. 133, N. 2. P. 139—164.
2. Largeron Y., Staquet C., Chamel C. Characterization of oscillatory motion in the stable atmosphere of a deep valley // *Boundary—Layer Meteorology*. 2013. Vol. 148, N. 2. P. 439—459.
3. Алексанин А. И., Ким В. Автоматическое обнаружение внутренних волн на спутниковых изображениях и оценка плотности перемешанного слоя // *Исследование Земли из космоса*. 2015. № 1. С. 44—52.
4. Alpers W., Huang W. On the discrimination of radar signatures of atmospheric gravity waves and oceanic internal waves on synthetic aperture radar images of the sea surface // *IEEE Transactions on Geoscience and Remote Sensing*. 2011. Vol. 49, N. 3. P. 1114—1126.
5. Кашкин В. Б. Внутренние гравитационные волны в тропосфере // *Оптика атмосферы и океана*. 2013. Т. 26, № 10. С. 908—916.
6. Вельтишев Н. Ф., Степаненко В. М. Мезометеорологические процессы. М.: МГУ, 2006. 101 с.
7. Rodenas J., Garello R. Internal wave detection and location in SAR images using wavelet transform // *IEEE Transact. Geosci. and Rem. Sens.* 1998. Vol. 36. P. 1494—1507.
8. Simonin D., Tatnall A. R., Robinson I. S. The automated detection and recognition of internal waves // *Int. J. Rem. Sens.* 2009. Vol. 30. P. 4581—4598.
9. Jackson C. Internal wave detection using the Moderate Resolution Imaging Spectroradiometer (MODIS) // *J. Geophys. Res.* 2007. Vol. 112. P. 11012, doi: 10.1029/2007JC004220.
10. MODIS-Atmosphere [electronic resource] / U. S. Government Computer System, 2000 — 2016. URL: <http://modis-atmos.gsfc.nasa.gov> (access data 25.08.2016).
11. Иванов А. Ю. О распознавании поверхностных проявлений океанских внутренних волн и атмосферных гравитационных волн на радиолокационных изображениях морской поверхности // *Исследование Земли из космоса*. 2011. № 1. С. 70—85.
12. Oxozle. Разработка и исследование метода распознавания объектов на базе алгоритма Виолы — Джонс, 2013—2016. URL: [http://oxozle.com/files/builtin/papers/azarov\\_development\\_and\\_research\\_of\\_object\\_recognition\\_method\\_based\\_on\\_viola\\_jones\\_algorithm.pdf](http://oxozle.com/files/builtin/papers/azarov_development_and_research_of_object_recognition_method_based_on_viola_jones_algorithm.pdf) (дата обращения 25.08.2016)
13. Амосов О. С., Иванов Ю. С. Модифицированный алгоритм локализации номерных знаков транспортных средств на основе метода Виолы—Джонса // *Информатика и системы управления. Автоматизированные системы и комплексы*. 2014. Т. 39, № 1. С. 127—140.
14. Чигорин А. А., Конушин А. С. Эксперименты с обучением методов распознавания дорожных знаков на синтетических данных // *Наука и образование*. 2013. № 8. С. 315—324.
15. Буй Т. Т. Ч., Фан Н. Х., Спицын В. Г. Распознавание лиц на основе применения метода Виолы — Джонса, вейвлет-преобразования и метода главных компонент // *Известия Томского политехнического университета*. 2012. Т. 320. № 5. С. 54—59.

A. V. Skorokhodov, Researcher, e-mail: vazime@yandex.ru,

V. E. Zuev Institute of Atmospheric Optics of the Siberian Branch of the Russian Academy of Sciences

## Application of Viola — Jones Method for Atmospheric Gravity Waves Detection above the Water Surface on Modis Satellite Imagery

*The purpose of this work is to develop an algorithm for automatic identification of cloud views of atmospheric gravity waves above the water surface on MODIS satellite imagery with a spatial resolution of 1000 m based on the Viola — Jones method. The coastal regions of the oceans with the highest repeatability of the researched events in the period from 2007 to 2015 (the coast of the Arabian Peninsula, the Madagascar Strait, the western coast of Australia, the Kurile Islands) are determined. Different views of atmospheric gravity waves on MODIS satellite imagery are shown. Paper describes Viola — Jones method, its advantages and disadvantages. Training and background samples for learning of the research events identification algorithm are formed. Proposed modified Haar features for detecting of quasi-parallel lanes of atmospheric waves views on satellite imagery. The initial size of "sliding" window at this work is  $40 \times 40$  pixels to reduce algorithm false positives at the satellite imagery fragments of stratocumulus undulatus clouds. The results of atmospheric gravity waves identification above water surface on MODIS and Himawari-8 satellite imagery with a spatial resolution of 1000 m are presented. The estimate of the correct classification probability of the test samples (more than 300 fragments of MODIS imagery) equals 0,87. The proposed algorithm can be used in operational monitoring of atmospheric waves because it allows to efficiently identifying cloud views of the research events with different structure on the background of the optically thick cloudiness using Nvidia CUDA technologies. The results of atmospheric waves identification on satellite imagery are discussed. The perspective development directions of this work are presented, which include the development of automatic methods to search for their generation sources, the estimation of their impact on the environment and the prediction of their occurrence over a certain territory.*

**Keywords:** atmospheric gravity waves, classification, cloudiness, Haar's features, satellite data

#### References

1. Zilitinkevich S. S., Elperin T., Kleorin N., L'vov V., Rogachevskii I. Energy and flux-budget turbulence closure model for stably stratified flows. Part II: The role of internal gravity waves, *Boundary—Layer Meteorology*, 2009, vol. 133, no. 2, pp. 139—164.
2. Largeron Y., Staquet C., Chamel C. Characterization of oscillatory motion in the stable atmosphere of a deep valley, *Boundary—Layer Meteorology*, 2013, vol. 148, no. 2, pp. 439—459.
3. Aleksanin A. I., Kim V. Avtomaticheskoe obnaruzhenie vnutrennih voln na sputnikovyh izobrazhenijah i ocenka plotnosti pere-meshannogo sloja (Automatic internal waves detection on satellite

imagery and estimate of mixed layer density), *Issledovanie Zemli iz kosmosa*, 2015, no. 1, pp. 44–52.

4. **Alpers W., Huang W.** On the discrimination of radar signatures of atmospheric gravity waves and oceanic internal waves on synthetic aperture radar images of the sea surface, *IEEE Transactions on Geoscience and Remote Sensing*, 2011, vol. 49, no. 3, pp. 1114–1126.

5. **Kashkin V. B.** Vnutrennie gravitacionnye volny v troposfere (Internal gravity waves in the troposphere), *Optika atmosfery i okeana*, 2013, vol. 26, no. 10, pp. 908–916.

6. **Vel'tishhev N. F., Stepanenko V. M.** Mezometeorologicheskie process (Mesometeorological processes), Moscow, MGU, 2006. 101 p.

7. **Rodenas J., Garello R.** Internal wave detection and location in SAR images using wavelet transform, *IEEE Transact. Geosci. and Rem. Sens.*, 1998, vol. 36, pp. 1494–1507.

8. **Simonin D., Tatnall A. R., Robinson I. S.** The automated detection and recognition of internal waves, *Int. J. Rem. Sens.*, 2009, vol. 30, pp. 4581–4598.

9. **Jackson C.** Internal wave detection using the Moderate Resolution Imaging Spectroradiometer (MODIS), *J. Geophys. Res.*, 2007, vol. 112, pp. 11012, doi: 10.1029/2007JC004220.

10. **MODIS-Atmosphere.** *U. S. Government Computer System, 2000–2016.* URL: <http://modis-atmos.gsfc.nasa.gov> (access data 25.08.2016).

11. **Ivanov A. Ju.** O raspoznavanii poverhnostnyh proyavlenij okeanskih vnutrennih voln i atmosferyh gravitacionnyh voln na ra-

diolokacionnyh izobrazhenijah morskoy poverhnosti (About detection of ocean internal waves and atmospheric gravity waves manifestations on water surface radar imagery), *Issledovanie Zemli iz kosmosa*, 2011, no. 1, pp. 70–85.

12. **Oxozle** [electronic resource]. *Razrabotka i issledovanie metoda raspoznavaniya ob#ektov na baze algoritma Violy — Dzhons* (Development and research of objects recognition method based on the Viola-Jones algorithm), 2013–2016. URL: [http://oxozle.com/files/builtin/papers/azarov\\_development\\_and\\_research\\_of\\_object\\_recognition\\_method\\_based\\_on\\_viola\\_jones\\_algorithm.pdf](http://oxozle.com/files/builtin/papers/azarov_development_and_research_of_object_recognition_method_based_on_viola_jones_algorithm.pdf) (access data 25.08.2016).

13. **Amosov O. S., Ivanov Ju. S.** Modificirovannyj algoritm lokalizatsii nomernykh znakov transportnykh sredstv na osnove metoda Violy—Dzhonsa (Modified algorithm of license plates vehicles localization based on the Viola-Jones method), *Informatika i sistemy upravleniya. Avtomatizirovannye sistemy i eompleksy*, 2014, vol. 39, no. 1, pp. 127–140.

14. **Chigorin A. A., Konushin A. S.** Jeksperimenty s obucheniem metodov raspoznavaniya dorozhnykh znakov na sinteticheskikh dannykh (Experiments with the training of traffic signs recognition methods on synthetic data), *Nauka i obrazovanie*, 2013, no. 8, pp. 315–324.

15. **Buj T. T. Ch., Fan N. H., Spicyn V. G.** Raspoznavanie lic na osnove primeneniya metoda Violy-Dzhonsa, vejvlet-preobrazovaniya i metoda glavnykh komponent (Face recognition by applying the Viola-Jones method, the wavelet-transform and principal component analysis), *Izvestija Tomskogo politehnicheskogo universiteta*, 2012, vol. 320, no. 5, pp. 54–59.

УДК 004.652.4(045)

DOI:10.17587/it.23.56-61

**Ю. В. Полищук**, канд. техн. наук, доц. кафедры компьютерной безопасности и математического обеспечения информационных систем, ФГБОУ ВО "Оренбургский государственный университет" [youra\\_polishuk@bk.ru](mailto:youra_polishuk@bk.ru)

## О способе мониторинга информационной энтропии больших технических систем

*Рассмотрен способ мониторинга информационной энтропии больших технических систем, который заключается в автоматизированном контроле количественной оценки информационной энтропии системы и позволяет сделать вывод о возможности принятия управленческих решений.*

**Ключевые слова:** энтропия системы, квазиструктурированная информация, обработка электронных документов, большие технические системы

### Введение

Академик Б. Н. Петров утверждал, что информационная энтропия характеризует неопределенность управления, т. е. его качество [1]. Таким образом, потеря информации при принятии управленческих решений не должна превышать определенного значения. В настоящей работе рассмотрен способ мониторинга информационной энтропии больших технических систем (БТС).

Принятие управленческих решений для БТС реализует группа лиц, принимающих решения (ГЛПР) на основе анализа фактографических данных, получаемых из сопутствующего эксплуатационного контента (СЭК). Необходимость участия

ГЛПР в процессе принятия решений продиктована масштабом управляемой системы и сложностью процесса управления.

Сложность БТС, а также трудоемкость получения и обработки фактографических данных СЭК вызывают расхождение реального состояния управляемой системы и состояния, отраженного в СЭК. Последнее способствует росту информационной энтропии БТС и принятию ГЛПР некорректных управленческих решений. Таким образом, принятие корректных управленческих решений для БТС допустимо только в случае соблюдения условия о максимально допустимом значении информационной энтропии [2].



## Постановка задачи

Фактографические данные, характеризующие состояние БТС, поступают в СЭК в виде импульсных информационных потоков, которые представлены электронными документами с квазиструктурированным информационным наполнением, т. е. информацией, в которой можно выделить некую структуру, которая заранее целиком или частично неизвестна, либо может меняться с течением времени [3].

По каждому информационному потоку поступает определенная фактографическая информация, представленная набором документов, как правило, через фиксированный интервал времени, который будем называть шагом импульса информационного потока [2]. Шаг импульса информационного потока может изменяться во время эксплуатации БТС как по требованию ГЛПР, так и в связи с появлением нестандартных эксплуатационных ситуаций, таких например, как внеплановый ремонт, авария или нехватка персонала.

Получаемые из СЭК фактографические данные характеризуют состояние различных параметров БТС в определенный момент времени. Выделяют две основные категории параметров: статические и динамические. Для статических параметров время их получения от БТС не важно, так как их значения не меняются во времени. Динамические параметры характеризуются "устареванием", т. е. расхождением значения хранимого параметра и реального значения данного параметра в настоящий момент, что обусловлено изменениями значения параметра во времени. В зависимости от вида динамического параметра ГЛПР формирует функцию "устаревания" для данного параметра в зависимости от времени его существования.

На этапе проектирования БТС определяется такое количество информационных потоков фактографических параметров с заданными шагами импульсов, чтобы их знание с учетом их функций "устаревания" было достаточным для принятия корректных управленческих решений в любой момент времени. На практике при эксплуатации БТС возникает расхождение реального состояния управляемой системы и состояния, отраженного в СЭК. Данное расхождение затрудняет или делает невозможным для ГЛПР идентификацию реального состояния БТС в момент принятия решения.

Вероятность возможных состояний для БТС снижается при увеличении числа известных характеризующих систему параметров, а информационная энтропия определяется как логарифм отношения всех известных параметров к их теоретическому числу с учетом весомости данных параметров, взятый со знаком минус [2]:

$$H_{\text{энтр}} = -\log_2 \frac{F_{\text{ст}}(P_{\text{ст}}, K_{\text{ст}}) + F_{\text{дин}}(P_{\text{дин}}, F_{\text{уст}}, K_{\text{дин}})}{F_{\text{ст}}(GP_{\text{ст}}(T_{\text{эксп}}, K_{\text{ст}}) + F_{\text{дин}}(GP_{\text{дин}}(T_{\text{эксп}}, F_{\text{уст}}, K_{\text{дин}}))}, \quad (1)$$

где  $F_{\text{ст}}, F_{\text{дин}}$  — функции, вычисляющие суммы произведений мощностей множеств статических и динамических (с учетом функции "устаревания") параметров на соответствующие весовые коэффициенты;  $P_{\text{ст}}, P_{\text{дин}}$  — множества известных соответственно статических и динамических (с временем их существования) параметров системы;  $K_{\text{ст}}, K_{\text{дин}}$  — множества весовых коэффициентов, соответствующих статическим и динамическим параметрам;  $GP_{\text{ст}}, GP_{\text{дин}}$  — функции, генерирующие множества теоретически возможных соответственно статических и динамических параметров системы за указанный период;  $T_{\text{эксп}}$  — время эксплуатации системы.

## Практическая реализация

В качестве примера выполним оценку информационной энтропии коллекторно-лучевой системы сбора (КСС) продукции газоконденсатного месторождения. Данная разновидность систем сбора продукции применяется в Оренбургском газоконденсатном месторождении (ОГКМ), которое эксплуатируется с 1974 г. и в данный момент находится на этапе падающей добычи.

КСС состоит из скважин, которые рассматриваются как отдельные системы, их контуры влияния могут пересекаться, а движение продукции по шлейфу в некоторых случаях приводит к "задавливанию" скважин. Таким образом, границами системы являются газоносный пласт и блок входных нитей (БВН).

Рассмотрим КСС совместно с насосно-компрессорными трубами (НКТ) и контурами влияния скважин. На рис. 1 показана схема наиболее распространенной в условиях ОГКМ КСС, обеспечивающей сбор продукции с трех скважин.

