ISSN 2220-3397

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Программная инженерия

ПрИн
4
2021
Том 12

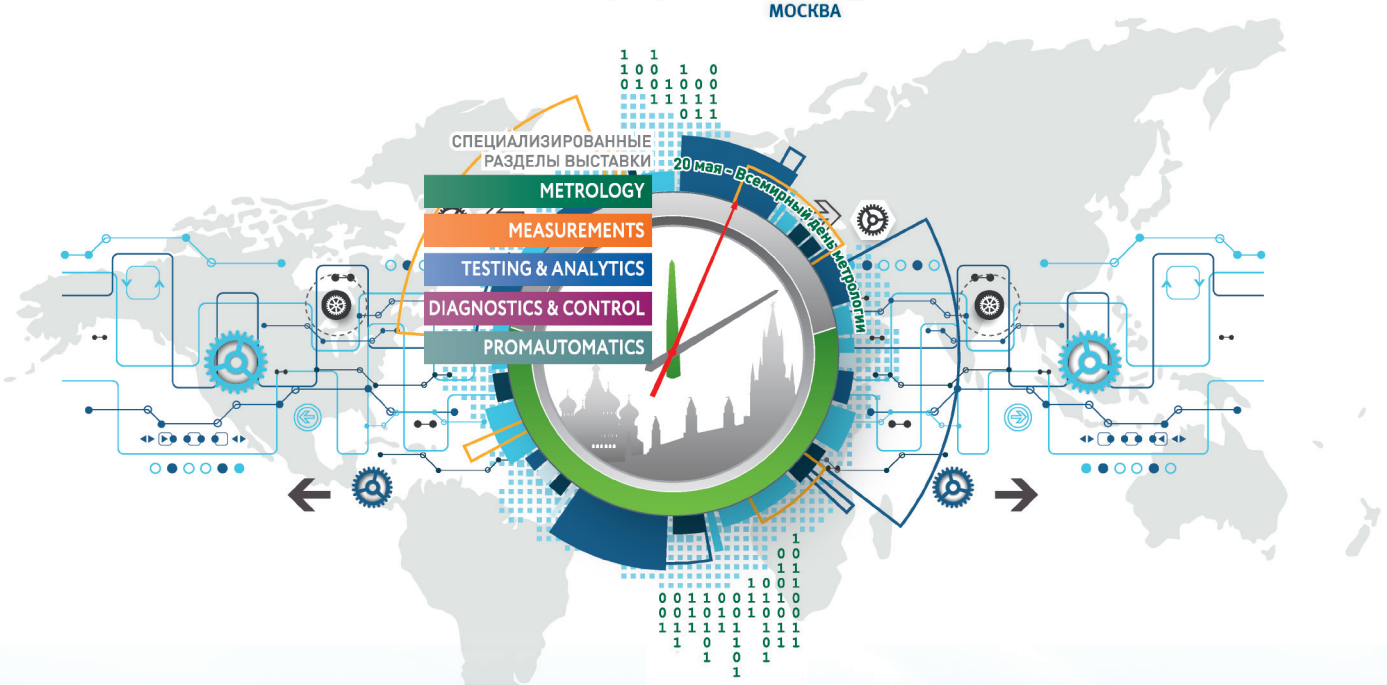
17-Я СПЕЦИАЛИЗИРОВАННАЯ ВЫСТАВКА

ТОЧНЫЕ ИЗМЕРЕНИЯ — ОСНОВА КАЧЕСТВА И БЕЗОПАСНОСТИ

MetrolExpo’2021

18–20 октября

ЭКСПОЦЕНТР
МЕЖДУНАРОДНЫЕ ВЫСТАВКИ И КОНГРЕССЫ
МОСКВА



ТЕМАТИКА ВЫСТАВКИ:

МЕТРОЛОГИЯ
ИЗМЕРЕНИЯ
ИСПЫТАНИЯ И АНАЛИТИКА
ДИАГНОСТИКА И КОНТРОЛЬ
ПРОМЫШЛЕННАЯ АВТОМАТИЗАЦИЯ

В РАМКАХ РОССИЙСКОЙ ПРОМЫШЛЕННОЙ НЕДЕЛИ

Синергия 6-и выставок
19.000 посетителей
600 участников
Экспозиция 17.000 м^2

ВК «ВЭСТСТРОЙ ЭКСПО»
Телефон: +7 (495) 937-40-23
E-mail: metrol@expoprom.ru
www.metrol.expoprom.ru

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с сентября 2010 г. DOI 10.17587/issn.2220-3397 ISSN 2220-3397

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана

СОДЕРЖАНИЕ

Журнал зарегистрирован в Федеральной службе по надзору в сфере связи, информационных технологий и массовых коммуникаций. Свидетельство о регистрации ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индекс по Объединенному каталогу "Пресса России" — **22765**) или непосредственно в редакции.
Тел.: (499) 269-53-97. Факс: (499) 269-55-10.
Http://novtex.ru/prin/rus E-mail: prin@novtex.ru
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.

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

SOFTWARE ENGINEERING

PROGRAMMNAYA INGENERIA

Vol. 12
N 4
2021

Published since September 2010 **DOI 10.17587/issn.2220-3397** **ISSN 2220-3397**

CONTENTS

Information about the jornal is available online at:
http://novtex.ru/prin/eng e-mail:prin@novtex.ru

УДК 519.72 DOI: 10.17587/prin.12.171-178

С. А. Букашкин, д-р техн. наук, проф., ген. конструктор, sergey.bukashkin@gmail.com, АО "Концерн "Автоматика", Москва,
М. А. Черепнёв*, д-р физ.-мат. наук, доц., cherepniov@gmail.com, МГУ им. М. В. Ломоносова

Квантовый компьютер и постквантовая криптография

Представлен обзор современного состояния проблемы построения квантового компьютера и его гипотетического использования для взлома криптографических протоколов. Рассмотрены необходимые для этого параметры. Представлен обзор существующих квантовых алгоритмов и стойких относительно них постквантовых криптографических протоколов. Проблема построения квантового компьютера рассмотрена в сравнении с развитием теории и практики обычных механических и электронных компьютеров. Приведены результаты конкурсов по тематике постквантовой криптографии.

Ключевые слова: *квантовый компьютер, квантовые алгоритмы, постквантовая криптография*

Введение

В сентябре 1979 г. в г. Ургенч на конференции, посвященной теории вычислений, Ю. И. Манин обратил внимание на принципиальное отсутствие возможности описания биологических автоматов механическими терминами. Например, репликация ДНК формально требует разворачивания и сворачивания молекулы ДНК в течение 20 мин со скоростью примерно 125 оборотов в секунду, чтобы реализовать 300 000 оборотов, на которые она обычно закручена. Поэтому он анонсировал необходимость создания математической теории квантовых автоматов, основой которых будут использование суперпозиции, отсутствие однозначного разделения квантовой системы на элементы, описание взаимодействия Эрмитовыми операторами (матрица обратного преобразования является транспонированной и сопряженной к исходной) и вероятностными терминами. Он считал, что "математическое описание квантового автомата должно быть абстрактным, не предрешая физических реализаций". С точки зрения теории вычислений принцип работы, реализованный в квантовом компьютере, был сформулирован Ю. И. Маниным в 1980 г. в работе [1]. Исследуя некоторые квантово-механические эффекты Р. Фейнман [2] в 1982 г. пришел к аналогичным выводам. В дальнейшем вместо бит как единиц хранения информации стали рассматривать так называемые кубиты (или квантовые биты), способные принимать уже не два, а большее число состояний, обычно представляемых на так называемой сфере Блоха. Физически это могут быть атомы, находящиеся в возбужденном состоянии, или другие физические объекты, поглощающие и отдающие энергию фиксированными порциями, квантами. Реализовать работу кубитов на практике оказалось сложной задачей.

* Работа этого автора поддержана грантом РФФИ № 18-29-03124/20.

Введем несколько определений.

Квантовый процессор — вычислительное устройство, в основе которого лежат Эрмитовы унитарные операции с кубитами. Например, n-разрядный квантовый регистр может хранить 2^n значений в одном месте, а квантовый процессор может все эти значения одновременно обрабатывать [3, 4].

Квантовый компьютер — вычислительное устройство на основе квантовых процессоров, способное работать по программе.

В отличие от квантового процессора квантовый компьютер трудно реализовать. Существующие реализации пока носят лабораторный характер. Камнем преткновения является проблема исправления ошибок, связанная с невозможностью копирования состояния, а также вытекающая из этого неустойчивость физических устройств и вычислений.

Важно отметить, что элементарные операции в квантовом компьютере (квантовая арифметика) связаны с тригонометрическими суммами, и поэтому принципиально отличаются от операций обычной арифметики.

Основы квантовых вычислений были заложены Д. Дойчем в 1985 г. в работе [5]. Однако за истекшие 35 лет квантовый компьютер построен не был. Есть рабочие модели квантовых процессоров [6], работающих с некоторыми ошибками, которые пока не удается исправить. Эти ошибки могут быть связаны с внешним шумом, работой соседних кубитов и попытками измерения состояния кубита или сравнения состояний разных кубитов. Вследствие этого в квантовых вычислениях происходят ошибки, и дальше эволюция квантового процессора становится уже неконтролируемой.