Продукция поступает к забою скважины из пласта, где находится под пластовым давлением  $P_{\text{пл}}$ . При движении к забою скважины происходит снижение давления до значения забойного давления  $P_3$ . Таким образом, в границах контура влияния сква-

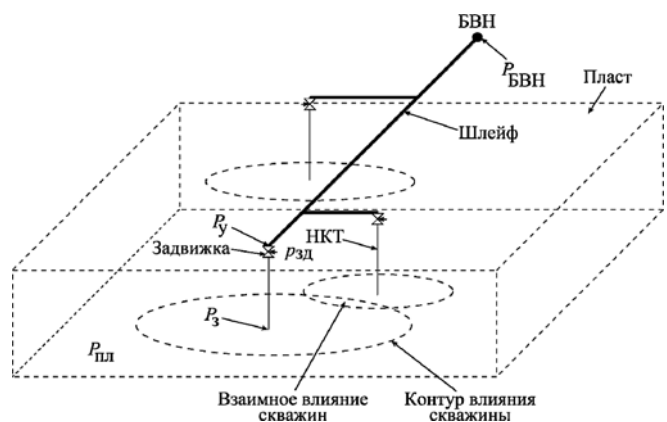


Рис. 1. Схема коллекторно-лучевой системы сбора продукции с трех скважин

жины образуется депрессионная воронка [4]. Далее продукция попадает в НКТ, по которым поднимается от забоя скважины до КСС, где на входе установлены регулирующие задвижки с местным сопротивлением  $P_{зд}$ , снижающие давление до значения устьевого давления  $P_y$ . При перемещении продукции по КСС давление в шлейфах падает, а в тройниках выравнивается по двум направлениям. При дальнейшем движении по шлейфу падение давления позволяет продукции попадать на БВН с давлением, заданным технологическим режимом.

Границы рассматриваемой системы определены известными заданными статическими давлениями газа: пластовое  $P_{пл}$  — геологией продуктивного пласта, а на блоке входных нитей  $P_{БВН}$  — технологическим режимом эксплуатации месторождения. Таким образом, внутренней характеристикой рассматриваемой системы является давление газа [5]. Установив его на границах участков КСС, найдем все ее параметры.

Основными элементами системы, кроме пласта и БВН, являются забой скважин, НКТ, соединительные трубопроводы (нити) и тройники. Пласт, забой скважины, БВН и тройники характеризуются значениями статических давлений, а трубопроводы и НКТ — падениями статических давлений между границами участков, которые определены постоянными значениями множества конструктивно-технологических параметров.

Потери статического давления при движении потока от газоносного пласта до БВН происходят в трех зонах: призабойной зоне скважин, при подъеме газа на поверхность и трении его о стенки ствола, в трубопроводе [11]. Для принятия корректных управляющих воздействий на КСС должны быть установлены все конструктивные параметры скважин и шлейфов из состава системы, например, внутренние диаметры труб, рельеф шлейфов и т. д. Несмотря на длительный период эксплуатации месторождения, перечисленные параметры известны в полном объеме не для всех КСС из состава ОГКМ. Эти параметры относятся к категории статических и характеризуют полноту знаний о состоянии системы.

В качестве динамических параметров данной системы, определяющих полноту знаний о ней, рассмотрим пластовые, забойные и устьевые давления, а также значения дебитов газа, конденсата и воды для всех скважин из состава КСС. Значения пластовых и забойных давлений ГЛПР получают из результатов гидродинамических исследований скважин (ГИС), проведение которых дает понимание состояния скважин. В условиях ОГКМ ГИС должны выполняться не реже одного раза в квартал для каждой скважины из состава КСС. Значения дебитов и устьевых давлений принимают по результатам геолого-технологического отчета, который в условиях ОГКМ формируется в конце каждого месяца. В качестве функции "устаревания" для всех

динамических параметров используем экспоненциальную зависимость от времени их существования, на основе которой строим кривые падения главных эксплуатационных показателей ОГКМ. Масштаб функции "устаревания" зависит от скорости изменения состояния конкретной БТС.

Для КСС функцию "устаревания" запишем в следующем виде:

$$F_{уст} = \exp(-\sqrt{T}), \quad (2)$$

где  $T$  — число календарных месяцев с момента получения значения динамического параметра.

Таким образом, если значение динамического параметра получено в текущем месяце ( $T = 0$ ), то функция "устаревания" для него  $F_{уст} = 1$ . Снижение значимости для динамических параметров БТС учитывается путем умножения значения веса параметра на значение функции "устаревания".

Стоит отметить, что в условиях эксплуатации БТС фактическое известное число параметров системы, как правило, меньше теоретического. Например, как уже говорилось выше, в условиях ОГКМ ГИС необходимо выполнять не реже одного раза в квартал для каждой скважины из состава КСС, но требование обеспечения заданного объема добычи газа на месторождении не позволяет проводить данные исследования в полном объеме, так как на время проведения исследований от скважины не поступает продукция. Конструктивные параметры КСС ОГКМ также известны не в полном объеме.

Для реализации системы мониторинга информационной энтропии КСС необходимо сформировать единый СЭК для каждой системы.

Фактографический контент СЭК будет квазиструктурированным, так как несмотря на схожесть фактографических данных по КСС для каждой КСС СЭК будет формироваться с учетом ее особенностей.

Таким образом, для реализации СЭК подходит язык XML, так как он обеспечивает возможность реализации моделей для квазиструктурированных данных [6].

Используем способ синтеза квазиструктурированных моделей, описанный в работе [7], для построения модели СЭК КСС. В результате получим модель, показанную на рис. 2.

Модель СЭК КСС состоит из необязательного сегмента scheme — описание конструкции трубо-

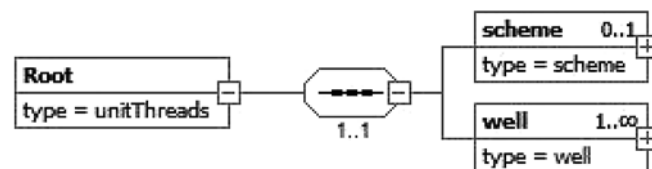


Рис. 2. Структура базового сегмента модели СЭК КСС

проводов КСС, и сегмента well — информация о работе скважин, входящих в состав КСС. Под сегментом модели следует понимать объект модели, в состав которого могут входить вложенные объекты и сегменты.

Сегмент scheme состоит из неограниченного числа сегментов thread — нитей трубопровода (рис. 3).

Так как конструкция трубопроводов КСС имеет древовидную структуру, сегмент thread включает в себя следующие объекты (рис. 4): обязательные объекты id — номера нити; parentId — ссылка на номер нити, с которой выполнена врезка, diameter — внутренний диаметр трубы; wallThickness — толщина трубы; x — координата широты на конце; y — координата долготы на конце; необязательный объект z — координата высоты на конце. Объект z является необязательным, так как рельеф КСС в условиях ОГКМ определен не у всех систем сбора.

В состав сегмента well входят обязательный объект number — номер скважины, обязательный сег-

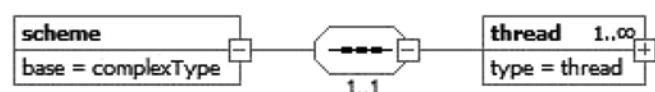


Рис. 3. Структура сегмента scheme

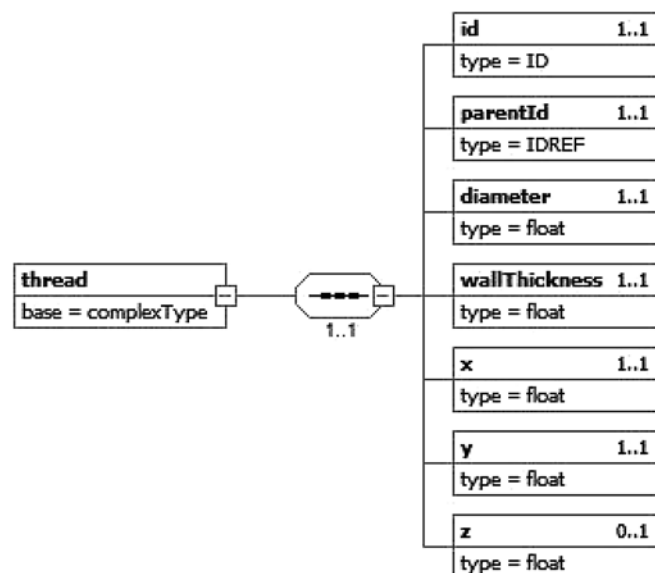


Рис. 4. Структура сегмента thread

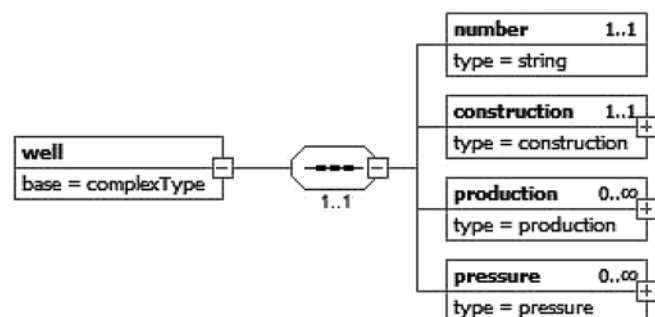


Рис. 5. Структура сегмента well

мент construction — информация о конструкции скважины и два необязательных сегмента, которые могут быть использованы неограниченное число раз, это production — информация о продукции скважины и pressure — информация об измерениях давлений в скважине (рис. 5).

Сегмент construction состоит из обязательного объекта idThread — ссылка на нить, к которой подключена скважина, и необязательных объектов depth — глубина скважины, data — прочая информация о конструкции скважины (рис. 6).

В условиях ОГКМ скважины имеют различную конструкцию, поэтому было принято решение выделить только глубину скважины, а остальные характеристики хранить в виде неформализованных текстовых данных.

В состав сегмента production входят обязательные объекты: date — дата получения продукции (в условиях ОГКМ за календарный месяц); naturalGas — количество добытого газа; condensate — количество добытого конденсата; wellheadPressure — значение устьевого давления; workingDays — число отработанных дней в месяце, и необязательный объект water — количество добытой воды (рис. 7).

Сегмент pressure состоит из обязательного объекта date — дата выполнения замера и двух возможных вариантов давлений: reservoirPressure — значение пластового давления в скважине; bottom-

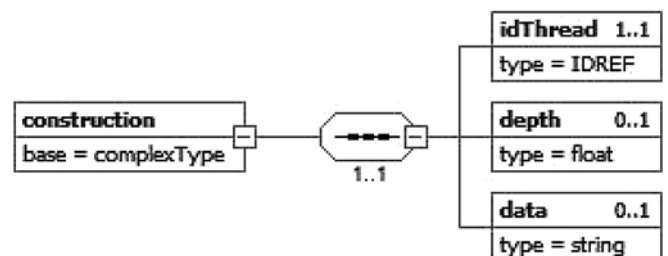


Рис. 6. Структура сегмента construction

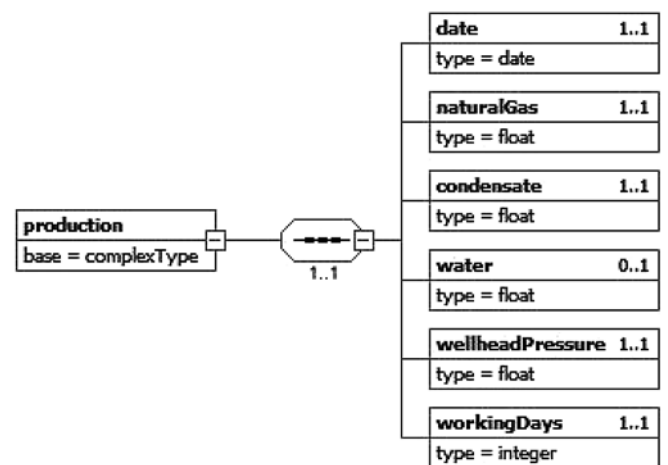


Рис. 7. Структура сегмента production

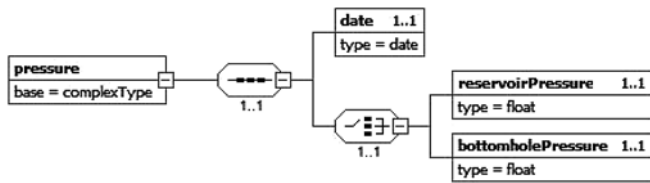


Рис. 8. Структура сегмента pressure

holePressure — значение забойного давления в скважине (рис. 8).

Синтезированная модель СЭК КСС позволяет сохранить фактографическую информацию по любой КСС ОГКМ.

Для оценки информационной энтропии КСС будем использовать формулу (1) с учетом функции устаревания (2) для ее динамических параметров. Тогда для КСС с полностью неизвестными параметрами значение информационной энтропии равно "стартовой энтропии", а для КСС с полностью известными параметрами — 0. Значение "стартовой энтропии" зависит от теоретического числа параметров описания БТС с учетом весомости данных параметров.

Предлагаемые значения весовых коэффициентов для статических и динамических параметров КСС представлены в табл. 1 и 2 соответственно.

При выборе весовых коэффициентов учитывались их значимость при моделировании КСС и трудоемкость получения.

Таблица 1

Значения весовых коэффициентов для статических параметров КСС

Параметр	Весовой коэффициент
Конструктивные параметры скважин КСС и трубопровода (за исключением его рельефа)	30
Рельеф трубопровода	20

Таблица 2

Значения весовых коэффициентов для динамических параметров КСС

Параметр	Весовой коэффициент
Дебиты газа, конденсата, воды и устьевое давление скважины	1
Пластовое давление скважины	10
Забойное давление скважины	3

Таблица 3

Результаты вычисления информационной энтропии КСС

Информационная энтропия	КСС1	КСС2
Без учета статических параметров	0,624256556	0,490918525
С учетом статических параметров	0,164869019	0,711980080

Для большинства КСС известны конструктивные (статические) параметры скважин и трубопровода за исключением его рельефа. По этой причине они сгруппированы и представлены двумя параметрами. Значения весов данных параметров объясняются их важностью. Отсутствие знания значений рассмотренных параметров препятствует моделированию КСС, а влияние рельефа учитывается в случаях эксплуатации обводненных скважин. Значения дебита газа, конденсата, воды и устьевого давления скважин получают ежемесячно из одного документа. Поэтому их состояние можно характеризовать одним параметром. В качестве оставшихся динамических параметров для скважин выбраны значения пластовых и забойных давлений. Трудоемкость получения пластового давления выше, чем забойного, что обусловлено геологическими особенностями продуктивных пластов ОГКМ. Для получения пластового давления необходимо дожидаться полного восстановления давления в забое скважины, что требует более длительной остановки скважины.

Проведем оценку информационной энтропии двух схожих по характеристикам КСС, в состав которых входят по три скважины. Рассмотрим период их эксплуатации, начиная с июня 1988 г. по настоящий момент. В обозначенный период конструкции шлейфов и скважин КСС не подвергались модификациям, т. е. система сбора функционировала в установленном режиме. Задвижки, размещенные на устьях скважин, находились в полностью открытом состоянии в обеих системах сбора продукции. Определим названия для систем сбора: КСС1 и КСС2, для которой отсутствует информация о рельефе шлейфа.

Для автоматизации процесса вычисления информационной энтропии КСС воспользуемся языком запросов XQuery, разработанным для обработки данных в формате XML. Результаты вычисления информационной энтропии представлены в табл. 3.

Анализ результатов, представленных в табл. 3, позволяет сделать следующий вывод: КСС2 обладает меньшим значением информационной энтропии без учета статических параметров, что обусловлено регулярными ГИС по всем ее скважинам в конце рассматриваемого периода, но при учете статических параметров значение ее энтропии больше, чем у КСС1, в связи с отсутствием информации о рельефе трубопровода. Полученный результат дает возможность ГЛПР оценить уровень информационной энтропии всех рассматриваемых систем, что позволяет реализовать возможность синтеза допустимого порогового значения. Соблюдение допустимого значения энтропии для БТС снижает вероятность принятия некорректных управляющих воздействий, обусловленных расхождениями между реальным и отраженным в эксплуатационной документации состояниями. В качестве порогового

значения выбирается среднее значение энтропии для анализируемых систем в определенный момент времени. Пороговое значение энтропии подвергается корректировке со стороны ГЛПР в зависимости от оперативной обстановки на БТС.

### Заключение

Рассмотренный способ мониторинга информационной энтропии БТС на основе СЭК повышает эффективность принятия управленческих решений за счет автоматизированной оценки полноты знаний о системе, а применение квазиструктурированной модели СЭК для КСС позволяет автоматизировать данный процесс. Применение описанного способа мониторинга информационной энтропии для КСС позволит повысить эффективность извлечения продукции месторождения за счет реализации контроля за максимально допустимым уровнем информационной энтропии на КСС.

### Список литературы

1. **Петров Б. Н.** Избранные труды. Том 1. Теория автоматического управления. М.: Наука, 1983. 432 с.
2. **Полищук Ю. В.** Мониторинг информационной энтропии в задачах описания больших технических систем // *Мехатроника, автоматизация, управление*, 2015. № 6. С. 396–401.
3. **Палей Д.** Моделирование квазиструктурированных данных // *Открытые системы*. 2002. № 9. С. 57–64.
4. **Корогаев Ю. П., Тагиев В. Г., Гергедава Ш. К.** Системное моделирование оптимальных режимов эксплуатации объектов добычи природного газа. М.: Недра, 1989. 264 с.
5. **Закиров С. Н., Закиров Э. С., Закиров И. С., Баганова М. Н., Спиридонов А. В.** Новые принципы и технологии разработки месторождений нефти и газа. М.: Недра, 2004. 520 с.
6. **Полищук Ю. В., Черных Т. А.** Построение хранилищ данных, ориентированных на работу с квазиструктурированным информационным наполнением документов // *Информационные технологии*. 2010. № 4 (164). С. 9–14.
7. **Полищук Ю. В., Черных Т. А.** Синтез квазиструктурированных моделей информационного наполнения электронных документов // *Вестник компьютерных и информационных технологий*. 2012. № 6. С. 20–27.

**Yu. V. Polishuk**, PhD, Associate Professor of Computer Security mathematical software and information systems, e-mail: Youra\_Polishuk@bk.ru, Orenburg State University

## On the Method of Monitoring Information Entropy the Large Technical Systems

*The paper considers a method of monitoring the information entropy of large technical systems.*

*Management decisions for large technical systems realizes a group of decision-makers, based on the analysis of factual data obtained from the accompanying operational content.*

*The need for the participation of a group of decision-makers, decision-making process is dictated by the scale of the control system and the complexity of the control process.*

*The complexity of large technical systems, as well as the complexity of obtaining and processing of factual data accompanying operational content, contribute to the divergence of the real state of the control system and the status reflected in an operational content. The latter contributes to the growth of information entropy of large technical systems and the adoption of incorrect management decisions. Thus, the adoption of correct management decisions for large technical systems is allowed only if they meet the conditions of the maximum allowable value of information entropy.*

*The method of monitoring the information entropy of large technical systems on the basis of the accompanying operational content increases the efficiency of management decision-making by the automated evaluation of the completeness of knowledge about the system and the use of semistructured operational model content for collector-ray acquisition system allows you to automate this process.*

**Keywords:** entropy of the system, semistructured data, electronic documents processing, large technical systems

### References

1. **Petrov B. N.** Izbrannye trudy, vol. 1. Teorija avtomaticheskogo upravlenija, Moscow, Nauka, 1983. 432 p.
2. **Polishhuk Ju. V.** Monitoring informacionnoj jentropii v zadachah opisanija bol'shijh tehniceskijh sistem, *Mehatronika, avtomatizacija, upravlenie*, 2015, vol. 16, no. 6, pp. 396–401.
3. **Palej D.** Modelirovanie kvazistrukturirovannyh dannyh, *Otkrytye sistemy*, 2002, no. 9, pp. 57–64.
4. **Korotaev Ju. P., Tagiev V. G., Gergedava Sh. K.** Sistemnoe modelirovanie optimal'nyh rezhimov jekspluatacii ob#ektov dobychi prirodnogo gaza, Moscow, Nedra, 1989, 264 p.
5. **Zakirov S. N., Zakirov E. S., Zakirov I. S.** Novye principy i tehnologii razrabotki mestorozh-denij nefi i gaza, Moscow, Nedra, 2004, 520 p.
6. **Polishhuk Ju. V., Chernyh T. A.** Postroenie hranilishh dannyh, orientirovannyh na rabotu s kvazistrukturirovannym informacionnym napolnieniem dokumentov, *Informacionnye tehnologii*, 2010, no. 4 (164), pp. 9–14.
7. **Polishhuk Ju. V., Chernyh T. A.** Sintez kvazistrukturirovannyh modelej informacionnogo napolnenija jelektronnyh dokumentov, *Vestnik komp'juternyh i informacionnyh tehnologij*, 2012, no. 6, pp. 20–27.

М. Ю. Чинков, студент, e-mail: michael.chinkov@gmail.com  
Пензенский государственный технологический университет, г. Пенза, Россия

## Отслеживание статуса сервисов инфраструктуры веб-приложения

Рассматривается проблема отслеживания большого количества сервисов, поддерживающих работу веб-приложения, а также решение данной проблемы с помощью *service discovery*-инструментов.

**Ключевые слова:** веб-приложение, отслеживание сервисов, *Consul*, проверки состояния

### Введение

Современная инфраструктура веб-приложений имеет множество автономных сервисов, каждый из которых отвечает за определенный функционал сервиса. Данный подход дает огромное преимущество в масштабировании системы, а также в обеспечении отказоустойчивости, так как при поломке одного из компонентов система по-прежнему продолжает работать. Однако появляются новые недостатки, связанные с эксплуатацией и обслуживанием веб-сервиса в целом [1].

Основным недостатком распределенных вычислительных систем является сложность отслеживания состояния всех сервисов одновременно. Трудно понять, какой именно компонент вышел из строя, опираясь лишь на традиционные средства мониторинга и проверяя в отдельности каждый компонент.

Решением проблемы отслеживания статуса сервисов веб-приложения является реализация дополнительного вида мониторинга в виде методологии исследования сервиса (*service discovery*). *Service discovery* позволяет построить на своей базе полноценную систему отслеживания сервисов, которая дает возможность не только обнаружить, какой именно сервис вышел из строя, но и автоматически исправить ошибки и перезапустить сервис [2].

### 1. Принцип работы *Consul* как *Service Discovery* системы

Самым популярным решением в реализации практик *service discovery* является применение программного инструмента *Consul*. *Consul* является клиент-серверной системой, нацеленной на решение нескольких задач.

1. *Service discovery*. *Consul* обнаруживает наличие тех или иных сервисов веб-приложения, отслеживая поведение агентов по протоколам *DNS* и *HTTP*.

2. Проверка состояния сервисов (*health check*). *Consul* позволяет настроить в рамках кластера неограниченное число проверок сервиса любыми методами. Например, проверкой состояния сервиса база

данных является попытка подключения к *TCP*-порту, который должна прослушивать СУБД, командой *netcat*.

*Health check* обычно имеет одно из двух состояний — проходящий (*passing*), в котором описывается рабочее состояние сервиса и критический (*critical*), которое говорит о том, что сервис в данный момент времени не работает.

3. Поддержка множества датацентров. Концепция датацентра является основной концепцией *Consul*. Каждый датацентр имеет несколько агентов, работающих в роли серверов и собирающих информацию о нодах кластера. Каждый датацентр имеет собственное имя и содержит определенное число нод и сервисов, работающих внутри нод.

### 2. Реализация кластера *Consul* на практике

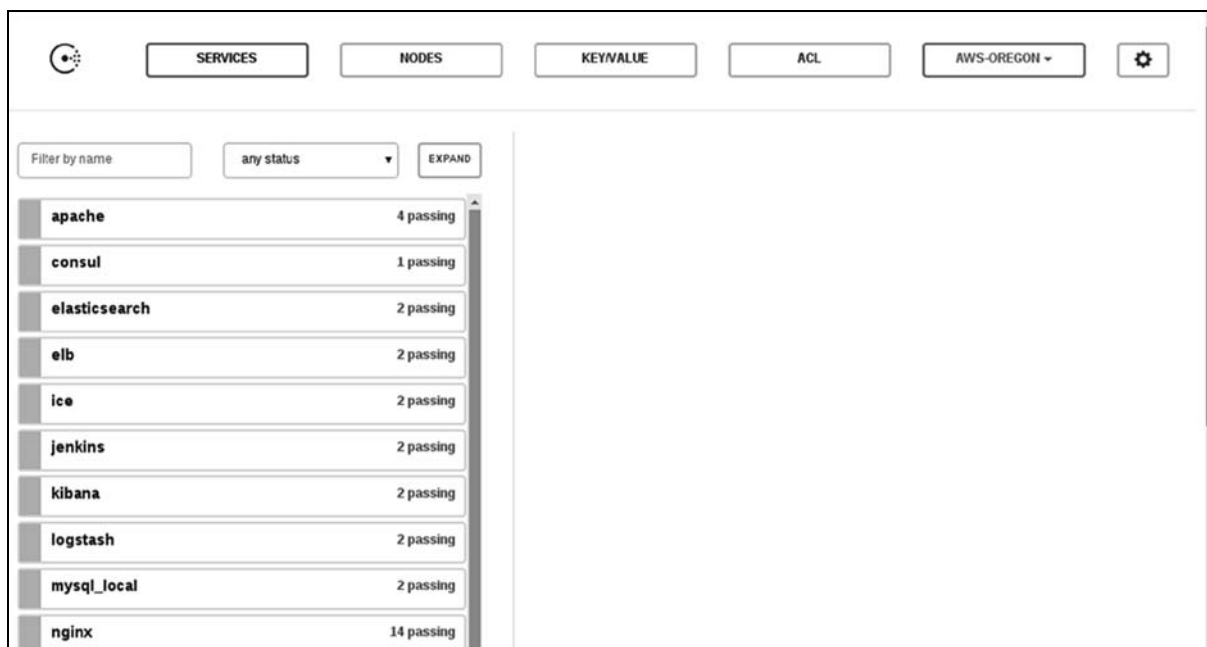
В рамках данного исследования был спроектирован полноценный кластер *Consul* из нескольких нод с конечной визуализацией состояния сервисов в виде статусной страницы (*status page*). Реализация кластера состояла из следующих базовых шагов.

1. Скачивание и установка *Consul* на нодах кластера. Процесс установки *Consul*-агента на одной ноде достаточно прост. Необходимо лишь скачать *zip*-архив и распаковать бинарный файл в директорию, из которой запускаются системные команды.

2. Скачивание пакета веб-интерфейса *Consul*. Отдельно на *Consul*-сервер скачивается пакет веб-интерфейса. Интерфейс разворачивается самостоятельно при запуске *Consul*, не требуя дополнительной конфигурации.

3. Создание аккаунта в *Hashicorp*. Для целевого веб-приложения был создан отдельный аккаунт в *Hashicorp*. Данный шаг не является необходимым, однако через созданный аккаунт возможна автоматическая интеграция новых нод в кластер без дополнительной переконфигурации сервиса.

4. Настройка системных демонов *Consul-agent* и *Consul-server*. По умолчанию *Consul* запускается из командной строки как *foreground*-процесс. Для полноценной работы кластера необходимо запускать



Веб-интерфейс системы отслеживания сервисов веб-приложения

агенты в виде фоновых процессов. Данная задача была реализована в виде работающих системных демонов на ОС Ubuntu 16.04.

5. Конфигурация прокси-сервера. Дополнительно на сервере *Consul*, где планировалось развернуть веб-интерфейс, был установлен и сконфигурирован прокси-сервер *Nginx*, который перенаправляет все соединения со стандартного 80 порта HTTP на 8500, который прослушивает *Consul*. Дополнительно настроена аутентификация для предотвращения входа в систему посторонних лиц.

6. Тестовый запуск кластера. На каждой ноде был запущен агент *Consul*. Таким образом, мы проверили, что работает базовый функционал кластера.

7. Добавление сущностей сервисов в кластер. На каждой ноде сконфигурированы сервисы веб-приложения, которые необходимо отслеживать. Каждый сервис имеет собственное имя, порт и сервер, на котором он работает. Дополнительно к сервисам были прикреплены проверки состояния (*health checks*), контролируемые через установленный временной интервал состояние работы сервисов.

8. Добавление триггеров состояния сервисов. На каждый сервис был прикреплен собственный триггер — *watch* — который запускает обработчик событий сразу после того, как *health check* сервиса приобрел критическое состояние.

9. Добавление обработчиков событий. В рамках исследования для каждой ноды был создан отдельный обработчик — *handler* — который проверяет доступность внутренних сервисов и пытается пе-

резапустить тот сервис, который был помечен как недоступный.

10. Перезагрузка конфигурации кластера. Для вступления изменений конфигурации в силу на каждой ноде кластера была проведена перезагрузка кластера запуском команды *Consul reload*.

11. Проверка работы кластера. Проверка работы кластера была проведена в два шага: запуск команды *Consul members* в консоли сервиса и проверка веб-интерфейса из локального браузера.

Все перечисленные выше шаги по созданию кластера были полностью автоматизированы. Программное обеспечение было установлено и сконфигурировано с помощью системы управления конфигурацией *Ansible*.

На рисунке показан один из результатов работы кластера в виде статусной страницы всех сервисов, работающих внутри инфраструктуры веб-приложения. Внутри веб-интерфейса можно отслеживать не только сервисы, но и ноды кластера, благодаря чему можно достаточно быстро понять, в какой конкретной точке произошел отказ.

## Заключение

Конечным итогом данного исследования является построение полноценно функционирующей системы отслеживания сервисов, которая делает процесс мониторинга работы веб-приложения и его компонентов гораздо проще.

Хотелось добавить, что в рамках данного исследования была реализована лишь самая примитив-

ная возможность *Consul* как инструмента для *service discovery*. Работая с полноценной микросервисной архитектурой, где сервисы, упакованные в контейнеры, не имеют постоянного места эксплуатации, необходима полноценная оркестрация инфраструктурой, динамически конфигурируя окружение путем изменений настроек подключения к DNS и системам мониторинга. *Consul* позволяет это делать автоматически и на лету. Также в *Consul* можно хранить данные в формате ключ—значение, записывая состояние о нодах кластера.

Таким образом, была рассмотрена проблема отслеживания состояния множества слабо связанных

сервисов веб-приложения с описанием одного из путей решения проблемы посредством внедрения в инфраструктуру приложения практики *service discovery* в виде исследования работы такого инструмента, как *Consul*.

#### Список литературы

1. Немет Э., Снайдер Г., Хейн Т. Р., Уэйли Б. *Unix и Linux. Руководство системного администратора* М.: Вильямс, 2012. 1202 с.
2. Limoncelli T., Chalup S. R., Hogan C. J. *The Practice of Cloud System Administration: Designing and Operating Large Distributed Systems*, Vol. 2. Addison-Wesley Professional, 2014. 560 p.