Применение известной технологии кодирования с последующим исправлением ошибок осложнено невозможностью копирования состояния кубита. Если $1/k^s$ — это вероятность успешного срабатывания одного кубита в s шагах программы для n-кубитного

процессора, вычисляющего функцию от n-битного аргумента, то для получения хотя бы одного правильного результата понадобится в среднем сделать k^{sn} квантовых шагов. При $k^s > 2$ это больше, чем число всех возможных аргументов данной задачи. Поэтому значительного выигрыша по сравнению с обычным компьютером, последовательно обрабатывающим эти аргументы, нет. По всей видимости, процедура исправления ошибок должна быть также квантовой.

Отметим, что ситуация с квантовым компьютером сейчас отличается от ситуации с обычным компьютером в середине XX века и ранее. Первые компьютеры хоть и медленно, но работали (например, суммирующая машина Паскаля 1642 г. осуществляла арифметические преобразования пятизначных — десятичных чисел, разностная машина Беббиджа 1822 г. вычисляла значения многочленов до 7-й степени, ЭВМ "Bombe" Тьюринга 1940 г. осуществляла взлом шифровальной машины Enigma), а первые квантовые — нет.

Рассмотрим несколько определений, важных для дальнейшего изложения.

Квантовая машина Тьюринга — это такая машина Тьюринга, которая на каждом шаге осуществляет Эрмитово унитарное преобразование (обратная матрица совпадает с транспонированной и сопряженной) конечномерного вектора, записанного на ее бесконечной ленте. Формально это означает, что на ленту можно записать все возможные значения аргумента и параллельно вычислять значения функции от всех этих аргументов. Если ставить своей целью обращение функции, то потом формально надо будет еще найти нужное значение, т. е. перебрать все результаты.

Поскольку шаг работы квантовой машины — это обратимое преобразование, то вычислимыми на квантовой машине Тьюринга могут быть только функции, сводящиеся к вычислению функций из некоторого подкласса взаимно однозначных функций.

Квантовый компьютер использует такие физические объекты "кубиты", которые позволяют хранить в одном месте все возможные значения аргумента. Тем самым не надо перебирать результаты для поиска значения, на котором нужно обратить рассматриваемую одностороннюю функцию.

Квантовый параллелизм — метод, с помощью которого некоторые вероятностные задачи могут быть выполнены универсальным квантовым компьютером быстрее, чем с помощью любого классического. Квантовый параллелизм — принцип, лежащий в основе работы квантовых компьютеров и потенциально позволяющий им превзойти в производительности классические компьютеры на некоторых задачах. В основе квантового параллелизма лежит использование при вычислениях суперпозиций базовых состояний, что позволяет одновременно проводить большое количество вычислений с различными исходными данными. Тем не менее извлечение результатов таких вычислений затруднено, что ограничивает область применения квантовых компьютеров [3]. Трудность заключается и в том, что измерение полученных значений приводит к изменению самих значений. Поэтому пока не удается добиться от квантового вычислителя безошибочного выполнения хотя бы простейших операций.

Отметим, что поскольку шаг работы квантового компьютера есть применение обратимого оператора, то для взаимно однозначных функций вычисление функции и ее обращение выполняются одновременно. Таким образом, в смысле квантовых вычислений нет односторонних взаимно однозначных функций. Как следствие, для построения криптографических схем, стойких относительно квантовых вычислителей, предпочтительно выбирать принципиально не взаимно однозначные отображения. Можно использовать взаимно однозначные отображения, вычислимые на обычном, но не вычислимые на квантовом компьютере за полиномиальное время.

В целом квантовый компьютер следует отнести к категории спецвычислителей (ранее известны векторные машины, SAT-solvers), основанных на природных (физических, биологических) принципах или автоматах, которые некоторые задачи могут решать эффективнее обычных компьютеров, а для некоторых других не могут быть эффективно применены. В работе [7] показано, что в модели "черных ящиков" квантовый компьютер может отличить некоторое специально построенное распределение на N битах от равномерного за время порядка $O(\log N)$ с вероятностью порядка $1/\log N$, а для классического компьютера (булева схема конечной глубины и размера, квазиполиномиально зависящего от N) таких оценок достичь невозможно. При этом квантовый компьютер строится как обычный компьютер, усиленный кубитами. Аналогичный пример — задача Саймона (1994 г.) "распознавания существования универсальной коллизии" [8].

Однако следует отметить, что приведенные примеры носят специальный характер, а для задач вскрытия многих криптосхем квантовых алгоритмов пока не построено.

Поскольку к описанию работы кубитов привлекается статистическая физика, то приходится использовать определения, подобные следующему: две вычислительные машины вычислительно эквивалентны (при заданных маркировках), если в любом возможном эксперименте или последовательности экспериментов, в которых их входные данные были подготовлены эквивалентно (при заданных маркировках), измеренные значения выходных наблюдаемых величин для двух машин статистически неразличимы. Таким образом, функции распределения вероятностей для выходов двух машин совпадают.