M. Yu. Chinkov, Student-Bachelor, e-mail: michael.chinkov@gmail.com  
Penza State Technological University (Penza, Russia)

## Tracking Services State in Web-Application Infrastructure

*This article regards the web-application services track and suggests a service discovery software as a one specific way of problem solution.*

**Keywords:** *web-application, service discovery, Consul, healthchecks*

#### References

1. Nemeth E., Sneyder H., Hayne T., Waley B. *Unix u Linux. Rukovodstvo Sistemnogo Administratora*, Moscow, Williams, 2012, 1202 p. (in Russian).

2. Limoncelli T., Chalup S. R., Hogan C. J. *The Practice of Cloud System Administration: Designing and Operating Large Distributed Systems*, Vol. 2. Addison-Wesley Professional, 2014. 560 p.

### ИНФОРМАЦИЯ

3—7 апреля 2017 г. в Казанском (Приволжском) федеральном университете состоится международная научная конференция, одиннадцатая в серии ежегодных конференций, посвященных развитию и применению параллельных вычислительных технологий в различных областях науки и техники

## «Параллельные вычислительные технологии (ПаВТ) 2017»

**Главная цель конференции** — предоставить возможность для обсуждения перспектив развития параллельных вычислительных технологий и представления результатов, полученных ведущими научными группами в использовании суперкомпьютерных технологий для решения задач науки и техники.

Тематика конференции покрывает все аспекты применения высокопроизводительных вычислений в науке и технике, включая приложения, аппаратное и программное обеспечение, специализированные языки и пакеты.

В первый день работы конференции будет объявлена **26-я редакция** списка Top50 самых мощных компьютеров СНГ.

Во все дни работы конференции будет действовать суперкомпьютерная выставка, на которой ведущие производители аппаратного и программного обеспечения представят свои новейшие разработки в области высокопроизводительных вычислений.

**Официальный сайт конференции:** <http://agora.guru.ru/pavt2017/>



**ЖУРНАЛ В ЖУРНАЛЕ**



**НЕЙРОСЕТЕВЫЕ  
ТЕХНОЛОГИИ**

**№ 1**  
**ЯНВАРЬ**  
**2017**

**Главный редактор:**

**ГАЛУШКИН А.И.**

**Редакционная коллегия:**

АВЕДЬЯН Э.Д.  
БАЗИЯН Б.Х.  
БЕНЕВОЛЕНСКИЙ С.Б.  
БОРИСОВ В.В.  
ГОРБАЧЕНКО В.И.  
ЖДАНОВ А.А.  
ЗЕФИРОВ Н.С.  
ЗОЗУЛЯ Ю.И.  
КРИЖИЖАНОВСКИЙ Б.В.  
КУДРЯВЦЕВ В.Б.  
КУЛИК С.Д.  
КУРАВСКИЙ Л.С.  
РЕДЬКО В.Г.  
РУДИНСКИЙ А.В.  
СИМОРОВ С.Н.  
ФЕДУЛОВ А.С.  
ЧЕРВЯКОВ Н.И.

**Иностранные  
члены редколлегии:**

БОЯНОВ К.  
ВЕЛИЧКОВСКИЙ Б.М.  
ГРАБАРЧУК В.  
РУТКОВСКИЙ Л.

**Редакция:**

БЕЗМЕНОВА М.Ю.  
ГРИГОРИН-РЯБОВА Е.В.  
ЛЫСЕНКО А.В.  
ЧУГУНОВА А.В.

**Харламов А. А.**

Ассоциативная память — нейросетевая среда для формирования единого пространства знаний . . . . . 66

**Пекунов В. В.**

Извлечение информации из нейронных сетей прямого распространения в виде простых алгебраических моделей 76

А. А. Харламов, д-р техн. наук., ст. науч. сотр., e-mail: kharlamov@analyst.ru,  
Институт высшей нервной деятельности и нейрофизиологии РАН, г. Москва

## Ассоциативная память — нейросетевая среда для формирования единого пространства знаний

*Ассоциативная память человека является средой для формирования единого пространства знаний. Рассмотрена обработка текстовой информации как пример процесса обработки человеком информации любой модальности с формированием однородной семантической сети ключевых понятий текста, ранжированных по степени их важности, а также иерархической тематической структуры, характеризующей сложность текста. Результаты текстового анализа наглядно интерпретируются на анализе конкретных текстов с применением аппарата теории вероятностей на основе подхода к тематическому моделированию текстов с использованием  $n$ -граммной модели текста вместо монограммной (так называемого "мешка слов").*

**Ключевые слова:** сложность, кортикоморфная ассоциативная память, структурная обработка информации, ассоциативная (однородная семантическая) сеть, иерархия тем, многомодальная информация, тематический анализ текстов,  $n$ -граммная модель текста

### Введение

Первым шагом к управлению сложностью является понимание понятия "сложность" применительно к конкретной задаче, и попытка оценить эту самую сложность. Для того чтобы подойти к этому вопросу конструктивно, используем некоторое, тоже конструктивное, представление о метрике в пространстве сложности. Для этого рассмотрим два базовых положения: (1) изоморфизм многомодального и лингвистического компонентов модели мира человека [1, 2]; и, в силу (1)—(2) рассмотрение в качестве модели мира корпуса текстов его (мир) описывающих.

С возможностью формирования модели мира [3] человек получил инструмент для оценки сложности мира. В отличие от модели мира животного (даже высокоорганизованного, например, высших приматов), модель мира человека (он имеет вторую сигнальную систему) состоит из двух компонентов — многомодального и лингвистического. Многомодальный компонент модели мира формируется у человека при участии социума как учителя. То есть человек не включает в модель ничего, что бы не было санкционировано социумом. Все это делается посредством языка, который является сегментирующей функцией при формировании модели мира: "Вот это стул — на нем сидят, а это стол — за ним едят!". Из этого следует, что многомодальный и лингвистический компоненты модели мира изоморфны: в многомодальном компоненте нет ничего, чему нет соответствия в лингвистическом компоненте.

Несомненно, что некоторый текст может описать некоторую ситуацию, корпус текстов — некоторую предметную область (раздел физики может быть описан текстами физического раздела библиотеки). Исчерпывающее множество корпусов тек-

тов, описывающих различные предметные области, может описать совокупность предметных областей — модель мира. Анализируя тексты таким образом, мы в состоянии проанализировать и описываемые ими ситуации, предметные области, мир в целом.

Когнитивный подход к обработке информации [4], например текстовой, позволяет оценить сложность обрабатываемой информации как количественно, так и качественно. Использование представлений об обработке информации в коре и гиппокампе, описанных в предыдущих частях работы [1, 2], является подходом к построению когнитивной семантической сети, характеризующей структуру анализируемой информации (например, текста).

Переописанная в терминах вероятностного тематического моделирования эта когнитивная семантическая сеть позволяет представить тематическую структуру анализируемой информации (например, текстовой) в виде тематического дерева, которое и характеризует ее (информации, или текста) сложность.

В результате такого переописания [5] выявляется тематический состав текста в терминах  $n$ -граммной модели текста, из которого формируется многоуровневая тематическая структура текста, ранжированная с помощью итеративной процедуры пересчета весовых характеристик слов в рамках  $n$ -граммной модели текста. Тематическая структура текста характеризует не только качественную, но и количественную сторону текста. Анализируя объем тематической структуры, число уровней, число тем на конкретном уровне, можно характеризовать сложность текста, корпуса текстов.

В первой части работы [1, 2] было показано, что обработка информации различными модальностями одинакова на всех уровнях и различается только на входе — первичной обработкой. Также было показано, что информация после первичной обработки,

по крайней мере для двух модальностей (речевой и зрительной), представляется в виде последовательностей событий. То есть она обрабатывается в коре головного мозга так же, как обрабатывается текстовая информация. При этом в коре формируются словари событий разной степени сложности (разной частоты встречаемости), которые объединяются в гиппокампе в образы целых ситуаций. Это значит, что и другая (кроме письменной текстовой) информация (далее — квазитекстовая информация), обрабатывается так же, как и текстовая информация. То есть ассоциативная память позволяет выявлять тематическую структуру квазитекстов других (помимо текстовой) модальностей, а также и многомодальных квазитекстов, формируя либо тематические представления отдельных модальностей с их (модальностей) поуровневым увязыванием, либо многомодальные тематические представления, аналогичные текстовым. В этом последнем случае квазитекст представляет собой последовательность образов событий разных модальностей, соответствующих имеющимся на входе сенсорам.

## 1. Тематическое моделирование на основе $n$ -граммной модели текста

В основе многоуровневого тематического представления лежат механизмы тематического моделирования, имеющего достаточно длительную историю [6]. Современные средства автоматического смыслового анализа текстов в основном делают акцент на тематическом анализе, который основан на использовании механизмов латентно-семантического анализа, вероятностного латентно-семантического анализа и скрытого распределения Дирихле [4]. Эти три типа тематических моделей рассматриваются исключительно в рамках монограммной модели текста — так называемого "Мешка слов".

### 1.1. $n$ -граммная модель текста

Монограммная модель текста основана на предположении, что каждое слово появляется в тексте независимо от остальных слов;  $n$ -граммные модели рассматриваются редко, и то в связи с использованием некоторых эмпирик частного характера (включение биграмм-словосочетаний в монограммную модель), которые не масштабируются на более общие случаи [7]. Простая монограммная модель в этих подходах используется исключительно ввиду большой сложности тематических моделей, которая и так зашкаливает в случае использования распределения Дирихле.

Если вероятности появления слов в любой позиции строки слов независимы и одинаково распределены, то вероятность появления строки равна произведению вероятностей появления слов:

$$p(w_1 \dots w_n) = p(w_1) \dots p(w_n).$$

Это, в частности, значит, что любые перестановки слов строки  $w_1 \dots w_n$  имеют одну и ту же вероятность, что заведомо неверно.

В случае использования монограммной модели, обозначив через  $C(w) = C(w_1 \dots w_n)$  число вхождений строки  $w = w_1 \dots w_n$  в некую генеральную совокупность всех текстов рассматриваемого языка, можно определить вероятность  $p(w)$  появления строки слов  $w = w_1 \dots w_n$  в тексте, оценка которой равна отношению  $C(w)$  к общему числу экземпляров всех встреченных в совокупности строк длины  $n$ :

$$\hat{p}(w) = \frac{C(w)}{\sum_{w^*} C(w^*)}.$$

В частности, для монограмм, т. е. отдельных слов,

$$\hat{p}(w^i) = \frac{C(w^i)}{\sum_{w^j} C(w^j)},$$

где  $w^i$  — слово словаря  $V_T$ , числитель — число вхождений  $w^i$  в генеральную совокупность, а сумма в знаменателе — просто общее число слов в ней.

Если нет достоверного априорного знания о равенстве распределений слов в разных позициях строки, следует попытаться ввести контекстную привязку — через условные вероятности [8]. Поэтому при переходе к  $n$ -граммной, а точнее, к "односторонней"  $n$ -граммной модели, а именно принятой при использовании " $n$ -грамм правосторонней" модели, вероятность очередного слова строки задается в зависимости от предшествующих ему  $(n - 1)$  слов, что записывается как  $p(w_n | w_1 \dots w_{n-1})$ . Тогда  $p(w_1 \dots w_{n-1} w_n) = p(w_n | w_1 \dots w_{n-1}) p(w_1 \dots w_{n-1})$ .

В терминах вероятности "быть справа" имеем:

$$p(w_1 \dots w_{n-1} w_n) = p(w_n | w_1 \dots w_{n-1}) \times \\ \times p(w_{n-1} | w_1 \dots w_{n-2}) \dots p(w_3 | w_1 w_2) p(w_2 | w_1) p(w_1)$$

или

$$p(w_1 \dots w_n) = \left( \prod_{k=1}^n p(w_k | w_1 \dots w_{k-1}) p(w_1) \right).$$

Введя фиктивный символ "начало" и договорившись, что  $p(w_1 | w_0) \cong p(w_1)$ , можно переписать:

$$p(w_1 \dots w_n) = \left( \prod_{k=1}^n p(w_k | w_1 \dots w_{k-1}) \right).$$

Оценкой вероятности  $n$ -граммы служит частота ее встречаемости:

$$\hat{p}(w_i | w_{i-n} \dots w_{i-1}) = f(w_i | w_{i-n} \dots w_{i-1}) = \\ = \frac{C(w_{i-n} \dots w_{i-1} w_i)}{C(w_{i-n} \dots w_{i-1})}.$$

Так, для биграммной модели оценкой вероятности биграммы является частота ее появления в тексте:

$$\hat{p}(w_j|w_i) = f(w_j|w_i) = \frac{C(w_j|w_i)}{C(w_i)}$$

## 1.2. Использование $n$ -граммной модели для выявления тематики текста

Использование  $n$ -граммной модели текста в тематическом моделировании позволяет корректно интерпретировать его результаты с лингвистической точки зрения. Для этого вводится иное представление о темах текста, по сравнению с представлением, используемым в перечисленных подходах.

Так в биграммной модели темами считаются те "вторые" (в биграмме "первое слово — второе слово") слова наибольшего ранга, которые связаны с наибольшим числом "первых" слов. В триграммной модели темами считаются те "третьи" слова (вторые "вторые") наибольшего ранга, которые связаны с наибольшим числом "вторых" слов, имеющих наибольший ранг в терминах биграммной модели. И так до  $n$ -го порядка модели. Тогда в  $n$ -граммной модели темами считаются те " $n$ -е" слова наибольшего ранга, которые связаны с наибольшим числом " $(n-1)$ -х" слов  $(n-1)$ -граммной модели. Таким образом, появляется возможность автоматического формирования тематических деревьев, в которых главными темами текста являются темы  $n$ -го уровня ( $n$ -е слова) наибольшего ранга, их подтемами являются темы  $(n-1)$ -го уровня ( $(n-1)$ -е слова), их подподтемами — темы  $(n-2)$ -го уровня, и т. д. Поскольку в процессе построения этих деревьев происходит ранжирование их вершин, автоматически ранжируются и построенные тематические деревья.

Эти построения будут более понятны, если использовать не традиционную терминологию: "текст", "биграммы", "триграммы", а понятия, связанные с однородной семантической сетью, описывающей анализируемый текст. Сеть строится как множество пар "первое слово — второе слово", где вторая пара стыкуется с первой парой через общее слово. Формальное описание ассоциативной (однородной семантической) сети  $N$  [5] имеет следующий вид.

**Определение 1.** Под семантической сетью  $N$  понимается множество несимметричных пар событий  $\{<c_i c_j>\}$ , где  $c_i$  и  $c_j$  — события, связанные между собой отношением ассоциативности (совместной встречаемости в некоторой ситуации):

$$N \equiv \{<c_i c_j>\}.$$

В случае анализа текста этими событиями являются слова  $w_i$  текста. Следующие друг за другом пары слов  $<w_i, w_j>$  текста и составляют однородную семантическую (ассоциативную) сеть.

**Определение 2.** Семантическая сеть, описанная таким образом, может быть представлена и как множество так называемых звездочек  $\{<c_i\{c_j\}>\}$ , где

все пары с одинаковым первым событием собраны вместе:

$$N \equiv \{z_i\} = \{<c_i\{c_j\}>\}.$$

**Биграммная модель.** Если рассматривать текст как последовательность из пар слов, получим биграммную модель. Для каждого второго слова  $w_j$  строки из двух слов  $<w_i, w_j>$  первое слово строки  $w_i$  ("слева-направо") является темой:  $w_i \equiv t_i^2$  (индекс 2 — поскольку биграммная модель). Все пары слов с одинаковыми темами объединяются в звездочки. В этом случае вершина сети, соответствующая слову  $w_i \equiv t_i^2$ , является корневой вершиной одного из тематических деревьев (в данном случае — звездочек). Так как вероятность появления строки из двух слов (правосторонняя модель) в тексте  $p(w_i, w_j) = p(w_j|w_i)p(w_i)$ , то вероятность появления темы  $w_i \equiv t_i^2$  в биграммной модели есть сумма вероятностей появления пар с одинаковым первым словом (вероятность появления звездочки):

$$p(t_i^2) = p(w_i^2) = \sum_{j=1}^{J_i} p(w_i, w_j) = \sum_{j=1}^{J_i} p(w_j|w_i)p(w_i^1),$$

где  $J_i$  — число слов  $w_j$  (ассоциантов  $w_j$  звездочки), связанных с первым словом  $w_i$ . А вероятности  $p(w_i^1)$  — это исходные вероятности распределения слов в тексте. Для однородности вводится условное понятие "темы"  $p(w_i^1) \equiv t_i^1$  для монограммного распределения. И так для каждого второго слова  $w_j$  первое слово пары  $w_i$  ("слева-направо") является темой, то  $w_i \equiv t_i^2$ :

$$p(t_i^2) = p(w_i^2) = \sum_{j=1}^{J_i} p(w_i, w_j) = \sum_{j=1}^{J_i} p(w_j|w_i)p(t_i^1). \quad (1)$$

Для того чтобы общая полученная сумма вероятностей  $P(t_i^2)$  была равна единице:  $\sum_{n=1}^N P(t_n^2) = 1$ ,

где  $n$  — число тем, полученные для каждой темы  $t_n^1$  суммы нормируются на единицу. В общем случае число тем совпадает с числом всех слов текста  $T = W$ , но обычно выбираются лишь несколько главных тем:  $T \ll W$ .

Нормирование осуществляется на сумму по всем темам  $t_i^1$ :

$$p(t_i^2) = \frac{\sum_{j=1}^{J_i} p(w_j|w_i)p(t_i^1)}{T \sum_{i=1}^{J_n} p(w_j|w_i)p(t_i^1)}. \quad (2)$$

Здесь  $p(t_i^1)$  в формуле (1) означает вероятность появления отдельного слова (т. е. вероятность из монограммной модели). В этом случае  $p(t_i^2)$  — вероятность появления звездочки из биграммной модели. Причем главное слово звездочки в терминах биграммной модели является темой для ее ближайших ассоциантов — семантических признаков — "вторых" слов.

**Триграммная модель текста.** Представление текста как последовательности трехсловных фрагментов является триграммной моделью текста:

$$p(w_i w_j w_k) = p(w_k | w_i w_j) p(w_i w_j) = p(w_k | w_i w_j) p(w_j | w_i) p(w_i).$$

В этом случае вероятность появления строки из первых двух слов в строке из трех слов получается, как и в (1), суммированием по третьему слову:

$$p(w_i w_j) = \sum_{k=1}^{K_j} p(w_i w_j w_k),$$

и вероятность появления первого слова в строке из трех слов получается суммированием по второму слову:

$$p(w_i) = \sum_{j=1}^{J_i} \sum_{k=1}^{K_j} p(w_i w_j w_k).$$

Тогда  $p(t_i^3)$  вычисляется в соответствии с выражением (1) в рамках биграммной модели,  $p(w_i) \equiv p(t_i^2)$ , а вероятность появления второго слова пары в зависимости от появления первого слова пары в тексте одна и та же, независимо от граммности модели текста, как следует из сетевого представления текста (взаимосвязанные пары слов остаются теми же):

$$p(t_i^3) = \frac{\sum_{j=1}^{J_i} p(w_j | t_i^2) p(t_i^2)}{\sum_{i=1}^{I} \sum_{j=1}^{J_i} p(w_j | t_i^2) p(t_i^2)}, \quad (3)$$

где  $t_i^2$  и  $t_i^3$  обозначают, соответственно, темы в соответствии с биграммной и триграммной моделями, а число тем, как и ранее, соответствует числу слов в тексте  $T = W$ , но обычно ограничивается левым решением до  $T \leq W$ . К корневым вершинам  $t_i^3$  крепятся через "вторые" слова-вершины  $t_i^2$  звездочки  $z_i^3$  звездочки биграммной модели  $z_i^2$ :

$$z_i^2 = \{ \langle t_i^2 \{ t_i^1 \} \rangle \}.$$

**Четырехграммная и более граммные модели.** Та же процедура, примененная для анализа четырехграммной модели, приводит к получению следующего

множества корневых вершин, к которым прикрепляются все графы (тематические структуры), полученные на этапе анализа триграммной модели. То есть мы можем вычислить веса тематических вершин семантической сети текста с учетом их зависимости в последовательности из четырех слов.

То же будет и для  $n$ -граммной модели. Хотя надо заметить, что начиная с некоторого  $n$  процесс сойдется, так как зависимость слов, как правило, в модели языка сказывается не далее, чем на длине простого предложения.

Звездочка, в которой главным словом является тема  $n$ -граммной модели  $t_i^n$ , а второстепенными словами (ближайшими ассоциантами) являются темы  $(n-1)$ -граммной модели  $t_i^{n-1}$ , к которой присоединены звездочки, в которых главными словами являются соответствующие темы  $t_i^{n-1}$   $(n-1)$ -граммной модели, а второстепенными словами — темы  $t_i^{n-2}$   $(n-2)$ -граммной модели, к которым, в свою очередь, таким же образом присоединены соответствующие звездочки моделей более низкого уровня (этакая гроздь), формирует **тематическое дерево**. И здесь на каждом уровне тематического дерева число тем, как и ранее, соответствует числу слов словаря текста  $T = W$ , но обычно ограничивается до  $T \leq W$ .

**Итеративная процедура перевзвешивания.** В результате имеется итеративная процедура перевзвешивания, которая позволяет найти значения вероятностей появления тем  $t_i^n$  (в случае использования  $n$ -граммной модели) в тексте:

$$p(t_i^2) = \frac{\sum_{j=1}^{J_i} p(w_j | w_i) p(t_i^1)}{\sum_{i=1}^I \sum_{j=1}^{J_n} p(w_j | w_i) p(t_i^1)};$$

$$p(t_i^3) = \frac{\sum_{j=1}^{J_i} p(w_j | w_i) p(t_i^2)}{\sum_{i=1}^I \sum_{j=1}^{J_i} p(w_j | w_i) p(t_i^2)};$$

...

$$p(t_i^n) = \frac{\sum_{j=1}^{J_i} p(w_j | w_i) p(t_i^{n-1})}{\sum_{i=1}^I \sum_{j=1}^{J_i} p(w_j | w_i) p(t_i^{n-1})},$$

где  $p(t_i^1) = p(w_2)$  и  $p(w_j | w_i)$  — одинаковая для всех шагов итерации вероятность появления последующего слова текста при условии появления предыдущего слова.

Если мы выберем из всех тематических деревьев одно, имеющее вершину с наибольшим весом, получим минимальный древовидный подграф (наибольшего ранга) циклического графа, характеризующего текст, — семантической сети текста. Полученная тематическая структура текста может быть использована для анализа сложности текста.

## 2. Ассоциативная память — среда для формирования единого пространства знаний в структуре управления сложными процессами

### 2.1. Тематическая структура текста

Тематическая структура текста получается как минимальный древовидный подграф циклического графа текста — однородной семантической сети — путем обрыва слабых циклических связей [3]. Если рассмотреть все множество звездочек анализируемого текста (что то же самое — множество пар слов текста), то можно заметить, что они образуют циклический граф — однородную семантическую сеть. После процедуры перенормировки весовых характеристик вершин сети те вершины, значение которых в тексте велико, вследствие их большей связности по отношению к другим вершинам, появляются возможность выявить наиболее важные вершины семантической сети текста. Самая весомая вершина, таким образом, может быть названа главной темой текста. Вершины, отстоящие от нее на один шаг, являются семантическими признаками этой вершины. Если отобразить только самые весомые из них (а среди них найдутся такие, иначе главная тема не была бы самой весомой), они будут как бы объяснениями главной темы — ее подтемами. То же можно сказать о вершинах — ближайших ассоциантах подтем в семантической сети: среди них можно найти достаточно весомые вершины, которые будут семантическими признаками подтем — подподтемами следующего уровня. Этот процесс можно продолжать до тех пор, пока последний уровень вершин-ассоциантов не окажется с весовыми характеристиками ниже заданного порога.

### 2.2. Тематическая сложность текста

Эта иерархическая структура тем — тематическое дерево текста — может быть использована для анализа сложности текста, но ровно также и для анализа сложности корпуса текстов, описывающих, например, одну предметную область, и для анализа корпуса текстов, описывающих несколько предметных областей, и даже модель мира в целом.

Рассмотрим на нескольких примерах что собой представляет тематическая структура (текста). Для этого проанализируем тематически компактный текст, несколько текстов в совокупности, представляющих одну предметную область, а также текстов, представляющих несколько разнородных предметных областей. Для анализа используем програм-

мный продукт TextAnalyst, который формирует тематическое представление анализируемого текста описанным выше способом [9].

В качестве примера рассмотрим текст "Анализ текстов: Лингвистика, семантика, прагматика в рамках когнитивного подхода" [5], опубликованный в журнале "Проблемы искусственного интеллекта" (г. Донецк) в 2015 г. Его тематическая структура представлена на рис. 1.

Анализ тематической структуры показывает, что основной темой текста является тема "Текст". Далее: тематическая структура этого текста содержит шесть основных тем (из них пять — самого высокого ранга): "Предложения", "Анализ текстов", "Обработка [текста]", "Представление [результатов]", "Получение (способ вычисления)".

Наибольшее число уровней подподтем — девять оказалось у подтемы "Предложение", в двух цепоч-

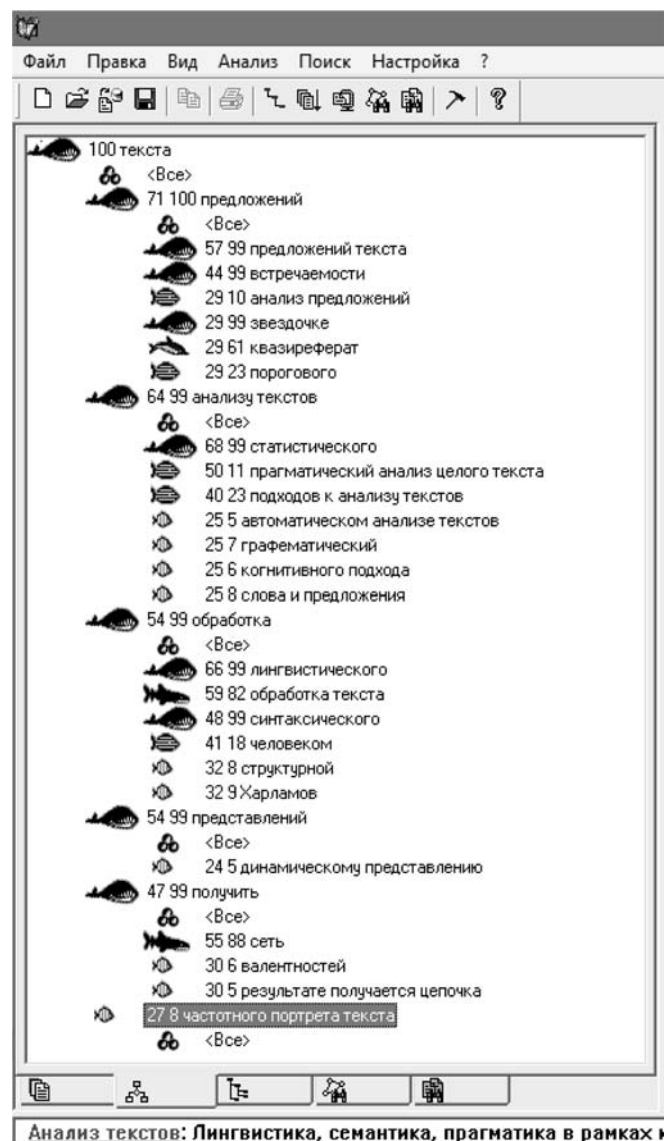


Рис. 1. Тематическая структура текста "Анализ текстов: Лингвистика, семантика, прагматика в рамках когнитивного подхода"

ках. Первая цепочка: "Предложение", "Звездочка", "Множество", "Точка", "Движение", "Изменить", "Тело", "Закон Ньютона", "Закон Ньютона выполняется не во всякой системе отсчета", "Называется инерциальной системой отсчета". Надо заметить, что последние семь уровней подподтем этой темы относятся к компактному фрагменту текста, включенному в основной текст — отрывку из учебника физики. Поэтому отбросим этот пример и рассмотрим другую цепочку: "Предложение", "Звездочка", "Предикатная структура", "Расширенная предикатная структура", "Предикат", "Главное [слово]", "Субъект", "Главный объект", "Связь", "Предикативная связь субъекта с главным объектом". Можно заметить логику связи тем между уровнями, касающуюся понятия "Расширенная предикатная структура", и его связь с понятием "Предложение" с точки зрения семантики: расши-

ренная предикатная структура исчерпывает семантику предложения. Вместе с тем она (расширенная предикатная структура) является "звездочкой".

Тематический анализ, таким образом, позволяет выявить сложность этого текста: шесть тем в ширину и одна из этих тем имеет девять уровней в глубину, т. е. не очень сложный текст.

Для более обобщенного представления текста оставим в сети только самые весомые темы (рис. 2). Можно заметить, что в этом случае тематическое представление текста довольно компактно: в нем только один длинный хвост.