Вообще универсальный квантовый компьютер мыслится как вычислительный аппарат, который может моделировать любую конечно реализуемую физическую систему (тезис Черча—Тьюринга—Дойча, опубликованный в работе [5]).

Квантовые (взаимно однозначные) схемы из функциональных элементов рассмотрены в работе [9].

Квантовые алгоритмы

Не все полиномиальные алгоритмы для обычных вычислителей могут быть легко перенесены на операции с кубитами в квантовом компьютере.

Однако в конце прошлого века были написаны полиномиальные алгоритмы для квантовых компьютеров, решающие ранее не решаемые на классическом компьютере за полиномиальное время задачи. К числу самых важных из их числа, по мнению авторов, можно отнести перечисленные далее.

Алгоритм Дойча—Йожа [10], лежащий в основе решения задачи разделения случаев, когда некоторая булева функция от нескольких булевых переменных является сбалансированной (т. е. в половине случаев принимает значение 0, а в половине — 1) или константой. Это исторически первый алгоритм для квантовых вычислителей (1992 г.). Его выполнение требует одного фазового запроса на вычисление соответствующей функции. При этом используемая для его описания функция f дана как черный ящик, т. е. в ходе решения мы можем задавать оракулу только вопрос типа: "чему равна f на данном x".

Алгоритм Бернштейна—Вазерани [11], в котором как черный ящик используется функция $f(x) = (x, s)$ — скалярное произведение двоичных векторов. Необходимо найти s за минимальное число запросов. Этот алгоритм является модификацией алгоритма Дойча—Йожа. Даже если разрешить использование вероятностных алгоритмов (с заранее ограниченной вероятностью ошибки), решение классической задачи потребует $O(n) < \mathrm{const}n$ обращений к оракулу (здесь const — постоянная, не зависящая от n), в то время как в квантовом алгоритме достаточно $O(1)$ обращений к нему [12].

Алгоритм решения задачи о скрытой подгруппе (*Hidden Subgroup Problem* — HSP) для коммутативных групп представлен в работе [13]. Постановка задачи заключается в том, что дана функция, которая принимает постоянные значения на смежных классах по некоторой подгруппе. Необходимо получить образующие этой подгруппы.

Существование эффективных квантовых алгоритмов для HSP для определенных некоммутативных групп подразумевало бы эффективные квантовые алгоритмы для решения двух основных задач: определения существования изоморфизма графов и определения самых коротких векторов (SVP) в решетках. Более точно — эффективный квантовый алгоритм для HSP для симметричной группы дал бы квантовый алгоритм для изоморфизма графа. Эффективный квантовый алгоритм для HSP для группы диэдра дал бы квантовый полиномиальный алгоритм для SVP.

Алоритм Шора [14] представляет собой квантовый алгоритм факторизации, позволяющий разложить число n на простые множители за время $O(\log^3 n)$, используя $O(\log n)$ логических кубитов. Содержательно смысл алгоритма Шора состоит в реализации возведения в степень ($\mathrm{mod} n$) Эрмитовым преобразованием размерности $O(n^2)$ с использованием свойств тригонометрических сумм. При этом вероятность появления степени, равной порядку элемента, принципиально увеличивается. Другая математическая задача, направленная на дискретное логарифмирование в конечном простом поле из p элементов, часто применяющаяся для создания систем асимметричной криптографии, также является уязвимой для квантового алгоритма, предложенного Шором в работе [15]. Здесь применяется Эрмитово преобразование размерности $O(p^2)$ и дискретное преобразование Фурье.

Квантовый алгоритм Шора для факторинга и дискретного логарифмирования (а также некоторые из его расширений) полагается на способность квантовых компьютеров решить HSP для конечных абелевых групп.

Этот алгоритм был разработан Питером Шором в 1994 г. Семь лет спустя, в 2001 г., его работоспособность была продемонстрирована группой специалистов IBM. Число 15 было разложено на множители 3 и 5 с помощью квантового компьютера с семью кубитами.

Для дискретного логарифмирования на эллиптических кривых квантовый алгоритм не построен. Использование именно эллиптических кривых для реализации криптографических протоколов, стойкость которых держится на трудоемкости задачи дискретного логарифмирования, обоснована тем, что в этом случае для решения этой задачи невозможно применить алгоритмы с факторной базой, имеющие наилучшие на сегодня оценки сложности.

Алгоритм Гровера [16] бинарного поиска решений систем алгебраических уравнений имеет корневую оценку сложности. Это квантовый алгоритм решения задачи перебора, т. е. нахождения решения уравнения $f(x) = 1$, где f — булева функция от n переменных. Он был предложен американским математиком Л. Гровером в 1996 г. Предполагается, что функция f задана в виде черного ящика, или оракула.