В качестве другого примера рассмотрим предметную область "Автоматический анализ текстов" (рис. 3), представленную несколькими нашими текстами, опубликованными ранее в различных изданиях: 1) Харламов А. А. Модель мира человека — семантическая сеть [10]; 2) Харламов А. А., Ермо-

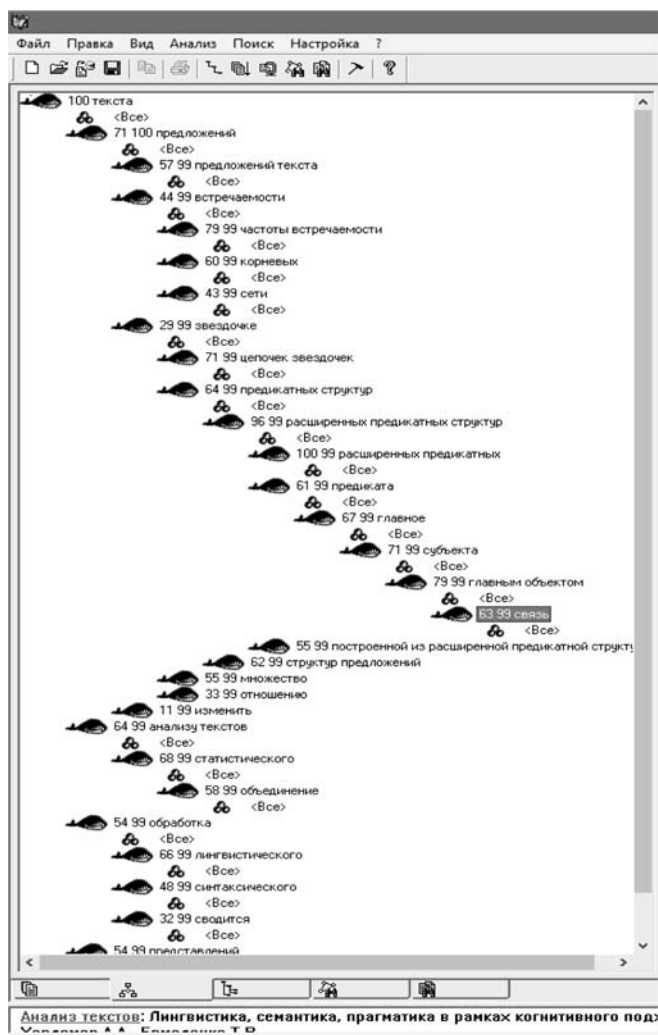


Рис. 2. Тематическое дерево того же текста с самыми весомыми темами

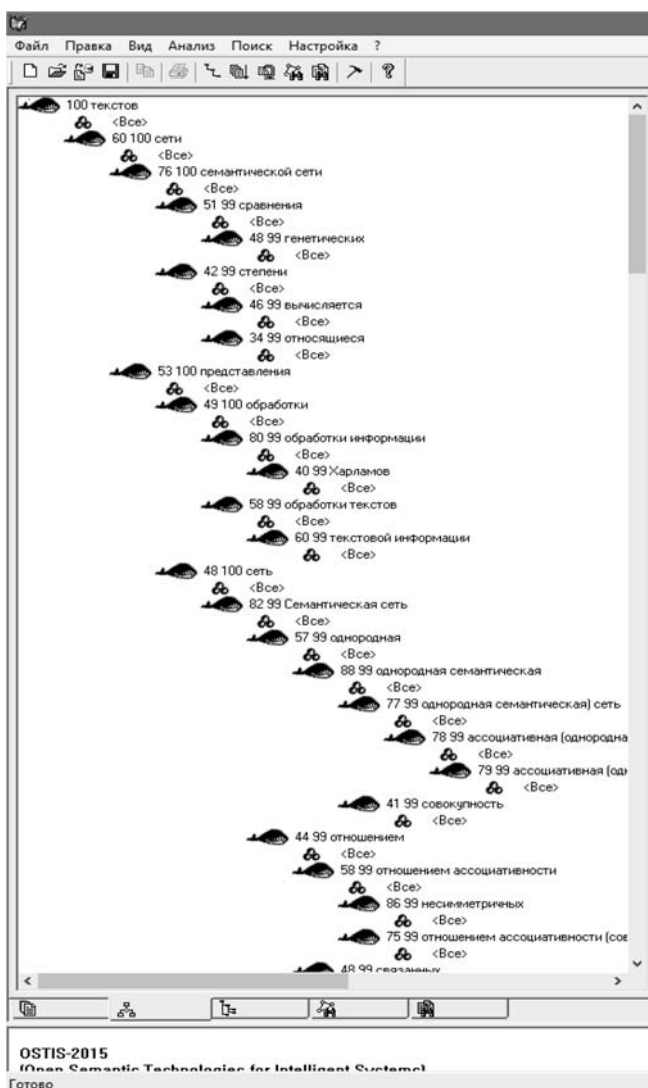


Рис. 3. Тематическая структура корпуса текстов, описывающего предметную область "Автоматический анализ текстов"

ленко Т. В., Жонин А. А. Моделирование динамики процессов последовательностью однородных семантических сетей на основе анализа последовательности описывающих процесс текстовых выборок [11]; 3) Харламов А. А. Нейросетевой тест на вхождение в сетевое сообщество [12]; 4) Харламов А. А. Информационная модель личности [13]; 5) Харламов А. А., Ермоленко Т. В. Анализ текстов: Лингвистика, семантика, прагматика в рамках когнитивного подхода. [4]; 6) Харламов А. А., Ермоленко Т. В. Нейросетевая среда (нейроморфная ассоциативная память) для преодоления информационной сложности. Поиск смысла в слабоструктурированных массивах информации. Часть I. Структурная обработка информации в коре [1]; 7) Харламов А. А., Ермоленко Т. В. Нейросетевая среда (нейроморфная ассоциативная память) для преодоления информационной сложности. Поиск

смысла в слабоструктурированных массивах информации. Часть II. Обработка информации в гиппокампе. Модель мира [2]; 8) Куликов А. М., Харламов А. А. Использование однородной семантической сети для классификации результатов генетического анализа [14]; 9) Харламов А. А., Рыжов В. А., Курдюмов В. С. Синхронизация предметных областей коммуникантов в процессе диалога [15].

Анализ тематической структуры этой текстовой выборки (также с учетом только самых весомых тем) показывает, что основной темой корпуса текстов является тема "[Анализ] текстов". Далее, тематическая структура выборки достаточно компактна — содержит четыре основных темы: "Сети", "[Виды] анализа текстов", "Предложения [лингвистический подход]", "[Модель] предметной [области]". Однако длинных хвостов значительно больше (около 10) — по числу наиболее важных в выборке

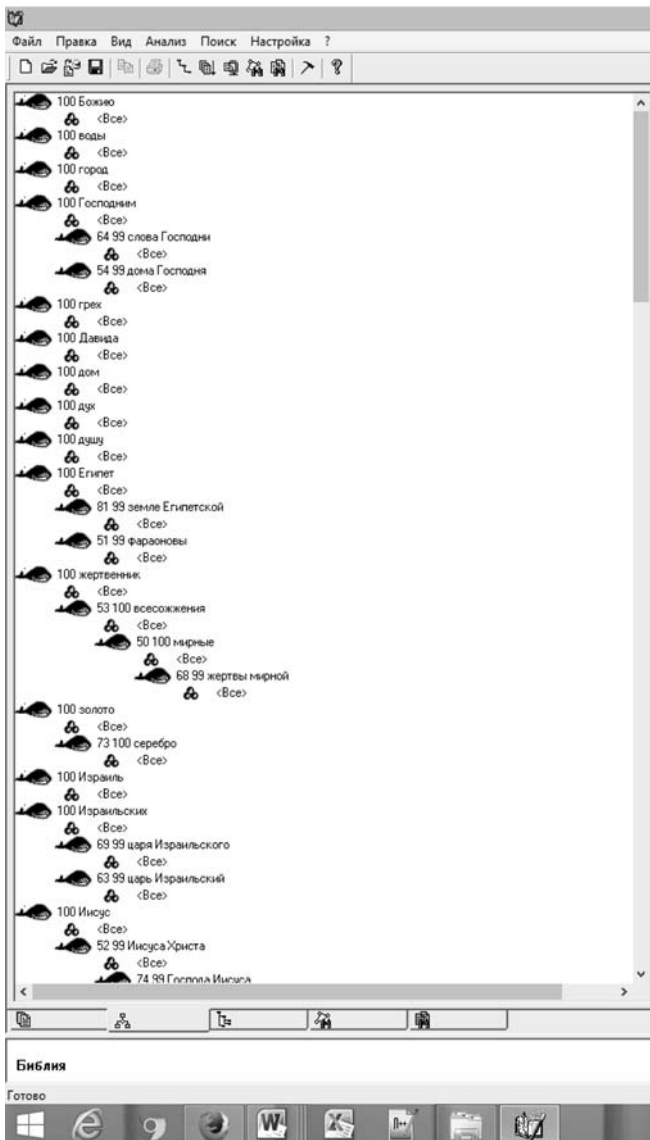


Рис. 4. Результаты анализа тем Библии (из рисунка видно, что тем много)

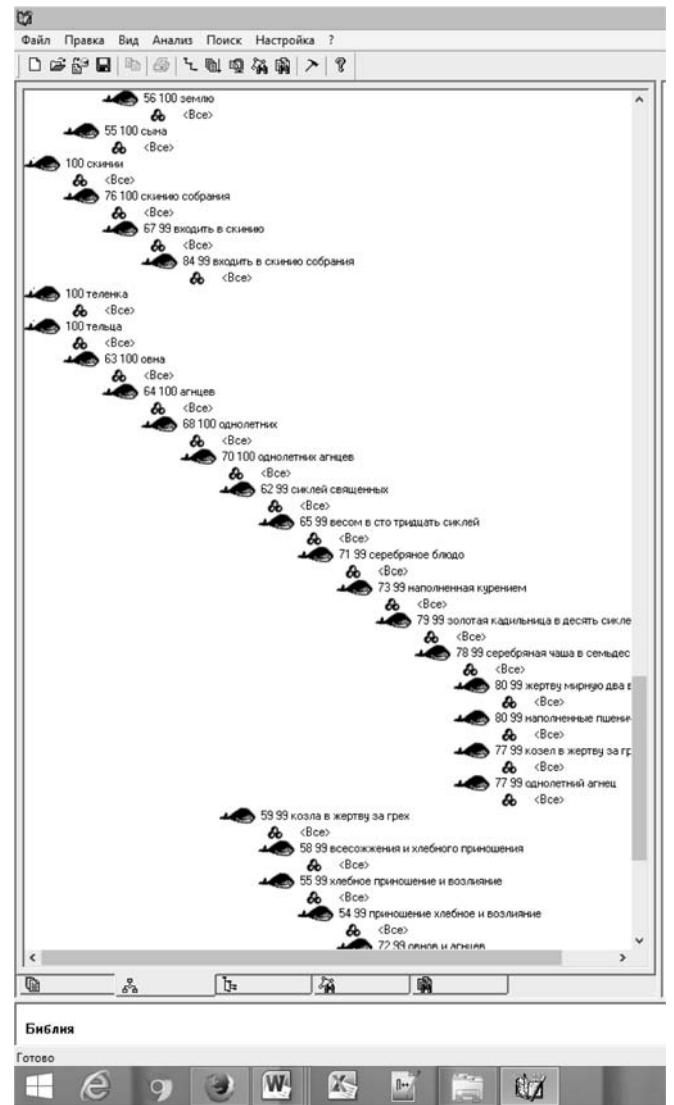


Рис. 5. Результаты анализа текста Библии (некоторые темы глубокие)



подтем. И по крайней мере три темы из четырех — "Сети", "Предложения [лингвистический подход]", "[Модель] предметной [области]" имеют длинные хвосты, т. е. тематическое дерево сложно в глубину.

И, наконец рассмотрим текст, например Библии (Ветхий завет. Книга "Бытие" вместо тематически разнородных текстов), как относящийся к разным предметным областям. Чтобы упростить анализ, не будем принимать во внимание связи слабее 50. Результаты анализа можно увидеть на рис. 4 и 5. Из которых видно, что в этом случае наблюдаются как первый, так и второй случаи: тем много, и некоторые из них — достаточно глубокие.

### 2.3. Ассоциативная память как средство структурирования модели мира

В первых частях работы [1, 2] было показано, что обработка информации разных модальностей одинакова на всех уровнях и различается только на входе — первичной обработкой. Также было показано, что информация после первичной обработки по крайней мере для двух модальностей (речевой и зрительной), представлена в виде последовательностей событий. То есть она обрабатывается в коре головного мозга так же, как обрабатывается текстовая информация — отображением множества входных последовательностей  $\{A\}_m$ , где  $m$  — номер модальности в многомерное сигнальное пространство  $R^n$ . При этом в колонках соответствующих отделов коры формируются словари  $\{\hat{B}_i\}_{jkm}$ , где  $i$  — номер слова в словаре,  $j$  — номер словаря на уровне,  $k$  — номер уровня,  $m$  — номер модальности событий разной степени сложности (разной частоты встречаемости), которые объединяются в ламелях гиппокампа в образы целых ситуаций. Это значит, что и другая (кроме текстовой) информация, обрабатывается так же, как и текстовая информация. То есть ассоциативная память позволяет выявлять тематическую структуру квазитекстов других (помимо текстовой) модальностей, а также и многомодальных квазитекстов, формируя либо тематические представления отдельных модальностей с их (модальностей) поуровневым увязыванием образов событий одного уровня, либо многомодальные тематические представления, аналогичные текстовым. В этом последнем случае квазитекст представляет собой последовательность образов событий разных модальностей, соответствующих имеющимся на входе сенсорам.

### 2.4. Ассоциативная память как средство для ранжирования элементов модели

В отличие от ассоциативной памяти, реализованной в коре головного мозга [1], которая структурирует входную сенсорную информацию, формируя иерархию словарей разной сложности обра-

зов событий разных модальностей, ассоциативная память гиппокампа, также принимающая участие в обработке разномодальной информации [2], объединяет образы этих событий в семантические сети  $N_p = \cup_i B_i$ , описывающие целые ситуации, и ранжирует образы событий в этих сетях  $w_i(t+1) = \left( \sum_{i \neq j} w_i(t)w_{ij} \right) \sigma(\bar{E})$  с точки зрения важности их (событий) участия в представлении упомянутых ситуаций. Здесь  $w_i(0) = z_i$ ;  $w_{ij} = z_{ij}/z_j$  и  $\sigma(\bar{E}) = 1/(1 + e^{-k\bar{E}})$  — функция, нормирующая на среднее значение энергии всех вершин сети  $\bar{E}$ ;  $z_i$  — частота встречаемости  $i$ -го слова в тексте;  $z_{ij}$  — частота совместной встречаемости  $i$ -го и  $j$ -го слов в фрагментах текста;  $t$  — номер итерации.

Причем образы событий, имеющие большую связность с образами других событий, имеют и больший ранг. Процедура тематического моделирования, представленная в начале статьи, фактически переописывает это представление в терминах теоретико-вероятностного подхода [5].

### 2.5. Многомодальная семантическая сеть как средство представления смысла

Таким образом, оба вида ассоциативной памяти участвуют в формировании фрагментов многомодальной семантической сети, которая опосредует модель мира человека. Как было сказано выше, эта модель включает в себя две части: лингвистическую и многомодальную.

Лингвистическая модель мира представляет собой многоуровневую иерархию, состоящую из словарей языковых событий разной сложности, соответствующих уровням языка от морфологического до уровня семантики сочетаемости слов, хранящихся в колонках коры. Нижний уровень (акустико-фонетический, или графематический) зависит от вида обрабатываемой информации — в нем представлены элементы устных, или письменных текстов, соответственно. Семантика предложений и текстов представлена в ламелях гиппокампа в виде семантических сетей, вершинами которых являются образы событий из колонок коры, а дуги — связи этих событий в ситуациях, представленных в ламелях гиппокампа как ассоциативной памяти.

Многомодальная часть модели мира представляет собой также иерархию, состоящую из словарей колонок коры, теперь уже разномодальных  $\hat{M}_m = \cup_{ij} \bigvee_{ijk < K} \hat{B}_{ijkm}$  (или объединенных в многомодальные  $\hat{M} = \cup_m \hat{M}_m = \cup_{ijm} \hat{B}_{ijKm} \bigvee_{ijk < Km} \hat{B}_{ijkm}$ ), событий разной сложности, соответствующих уров-

ням сложности описания событий внешнего и внутреннего мира человека. Они в той или иной степени соответствуют некоторым образам событий лингвистической части модели (изоморфизм лингвистического и многомодального компонентов модели мира). И между этими многомодальными и лингвистическими образами имеются поуровневые ассоциативные связи. Семантика многомодальных квазипредложений и квазитextов представлена в ламелях гиппокампа в виде семантических сетей, вершинами которых также являются образы  $\{\hat{B}_i\}_{jkm}$  событий из колонок коры, а дугами — связи этих событий в ситуациях, представленных в ламелях гиппокампа.

В результате совместной обработки при формировании словарей образов событий в коре и формирования семантических сетей и ранжирования образов событий в гиппокампе возникает хорошо структурированная и упорядоченная система представлений, проекция на которую входной ситуации приводит к отнесению ее (входной ситуации) к той или иной предметной области, к той или иной ситуации, т. е. выявление ее смысла.

### Заключение

В работе показано, что одна часть ассоциативной памяти как модель иерархии колонок коры полушарий головного мозга человека формирует статистический портрет входной информации в виде иерархии словарей входных событий различной частоты встречаемости. Образы событий объединяются в другой части ассоциативной памяти — ламелях гиппокампа в модель ситуации в виде семантической сети, которая становится семантическим портретом ситуации после итеративной процедуры перенормировки весов образов событий. Эти процессы наглядно интерпретируются в теоретико-вероятностных терминах тематического моделирования в случае использования  $n$ -граммной модели текста в процессе тематического моделирования. Изоморфизм двух частей модели мира человека — лингвистической и многомодальной позволяет предполагать, что такой способ анализа информации можно использовать для анализа информационных последовательностей (квазитextов) любой модальности, а также для анализа многомодальной информации в целях получения характеристики ее сложности. Сложность анализируемых корпусов текстов и квазитextов определяется глубиной и шириной полученных в результате анализа тематических деревьев, что хорошо видно на примере анализа естественно-языковых текстов.

*Работа выполнена при поддержке РГНФ (проект № 15-03-00860 "Методология управления сетевыми структурами в контексте парадигмы сложности").*

### Список литературы

1. Харламов А. А., Ермоленко Т. В. Нейросетевая среда (нейроморфная ассоциативная память) для преодоления информационной сложности. Поиск смысла в слабоструктурированных массивах информации. Часть I. Структурная обработка информации в коре // Информационные технологии. 2015. Т. 21, № 11. С. 814—820.
2. Харламов А. А., Ермоленко Т. В. Нейросетевая среда (нейроморфная ассоциативная память) для преодоления информационной сложности. Поиск смысла в слабоструктурированных массивах информации. Часть II. Обработка информации в гиппокампе. Модель мира // Информационные технологии. 2015. Т. 21, № 12. С. 883—889.
3. Харламов А. А. Нейросетевая технология представления и обработки информации (естественное представление знаний). М.: Радиотехника, 2006.
4. Харламов А. А., Ермоленко Т. В. Анализ текстов: лингвистика, семантика, прагматика в рамках когнитивного подхода // Проблемы искусственного интеллекта. 2015. № 0 (1). С. 107—115.
5. Харламов А. А. Формирование  $n$ -граммной тематической модели текста // Речевые технологии. 2016. № 1—2. С. 16—25.
6. Коршунов А., Гомзин А. Тематическое моделирование текстов на естественном языке // Труды Института системного программирования РАН. 2012. Том 23. С. 215—244.
7. Нокель М. А. Метод учета структуры биграмм в тематических моделях // Вестник ВГУ. Серия: Системный анализ и информационные технологии. 2014. № 4. С. 89—97.
8. Бузикашвили Н. Е., Самойлов Д. В., Крылова Г. А.  $N$ -граммы в лингвистике // Методы и средства работы с документами. Сборник. М.: Эдиториал УРСС, 2000. С. 91—130.
9. Харламов А. А. Система для автоматического смыслового анализа текстов. URL: <http://www.analyst.ru/index.php?lang=rus&dir=content/downloads/>. (дата обращения 20.12.2014 г.).
10. Харламов А. А. Модель мира человека — семантическая сеть // Тезисы доклада на семинаре, посвященном памяти С. П. Капицы, 2013. URL: <http://spkurdyumov.ru/network1/model-mira-cheloveka-semanticheskaya-set/>.
11. Харламов А. А., Ермоленко Т. В., Жонин А. А. Моделирование динамики процессов последовательностью однородных семантических сетей на основе анализа последовательности описывающих процесс текстовых выборок // Труды Международной научно-технической конференции "Открытые семантические технологии проектирования интеллектуальных систем" (OSTIS'2014), Минск, 2014. С. 161—168.
12. Харламов А. А. Нейросетевой тест на вхождение в сетевое сообщество // Экономические стратегии. 2014. № 9. С. 121—125.
13. Харламов А. А. Информационная модель личности // Труды Международной научно-технической конференции "Открытые семантические технологии проектирования интеллектуальных систем" (OSTIS'2015), Минск, 2015. С. 529—532.
14. Куликов А. М., Харламов А. А. Использование однородной семантической сети для классификации результатов генетического анализа // Труды OSTIS'2016, Минск, 2016.
15. Харламов А. А., Рыжов В. А., Курдюмов В. С. Синхронизация предметных областей коммуникантов в процессе диалога // Экономические стратегии, N 6, 2016. С. 124—136.

## Associative Memory as a Neuronetwork Environment of Combined Knowledge Space

*An associative human memory is an environment for combined knowledge space development. Text information processing as an example of any modality information processing by human is shown. The results of the process are a homogenous network of key text concepts which are weighed by their rank in text and hierarchical thematic structure tree which characterize the text complexity. The results are interpreted clearly by certain text analysis using thematic modelling approach. The n-gram text model is used instead of traditional unigram (bag of words) one.*

**Keywords:** complexity, corticoform associative memory, structured information processing, associative (homogenous semantic) network, thematic hierarchy tree, multimodal information, thematic text analysis, n-gram text model

### References

1. Kharlamov A. A., Yermolenko T. V. Neurosetevaya sreda (neuromorphnaya associativnaya pamyat') dlya preodoleniya informatsionnoy slozhnosti. Poisk smysla v slabostrukturirovannykh massivakh informatsii. Chast' I. Structurnaya obrabotka informatsii v kore, *Informatsionnye tekhnologii*, 2015, vol. 21, no. 11, pp. 814–820.
2. Kharlamov A. A., Yermolenko T. V. Neurosetevaya sreda (neuromorphnaya associativnaya pamyat') dlya preodoleniya informatsionnoy slozhnosti. Poisk smysla v slabostrukturirovannykh massivakh informatsii. Chast' II. Obrabotka informatsii v gippokampe. Model' mira, *Informatsionnye tekhnologii*, 2015, vol. 21, no. 12, pp. 883–889.
3. Kharlamov A. A. *Neurosetevaya tekhnologiya predstavleniya i obrabotki informatsii (estestvennoye predstavlenie znaniy)*, Moscow, Radiotekhnika 2006.
4. Kharlamov A. A., Yermolenko T. V. Analiz tekstov: lingvistika, semantika, pragmatika v ramkakh kognitivnogo podkhoda, *Problemy iskusstvennogo intellekta*, 2015, no. 0 (1), pp. 107–115.
5. Kharlamov A. A. *Formirovaniye n-grammnoi tematicheskoy modeli teksta*. 2016. no. 1–2, (v pechati).
6. Korshunov A., Gomzin A. Tematicheskoye modelirovaniye tekstov na estestvennom yazyke, *Trudy Instituta sistemnogo programmirovaniya RAN*, 2012, vol. 23, pp. 215–244.
7. Nokel' M. A. Metod uchya struktury bigram v tematicheskikh modelyakh, *Vestnik VGU, Seriya: Systemny analiz i informatsionnye tekhnologii*, 2014, no. 4, pp. 89–97.
8. Buzikashvily N. E., Samoylov D. V., Krylova G. A. N-grammy v lingvistike, Sbornik: *Metody i sredstva raboty s dokumentami*, Moscow, Editotial URSS. 2000. pp. 91–130.
9. Kharlamov A. A. Systema dlya avtomaticheskogo smyslovogo analiza tekstov, available at: <http://www.analyst.ru/index.php?lang=rus&dir=content/downloads/>
10. Kharlamov A. A. Model' mira cheloveka — semanticeskaya set', Tezisy doklada na seminare, posvyachennom pamyaty S. P. Kapitsy, 2013. URL://spkurdyumov.ru/networks/model-mira-cheloveka-semanticeskaya-set/
11. Kharlamov A. A., Yermolenko T. V., Shonin A. A. Modelirovaniye dinamiky processov posledovatel'nostyu odnorodnykh semanticeskikh setey na osnove analiza posledovatel'nosti opisyyvayuchikh process tekstovikh vyborok., *Trudy meshdunarodnoy nauchno-tekhnicheskoy konferencii "Otkrytiye semanticeskoye tekhnologii proektirovaniya intellektual'nykh system"*. (OSTIS'2014), Minsk, 2014, pp. 161–168.
12. Kharlamov A. A. Neyrosetevoy test na vkhozhdenie v neurosetevoye soobchestvo, *Ekonomicheskoye strategii*, 2014, no. 9, pp. 121–125.
13. Kharlamov A. A. Informatsionnaya model' lichnosti, *Trudy meshdunarodnoy nauchno-tekhnicheskoy konferencii "Otkrytiye semanticeskoye tekhnologii proektirovaniya intellektual'nykh system"*. (OSTIS'2015), Minsk, 2015, pp. 529–532.
14. Kulikov A. M., Kharlamov A. A. Ispol'zovaniye odnorodnoy semanticeskoy sety dlya klassifikacii rezultatov geneticheskogo analiza, *Trudy meshdunarodnoy nauchno-tekhnicheskoy konferencii "Otkrytiye semanticeskoye tekhnologii proektirovaniya intellektual'nykh system"*. (OSTIS'2016), Minsk, 2016.
15. Kharlamov A. A., Ryshov V. A., Kurdiymov V. A. Synkhronizatsiya predmetnykh oblastei kommunikantov v processe dialoga, *Ekonomicheskoye strategii*, no. 6, 2016, pp. 124–136.

ИНФОРМАЦИЯ

Статья «Корректировка методических и случайных составляющих погрешностей вычисления коэффициентов корреляции, возникающих на малых выборках биометрических данных» (авторы Кулагин В. П., Иванов А. И., Серикова Ю. И.), опубликованная в № 9 2016 года, подготовлена на основе материалов научно-исследовательской работы, имеющей финансовую поддержку государства в лице Минобрнауки России (уникальный идентификатор RFMEFI60815X0003).

**В. В. Пекунов**, д-р техн. наук, инженер-программист, e-mail: pekunov@mail.ru,  
ОАО "Информатика", г. Иваново

## Извлечение информации из нейронных сетей прямого распространения в виде простых алгебраических моделей

*Рассматривается построение упрощенных аналитических моделей нейронных сетей при некотором снижении точности аппроксимации. Передаточные функции нейронов замещаются полиномиальными соотношениями, справедливыми на рабочих участках нейронов. Учитываются значимости отдельных интервалов рабочих участков. Предложено использовать стратегию невозрастания максимальной степени входных переменных в конечном выражении для получения более компактных соотношений. Комбинирование конечных соотношений выполняется с помощью разработанного ядра символической математики. Ядро использует ускоренные алгоритмы расчета, в том числе параллельные вычисления. Показаны приемлемые ускорение и эффективность распараллеливания на 16-ядерной машине платформы Google's Compute Engine.*

**Ключевые слова:** нейронная сеть, аналитическая модель, упрощение, символическая математика, параллельный расчет

### Введение

Искусственные нейронные сети [1] известны своими неплохими интерполяционными способностями, позволяющими строить приближенные модели как одномерных рядов, так и многомерных данных. Однако такие модели, по сути, близки к "черным ящикам", поскольку их непосредственная математическая интерпретация при наличии нелинейных передаточных функций (например, классических сигмоид) отличается непрозрачностью и не дает, фактически, возможности простой аналитической доработки общей функции, заставляя снова и снова обращаться к тому же аппарату нейронных сетей, варьируя их архитектуру (число слоев и нейронов в слоях) для повышения точности результата. Актуальна проблема получения структурно более простых и прозрачных аналитических моделей, эквивалентных исходной нейронной сети, по меньшей мере, на той же обучающей выборке. Такой эффект может быть достигнут путем решения не менее чем двух задач: а) первичного упрощения исходной сети; б) анализа поведения передаточных функций каждого отдельного нейрона на его входных данных с заменой этих функций на более простые для анализа. Интересна также задача автоматического комбинирования полученных после замены функций выражений в единую формулу с ее последующим (также автоматическим) упрощением. Поскольку при рассмотрении нейронных сетей больших размерностей процессы символического комбинирования конечной формулы и перебора скомбинированных вариантов могут оказаться весьма трудозатратными (особенно в операции деления полинома на полином), то актуальны следующие задачи: а) выработка стратегий, позволяющих за счет некоторого снижения качества упрощения существенно повысить скорость генерации формулы;

б) распараллеливание вычислений для многоядерной системы.

В данной работе рассматривается случай сети с одним выходом, заметим только, что все нижеследующие методики и алгоритмы легко обобщаются на случай многих выходов, необходимо лишь рассматривать процесс комбинирования символической формулы для каждого выхода (для каждого нейрона выходного слоя) по отдельности.

### Общий подход к упрощению сети

Первичное упрощение сети можно осуществить путем последовательного исключения из нее связей и нейронов, пока погрешность не достигнет заданного исследователем уровня. Для этого организуется итерационная процедура, на каждом этапе для каждой связи определяется ее значимость как разность ошибок приближения исходных данных после и до изъятия связи (изъятие осуществляется приравниванием соответствующего веса связи нулю). Выбирается связь с наименьшей значимостью, и соответствующий коэффициент уже постоянно приравнивается нулю. Если обнаруживается нейрон, у которого все входные связи исключены, то он изымается из сети вместе со всеми своими связями. Процедура продолжается до тех пор, пока конечная ошибка приближения исходных данных (после изъятия очередной связи с наименьшей значимостью) не превысит некоторый определенный предел.

Замена передаточных функций на более простые для анализа является обычной задачей регрессии, решение которой может быть осуществлено методом наименьших квадратов. Целесообразно ограничиться такими функциями, для которых данный метод сводится к однократному решению системы линейных алгебраических уравнений, причем сами функции должны быть способны эффективно при-

ближать хотя бы отдельные "рабочие" участки передаточных функций. Исходя из этого, были выбраны следующие четыре вида заместительных функций:

$$f(s) = \sum_{i=0}^P a_i s^i;$$

$$f(s) = \sum_{i=0}^P a_i s^{-i};$$

$$f(s) = \sum_{i=0}^P a_i s^{0,5i};$$

$$f(s) = \sum_{i=0}^P a_i s^{-0,5i},$$

где параметр  $P$  выбирается минимально возможным при условии достижения требуемой точности приближения передаточных функций.

Использовался метод наименьших квадратов (МНК) с весами в целевой функции. Рабочая область значений аргумента  $s$  заместительной функции  $f(s)$  определялась динамически при вычислении значений отклика сети на заданной обучающей выборке  $(\mathbf{x}_i; y_i)$ ,  $i = \overline{1, N}$ , где  $\mathbf{x}_i$  — вектор значений входов;  $y_i$  — значение выхода;  $N$  — объем выборки. Рабочая область делилась на некоторое число равновеликих участков, для каждого из которых подсчитывалось число элементов обучающей выборки, дающих значение аргумента передаточной функции, попадающее в этот участок. Эти частоты попадания и выступали в роли весов элементов  $w_i$  обучающей выборки в целевой функции МНК:

$$\sum_{i=0}^N w_i (y_i - f(s(\mathbf{x}_i)))^2 \rightarrow \min.$$

В общем случае, с учетом того, что заместительные функции являются лишь промежуточными, а конечной будет их комбинация, для которой достаточно сложно изначально определить, заместительные функции какого вида предпочтительнее для отбора, для каждого нейрона целесообразен параллельный (с применением OpenMP) отбор сразу нескольких вариантов (в дальнейшем будут перебираться все комбинации таких, найденных для различных нейронов, вариантов) функции, руководствуясь не только критерием наилучшего приближения рабочего участка передаточной функции, но и какими-либо простыми стратегиями получения возможно более оптимальной конечной комбинации.