Задача решения уравнения $f(x) = 1$ является общей формой задачи перебора; здесь требуется отыскать "пароль к устройству f ", что классически требует прямого перебора всех 2^n вариантов, где n — битовый размер входного слова. Алгоритм Гровера находит какой-нибудь корень уравнения, используя $O(2^{n/2})$ обращений к функции f, с использованием $O(n)$ кубитов.

Смысл алгоритма Гровера состоит в "усилении амплитуды" [17] целевого состояния за счет убывания амплитуды всех других состояний. Геометрически алгоритм Гровера заключается во вращении текущего вектора состояния квантового компьютера по направлению точно к целевому состоянию (движение по наикратчайшему пути обеспечивает оптимальность алгоритма). Это означает, что каждая итерация уточняет направление к искомому решению. Например, если симметричное шифрование может быть реализовано с помощью квантовых операций, то нахождение ключа по одной паре открытого и шифрованного текста может быть достигнуто за одно обращение к суперпозиции, выражающей значение ключа. Если симметричное шифрование может быть реализовано с помощью алгебраических операций, то нахождение ключа по одной паре открытого и шифрованного текста может быть достигнуто с помощью алгоритма Гровера с корневой оценкой сложности.

Квантовый компьютер может быть напрямую использован для дешифрования блочных шифров.

Пусть блочный шифр осуществляет взаимно однозначное отображение открытого текста t в шифрованный текст той же длины при ключе k той же длины $F(k, t) = c$. Пусть при разных k в один и тот же шифротекст преобразуются разные t. Пусть также нам удалось реализовать расшифрование с помощью унимодулярного преобразования: $F^{-1}(k, c) = t$. Тогда, сравнивая статистику результата со статистикой случайного осмысленного текста в языке, получаем открытый текст t.

Значение многокубитных квантовых компьютеров для криптографии

Угроза информационной безопасности, обусловленная созданием многокубитных квантовых компьютеров, в последнее время стала реальной. Начиная с 2017 г., были предложены первые квантовые процессоры с относительно большим числом кубитов. По поводу этого появилось много околонаучных публикаций. Например, публикации журналистов Castelvecchi D. Quantum computers ready to leap out of the lab in 2017 // Nature. 2017. Vol. 541. P. 9—10 и Pednault E., Gunnels J., Nannicini G. et al. (2019) Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits, https://arxiv.org/abs/1910.09534. Многие конференции по этому вопросу напрямую спонсируются разведкой или ВВС США (Quantum Computing: Progress and Prospects / Eds. E. Grumbling, M. Horowitz, Washington, DC: National Academies Press, 2018. DOI:10.17226/25196). В 2019 г. появилась научная статья [6], где описаны эксперименты с 53-кубитным процессором. Он работает в гильбертовом пространстве размерности 2^{53} с двоичными переменными. Однако избавиться от ошибок в работе этого процессора авторы не смогли.

Отметим, что для эффективного использования квантового компьютера при реализации указанных выше квантовых алгоритмов достаточно 500...1000 кубитов, по числу двоичных разрядов, используемых для записи параметров современных систем защиты информации. Например, если удастся записать преобразования секретного ключа в виде небольшой (полиномиальной длины) последовательности Эрмитовых унитарных преобразований, то при известном результате можно получить секретный ключ, поскольку все преобразования обратимы.

Криптографическая угроза от построения квантового компьютера была официально признана в 2017 г. Национальным институтом стандартов и технологий США, который объявил конкурс на создание постквантовых криптографических примитивов. Отметим, что в 2019 г. в рамках подкомитета 2 Технического комитета России была создана рабочая группа 2.5 "Постквантовые криптографические механизмы" (см. https://tc26.ru/about/structure/), в задачи которой входит разработка постквантовых криптографических механизмов.

Алгоритмы постквантовой криптографии

В силу того, что в 1997 г. П. Шором были предложены эффективные алгоритмы дискретного логарифмирования в простом поле и целой факторизации, следует обратить внимание на схемы, для которых квантовые алгоритмы пока остаются экспоненциальными. На настоящее время известны такие представленные далее схемы.

Шифрование на основе кодов. В 1978 г. Р. Мак-Элисом была предложена криптосистема с открытым ключом. Главная ее особенность состоит в том, что стойкость этой криптосистемы основана на сложности задачи исправления ошибок (декодирования), без знания проверочной матрицы. Этим эта схема отличается, например, от схем, основанных на сложности теоретико-числовых задач факторизации или дискретного логарифмирования. Фактически эта стойкость основана на сложности поиска наименьшего (ближайшего) вектора решетки, отмеченного ранее.

Стойкость криптосистемы Мак-Элиса, а также схем подписи, построенных на ней, основывается на сложности задачи синдромного декодирования кодов, исправляющих ошибки. На настоящее время лучший квантовый алгоритм решения этой задачи [18] имеет экспоненциальную сложность $\exp\{0{,}05869n\}$ от длины кода n. В 2001 г. группа французских исследователей на основе кодовой конструкции Нидеррайтера предложила постквантовую схему подписи CFS [19].

Слабым местом схем шифрования на основе кодов является большая длина открытого ключа и достаточно громоздкие вычисления при производстве подписи. Кроме того, линейный характер общей конструкции вызывает определенные опасения. Это, в частности, позволяет строить подписи, достаточно близкие в некотором смысле к легитимным. Данная схема в оригинальной версии Мак-Элиса для кодов Гоппы считается стойкой, она рассматривается Еврокомиссией как перспективная.

Кроме криптосистемы Мак-Элиса можно привести пример следующих четырех подходов.

Построения на основе хеш-функции. Подпись Меркла в качестве открытого ключа предлагает использовать корень дерева Меркла, а в качестве закрытого ключа — исходный массив записей. Сама подпись представляет собой набор смежных узлов к узлам аутентификационного пути дерева Меркла. В качестве слабости можно указать на необходимость иметь большую память для хранения секретных ключей. Основной недостаток схемы Меркла состоит в том, что для любого открытого ключа на основе хеш-функции существует ограничение на число подписей, которые могут быть получены из соответствующего одного набора закрытых ключей. Однако следует заметить, что это лучше, чем одноразовая подпись. Отмеченный недостаток схемы Меркла снижал уровень интереса к подписям такого типа, пока не появилась потребность в криптографии, устойчивой к воздействию квантовых компьютеров.

Построения на решетках. Классическим примером таких схем шифрования являются схемы Ring-Learning with Errors [18, 20, 21], NewHope (2015) или более старые NTRU (использование усеченных многочленов по двум взаимнопростым модулям), GGH и криптосистема Миччанчо. Стойкость алгоритма, лежащего в основе этих схем, обеспечивается трудно-

стью поиска кратчайшего вектора решетки (SVP). Эта задача на нынешний момент является более стойкой к атакам, осуществляемым на квантовых компьютерах. В отличие от своих конкурентов, а именно — RSA, ECC, Elgamal, рассматриваемый алгоритм использует относительно "дешевые" с точки зрения вычислений операции над кольцом $Z[X]/(X^N - 1)$ усеченных многочленов степени, не превосходящей $N - 1$. Однако с точки зрения стойкости относительно квантового вычислителя для задачи, которую он решает, достаточно перенести алгоритм решения задачи о скрытой подгруппе для коммутативных групп на случай группы диэдра. Речь идет о получении многочлена (делителя $X^N - 1$), по модулю которого проводятся арифметические операции со смежными классами. Сложность этого перехода пока недостаточно изучена.

Ключи для реализации перечисленных выше схем могут быть выбраны небольшими, а для преодоления изученных атак существуют следующие рекомендации — не выбирать слишком одиозные ключи, не передавать дважды одно и то же сообщение, использовать криптографически стойкую хеш-функцию и т. п. Шифрование и расшифрование — это схема, представляющая умножение и сложение многочленов с коэффициентами в простом поле. Поэтому она может быть реализована с использованием минимальных средств как по памяти, так и по вычислительной мощности (пластиковые карточки). Видимо поэтому подобный тип схем считается сейчас в США наиболее перспективным.

Построения с использованием многомерных квадратичных систем. Одной из самых интересных схем этого типа является подпись с открытым ключом Ж. Патарина HFE (первоначальная версия была предложена Имаи и Матсумото) [22]. Это схема с использованием алгебраической теории чисел, вскрытие которой сводится к решению большой квадратичной системы алгебраических уравнений. В этой схеме большой объем занимает открытый ключ. Для решения квадратичной системы может быть применен алгоритм с построением базиса Гребнера на обычном компьютере (в 2002 г. подобная атака реализована Шамиром и Фожером), а также алгоритм Гровера на квантовом компьютере.

Блочное шифрование. Примером блочного шифрования является шифр Rijndael [23], предложенный в 1998 г., впоследствии переименованный в AES (*Advanced Encryption Standard*). Этот вариант уже используется для шифрования в США. Предшественником AES является шифр DES [24], который опубликован в 1976 г. Таким образом, в США уже много лет назад выбрали шифр, стойкий к атакам квантового вычислителя. Аналогичную структуру (сеть Фейстеля) имеют базовые алгоритмы шифрования, представленные в ГОСТ 34.12—2018 "Информационная технология. Криптографическая защита информации. Блочные шифры", предшественником которого является ГОСТ 28147—89 "Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования" (введен в 1990 г.).