Примером такой стратегии может быть невозрастание свыше определенного предела максимальной степени входных переменных в конечном выражении. Мы рассматривали трехслойные сети с

линейным третьим слоем, в которых данная стратегия может быть реализована путем определения большей толерантности выбора упрощенных функций (типа *REV*) с отрицательными степенями во втором слое, если в первом преобладают функции (типа *DIR*) с положительными степенями, и наоборот, повышения толерантности выбора функций с положительными степенями во втором слое, если в первом преобладают функции с отрицательными степенями. Под толерантностью подразумевается допустимое относительное отклонение анализируемого варианта функции от наилучшего варианта функции. Толерантность  $\varepsilon_{\text{тол}}$  для анализируемой во втором слое функции типа  $TYPE \in \{DIR, REV\}$  можно представить следующим образом:

$$\varepsilon_{\text{тол}} = \varepsilon_{\text{мин}} + \alpha \left( 1 - \frac{N_{TYPE}}{N_{DIR} + N_{REV}} \right),$$

где  $\varepsilon_{\text{мин}}$  — минимально возможная толерантность;  $\alpha$  — параметр;  $N_{DIR}$  — число отобранных в первом слое функций с положительными степенями;  $N_{REV}$  — число отобранных в первом слое функций с отрицательными степенями;  $N_{TYPE}$  — число отобранных в первом слое функций типа *TYPE*.

### Символическое упрощение

Для автоматического комбинирования полученных функций с последующим упрощением полученного выражения можно использовать любое ядро символьной математики, например, пакета Maple. Кроме того, автором разработано простое быстродействующее ядро для таких преобразований, написанное на C++, отличительными чертами которого являются:

а) ускоренный параллельный расчет (с применением OpenMP) наиболее трудозатратной операции поиска очередного члена частного при делении полинома  $A = \sum_i A_i$  на полином  $B = \sum_j B_j$  (рас-

параллеливается цикл по слагаемым делителя  $B_j$ , в котором определяется очередной член делимого  $A_i$ , такой, что в остатке от деления  $A - (A_i/B_j)B$  будет наименьшая сумма степеней переменных);

б) контроль времени поиска очередного члена делимого  $A_i$ , реализуемого в цикле, упомянутом в пункте а);

в) ускоренный бинарный поиск подобных слагаемых по равенству специальных битовых масок, в которых кодируются степени входящих в слагаемое переменных. Маски выступают в роли ключа стандартной *map*-структуры [2], а в роли значений выступают сами слагаемые.

Для контроля времени поиска очередного члена делимого  $A_i$  при заданном  $B_j$  предлагается (чтобы не вычислять остаток  $A - (A_i/B_j)B$  для всех  $A_i$ ) ограничить перебор неким множеством  $i \in V$ , где

$V \subseteq \{k | k = \overline{1, M}\}$ , а  $M$  — число членов  $A_i$ . Предлагается следующий алгоритм.

Пусть  $V := \emptyset$ ,  $C := -\infty$ .

Цикл по  $i = \overline{1, M}$ .

Находим  $D := A_i/B_j = \prod_r v_r^{g_r}$ , где  $v_r$  — переменная, соответствующая одному из входов сети.

Находим  $F := \sum_r g_r$ .

Если  $F \in [C - d; C + d]$ , то  $V := V \cup \{i\}$ .

Иначе, если  $F > C + d$ , то  $V := \{i\}$ ,  $C := F$ .

Конец цикла.

В описании алгоритма присутствует  $d$  — допустимое отклонение значения от лучшего из вариантов. В начале выполнения операции деления полинома на полином целесообразно установить значение  $d$ , равным максимально возможной сумме степеней при переменных в членах делимого/делителя, обозначим ее  $d_{\max}$ . Далее для каждого очередного  $B_j$  замеряется время  $t$  поиска требуемого  $A_i$ , которое сравнивается с допустимым  $t_{\text{доп}}$ . Если  $t > t_{\text{доп}}$ , то  $d := \max(0; d - 1)$ , если  $t < t_{\text{доп}}$ , то  $d := \min(d_{\max}; d + 1)$ . Новое значение  $d$  используется для следующего  $B_j$ .

### Апробация предложенной методики упрощения

Приведем два примера полученных упрощенных выражений для трехслойной сети (три сигмоидальных нейрона — в первом слое, два сигмоидальных нейрона — во втором слое, один линейный — в третьем слое) с двумя входами, описывающей изменение турбулентной вязкости (соответственно,  $v_{\text{турб}}^1$  и  $v_{\text{турб}}^2$ ) в зависимости от квадрата расстояния до твердой стенки  $a$  и оценки значения турбулентной вязкости  $b$  [3]. Входные и выходные данные предварительно были интерполированы к интервалу  $[0; 1]$ . При обучении сети использовали данные, полученные с помощью модели турбулентности К-Е-RNG (см., например, [4]) для обтекания одиночного небольшого здания, находящегося между двумя многоэтажными сооружениями. На рис. 1 (см. третью сторону обложки) показаны схематическая постановка задачи (а) и полученное в численном эксперименте распределение турбулентной вязкости (б):

$$v_{\text{турб}}^1 = \frac{1,027 - 0,163a - 0,0034a^2 + 9,29b + 0,26ab - 6,35b^2 - 0,62z}{1,988 - 0,14a - 0,003a^2 + 8b + 0,224ab - 5,47b^2 - 0,53z - 6,49a + 2,04a^2 - 0,099b - 0,0006ab + 0,049b^2 - 0,012z},$$

$$v_{\text{турб}}^2 = 0,431 - 0,028a - 0,00056a^2 + 1,48b + 0,044ab - 1,03b^2 - 0,12z;$$

$$z = \sqrt{0,66 + 1,27a - 0,61b}.$$

Здесь при анализе использовали заместительные функции вплоть до третьего порядка ( $1 \leq P \leq 3$ ). Регулировку толерантности осуществляли при

$$\varepsilon_{\text{мин}} = 10^{-3}, \alpha = 4 \cdot 10^{-2}.$$

Легко подсчитать, что для расчета  $v_{\text{турб}}^1$  и  $v_{\text{турб}}^2$  требуется не менее 39 и 17 одноклаковых команд современных процессоров архитектуры x86, тогда как для расчета двухвходовой нейронной сети конфигурации  $3 \times 2 \times 1$  (с активационной функцией "экспоненциальная сигмоида" в первых двух слоях) требуется не менее 56 одноклаковых команд (для вычисления каждой экспоненты требуется не менее трех команд). Таким образом, даже теоретический выигрыш в скорости расчета достаточно очевиден.

Проведенные численные эксперименты по моделированию обтекания одиночного здания с применением как полноценной трехвходовой (на третий вход подавалась функция деформации, см. [3]) нейронной сети конфигурации  $4 \times 3 \times 1$ , описывающей турбулентную вязкость, так и полученных (замещающих сеть) упрощенных соотношений, показали, что при сохранении приемлемой точности (средняя относительная погрешность нейронной сети по отношению к модели К-Е (RNG) составила около 20 %, средняя относительная погрешность упрощенных соотношений — около 23 %), упрощенные соотношения обеспечивают общее ускорение (при расчете системы из 3—5 дифференциальных уравнений в частных производных) в 1,06—1,08 раза, ускорение расчета исключительно турбулентной вязкости составляет около 1,8 раза (на 45 %). На рис. 2 (см. третью сторону обложки) показаны значения турбулентной вязкости, полученные в экспериментах с нейронной сетью (а) и с упрощенными выражениями (б). Очевидно, что значения согласуются как качественно, так и количественно, что подтверждает корректность предложенных методик и алгоритмов. Более того, упрощенные модели успешно выдержали проверку по критерию Фишера.

### Общая схема распараллеливания. Показатели распараллеливания

Как уже было указано выше, при поиске очередного члена частного деления полинома на полином использовано векторное распараллеливание по виткам цикла. В то же время для процесса перебора вариантов комбинаций заместительных функций было применено распараллеливание по идеологии "портфеля задач" [5] (с применением порождения задач в OpenMP), в соответствии "один вариант — одна задача". Такой комбинированный вариант распараллеливания потребовал гибкого разделения множества процессоров на несколько подмножеств, каждому из которых соответствует

Таблица

Данные о времени счета и показателях распараллеливания

Число ядер	Время счета, с	Показатели распараллеливания	
		Ускорение	Эффективность
1	38,68	1	1
2	28,33	1,37	0,68
3	23,29	1,66	0,55
4	15,89	2,43	0,61
5	13,59	2,85	0,57
6	13,13	2,94	0,49
7	12,42	3,11	0,45
8	9,87	3,91	0,49
9	10,85	3,56	0,4
10	10,41	3,72	0,37
11	9,92	3,9	0,35
12	9,57	4,04	0,34
13	8,79	4,4	0,34
14	7,92	4,88	0,35
15	8,99	4,3	0,29
16	8,4	4,6	0,29

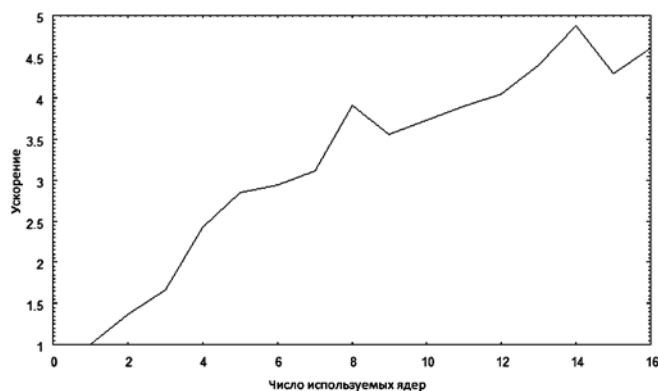


Рис. 3. Ускорение в зависимости от числа используемых ядер

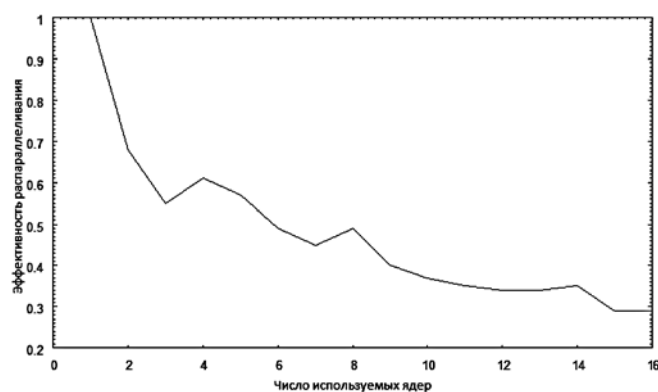


Рис. 4. Эффективность распараллеливания в зависимости от числа используемых ядер

одна задача, а уже в пределах подмножества применен векторный параллелизм. Это позволило достаточно равномерно загрузить процессорные ядра и добиться более высокой степени ускорения. В частности, если принять за единицу время расчета при использовании только векторного параллелизма, то даже для задач небольшой размерности время счета при использовании только "портфеля задач" составляет уже 0,96, а при комбинированном варианте распараллеливания — от 0,78 до 0,81 (для задач большой размерности — от 0,5 до 0,73).

В таблице приведены данные о времени счета при использовании различного числа ядер 16-ядерной машины платформы Google's Compute Engine, также проведены расчеты ускорения вычислений (рис. 3) и эффективности распараллеливания (рис. 4).

Учитывая, что использовались не реальные, а HyperThreading-ядра, это весьма высокие результаты.

### Заключение

Таким образом, в данной работе предложен подход к получению сравнительно простых алгебраических моделей, являющихся приближенным описанием нейросетевых моделей. Такие модели могут быть менее затратны с вычислительной точки зрения, кроме того, их открытость и простота говорят о целесообразности их принятия за основу эмпирических или полуэмпирических моделей изучаемого явления, иных, нежели нейронные сети. Разработана программа, извлекающая алгебраические модели из нейронных сетей, на базе предложенного автором ускоренного символического ядра. Адекватность извлекаемых моделей подтверждена на примере нейросетевого моделирования турбулентной вязкости. Эффективность распараллеливания на 16-ядерной машине платформы Google's Compute Engine составила от 29 до 68 % в зависимости от числа используемых ядер.

### Список литературы

1. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. М.: Мир, 1992. 184 с.
2. Остерн М. Г. Обобщенное программирование и STL: Использование и наращивание стандартной библиотеки шаблонов C++. СПб.: Невский Диалект, 2004. 544 с.
3. Пекунов В. В. Локальные нейросетевые модели турбулентности // Мат. Междунар. науч.-техн. конф. "XVIII Бенардосовские чтения". Иваново, 2015. Т. 2. С. 331—334.
4. Пекунов В. В. Новые методы параллельного моделирования распространения загрязнений в окрестности промышленных и муниципальных объектов. Дис. ... докт. тех. наук. Иваново, 2009.
5. Эндриус Г. Р. Основы многопоточного, параллельного и распределенного программирования. М.: Вильямс, 2003. 512 с.

## The Derivation of the Information from Artificial Feed-Forward Neural Networks in the Form of the Simple Algebraic Models

*In this work the building of the simplified analytical models of the artificial neural networks with some diminishing of accuracy is investigated. The activation functions of the neurons are replaced by polynomial expressions which are accurate at the work areas of the neurons. The importance of the different intervals of work areas is taken into account. It is proposed to use the strategy of non-increasing the maximal power of input variables in the result to obtain the more compact solution. The result combining is performed by means of the developed symbolic math kernel. This kernel uses accelerated algorithms including the parallel computations. An appropriate acceleration and effectiveness of the parallelizing are achieved on the 16-core machine of the Google's Compute Engine platform.*

**Keywords:** artificial neural network, analytical model, simplifying, symbolic math, parallel computations

### References

1. **Uossermen F.** *Nejrokomputernaja tehnika: Teorija i praktika* [Neural Computing: Theory and Practice], Moscow, Mir, 1992, 184 p.
2. **Ostern M. G.** *Obobshhennoe programirovanie i STL: Ispol'zovanie i narashhivanie standartnoj biblioteki shablonov C++* [Generic programming and the STL: using and extending the C++ Standard Template Library], Sankt-Petersburg, Nevskij Dialekt, 2004, 544 p.
3. **Pekunov V. V.** Lokal'nye neyrosetevye modeli turbulentnosti [Local neural network models of turbulence], *Mat. Mezhdunar. nauch.-tekhn. konf. "XVIII Benardosovskie chteniya"* [Proceedings of the International Scientific-Technical Conference "XVIII Benardos readings"]. Ivanovo, 2015, vol. 2, pp. 331–334.
4. **Pekunov V. V.** Novye metody paralel'nogo modelirovaniya rasprostraneniya zagryazneniy v okrestnosti promyshlennykh i munitsipal'nykh ob'ektov [New methods of parallel modeling of pollution propagation near the plants and city objects] // Dis. dokt. tekhn. nauk [Thesis of doctor of engineering], Ivanovo, 2009.
5. **Jendrus G. R.** Osnovy mnogopotochnogo, paralel'nogo i raspredelenного programirovanija [Foundations of Multithreaded, Parallel, and Distributed Programming]. Moscow, Vil'jams, 2003, 512 p.

---

---

### Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *Е. В. Комиссарова*.

Сдано в набор 10.11.2016. Подписано в печать 23.12.2016. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ117. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".  
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.