Шифрование с использованием суперсингулярной изогении (SIDH). Такое шифрование [25] представляет собой аналог протокола Диффи—Хеллмана. Оно основано на блуждании в суперсингулярном изогенном графе. Суперсингулярные кривые, в отличие от обычных, имеют кольцо эндоморфизмов ранга 4. Здесь вместо возведения в степень используется рациональное отображение, гомоморфизм, переводящий исходную кривую или кривую, преобразованную вторым абонентом в изогенную. Ввиду коммутативности композиции изогений получается одинаковая кривая (секретным ключом может быть и ее *j*-инвариант). Вскрытие этой схемы с помощью алгоритмов дискретного логарифмирования осложнено невозможностью использования техники факторных баз. Из всех постквантовых протоколов обмена ключами SIDH имеет наименьшую длину ключа; с учетом сжатия SIDH использует 2688-битный [26] публичный ключ на 128-битном квантовом криптографическом уровне. Следует также отметить, что SIDH отличается от других похожих систем, таких как NTRU и Ring-LWE, поддержкой совершенной прямой секретности. Она гарантирует, что сессионные ключи, полученные с помощью набора ключей долговременного пользования, не будут скомпрометированы при компрометации одного из долговременных ключей. Эти свойства SIDH делают его одним из кандидатов на замену протокола Диффи—Хеллмана в конечных полях (DHE) и протокола Диффи—Хеллмана на эллиптических кривых (ECDHE), которые используются сейчас при защите данных, передаваемых через сеть.

Суперсингулярные кривые над полем из q элементов — это только кривые порядка $q + 1$. Поэтому существует опасность попасть на уже изученную плохую орбиту кривых. Кроме того, задача вычисления изогений достаточно трудоемка.

В качестве постквантового механизма открытого распределения ключей можно предложить хеширование абонентами всей общей секретной информации, накопившейся к данному моменту. Для сокращения памяти можно использовать дерево Меркла.

Результаты конкурсов

30 января 2019 г. NIST (*National Institute of Standards and Technology*) обнародовал результаты второго этапа стандартизации протоколов постквантовой криптографии, проводимой в США, в которых как наиболее перспективный был выделен алгоритм на решетках (см. третью сторону обложки).

Ниже представлены кандидаты, прошедшие в третий этап.

- CRYSTALS-DILITHUM — является представителем криптографии на решетках. За основу взята схема Фиата—Шамира с прерываниями. Криптоанализ сводится к решению задач Module-LWE, Module-SIS. Имеет хорошую производительность и может быть эффективно реализован на малоресурсных устройствах. NIST попросил авторов добавить набор общесистемных параметров для пятого уровня безопасности.

• FALCON — также является представителем криптографии на решетках. Но за основу взят фреймворк GPV. Криптоанализ сводится к задаче SIS на NTRU-решетках. Главным недостатком этой схемы является сложная программная и аппаратная реализация. Схема использует вычисления над числами с плавающей запятой, что как сильно усложняет анализ стойкости к атакам по сторонним каналам, так и делает сложным реализацию для малоресурсных устройств.

• RAINBOW — является представителем криптографии на мультивариативных преобразованиях. За основу взята схема UOV. Главным преимуществом является размер цифровой подписи. Но в силу большого размера ключа эту схему рекомендуется использовать только для специфических задач, где размер ключей не критичен.

NIST также заявил, что хотя бы одна из схем CRYSTALS-DILITHUM, FALCON будет стандартизована. Таким образом, для цифровой подписи в будущем скорее всего будут использоваться схемы на основе криптографии на решетках, а для более специфических задач — RAINBOW.

Для асимметричного шифрования в третий этап вышли перечисленные далее схемы.

CLASSIC McElice — является представителем криптографии на кодах, исправляющих ошибки. Основная конструкция схемы была предложена еще в 1979 г. и хорошо изучена. Имеет малые размеры шифротекстов, но очень большой размер ключа, вследствие чего имеет те же проблемы, что и RAINBOW и рекомендуется к использованию только в специальных задачах.

CRYSTALS-KYBER — является представителем криптографии на решетках. Криптоанализ сводится к решению задачи Module-LWE. Для обеспечения стойкости к атакам с адаптивно подобранными шифротекстами используется преобразование Фуджисаки—Окамото. Имеет хорошую производительность и безопасность, но NIST также напоминает, что Module-LWE — это относительно малоизученная проблема, которая требует более детального криптоанализа.

• NTRU — является представителем криптографии на решетках. За основу взята схема NTRUEncrypt, предложенная более 20 лет назад. Проблема NTRU в отличие от Module-LWE (и других модификаций) была очень хорошо изучена, что является очень важным фактором.

• SABER — является представителем криптографии на решетках. Криптоанализ сводится к проблеме Module-LWE, где вместо сложения с вектором ошибки используется округление по меньшему модулю. Используется преобразование Фуджисаки—Окамото, как и в CRYSTALS-KYBER.

В целом, ситуация аналогичная — для общего использования рекомендуются схемы на основе решеток. Но NIST сделал замечание, что только одна из схем на решетках (CRYSTALS-KYBER, NTRU, SABER) будет стандартизована.

Заключение

Проблемы, с которыми столкнулись технологи при построении квантового компьютера, имеют качественный, а не количественный характер. Поэтому в самое ближайшее время он, вероятно, не может быть построен. Вместе с тем открытые в последнее время квантово-механические эффекты в физике и теория тригонометрических сумм могут быть положены в основу новой теории квантовых вычислений вне зависимости от вида и времени ее реализации на практике. Эта теория может быть использована как для моделирования природных квантово-механических процессов, так и для решения с помощью таких процессов вычислительных задач, решение которых на обычном компьютере практически невозможно.

Список литературы

1. **Манин Ю. И.** Вычислимое и невычислимое. М.: Советское радио, 1980. — 130 с.
2. **Feynman R. P.** Simulating Physics with Computers // International Journal of Theoretical Physics. — 1982. — Vol. 21, No. 6—7. — P. 467—488. DOI: 10.1007/BF02650179.
3. **Steane A. M., Rieffel E. G.** Beyond Bits: The Future of Quantum // Information Processing. Computer. — 2000. — Vol. 33, No. 1. — P. 38—45. DOI: 10.1109/2.816267.
4. **Rieffel E., Wolfgang P.** An Introduction to Quantum Computing for Non-Physicists // ACM Computing Surveys. — 2000. — Vol. 32, No. 3. — P. 300—335. DOI: 10.1145/367701.367709.
5. **Deutsch D.** Quantum theory, the Church-Turing principle and the universal quantum computer // Proceedings of the Royal Society of London; Series A, Mathematical and Physical Sciences. — 1985. —Vol. 400, No. 1818. — P. 97—117. DOI:10.1098/rspa.1985.0070.
6. **Arute F., Arya K., Babbush R.** et al. Quantum supremacy using a programmable superconducting processor// Nature. — 2019. — Vol.574. — P. 505—510. URL: https://www.nature.com/articles/s41586-019-1666-5
7. **Ras R., Tal A.** Oracle Separation of BQP and PH. Electronic Colloquium on Computational Complexity, Report No. 107. 2018. — 22 p.
8. **Simon D. R.** On the Power of Quantum Computation // 35th Annual Symposium on Foundations of Computer Science — FOCS 1994, IEEE Computer Society, 1994. — P. 116—123.
9. **Chi-Chih Yao A.** Quantum circuit complexity // Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science, Palo Alto, CA, USA, 1993. — P. 352—361. DOI: 10.1109/SFCS.1993.366852.
10. **Deutsch D., Jozsa R.** Rapid solution of problems by quantum computation // Proceedings of the royal society A math., phyz., eng. sci. — 1992. — Vol. 439, No. 1907. — P. 553—558. DOI: 10.1098/rspa.1992.0167.
11. **Bernstein E., Vazirani U.** Quantum Complexity Theory // Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing. — NY, USA: ACM, 1993. — P. 11—20. DOI: 10.1145/167088.167097.
12. **Hidary J. D.** Quantum Computing: An Applied Approach. Springer International Publishing, 2019. — P. 104—107. DOI:10.1007/978-3-030-23922-0.
13. **Koiran P., Nesme V., Portier N.** The quantum query complexity of the abelian hidden subgroup problem // Theoretical Computer Science. — 2007. — Vol. 380, Iss. 1—2. — P. 115—126. DOI: 10.1016/j.tcs.2007.02.057.
14. **Shor P. W.** Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer // SIAM J. Comput. — 1997. — Vol. 26, No. 5. — P. 1484—1509.
15. **Shor P.** Algorithms for Quantum Computation: Discrete Logarithms and Factoring // Foundations of Computer Science, 1994 Proceedings, 35th Annual Symposium on IEEE, 1994. — P. 124—134. DOI: 10.1109/SFCS.1994.365700.
16. **Grover L. K.** A fast quantum mechanical algorithm for database search // Proceedings 28th Annual ACM Symposium on the Theory of Computing, — May 1996, — P. 212—219.

17. **Brassard G., Hoyer P.** An Exact Quantum Polynomial-Time Algorithm for Simon's Problem // Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems (ISTCS'97). IEEE Computer Society Press, 1997. — P. 12—23. DOI: 10.1109/ISTCS.1997.595153.

18. **Kachigar G., Tillich J.-P.** Quantum information set decoding algorithms/ Eds T. Lange, T. Takagi // PQCrypto 2017. Springer, 2017. — Vol. 10346 of LNCS. — P. 69—89.

19. **Courtois N., Finiasz M., Sendrier N.** How to Achieve a McEliece-Based Digital Signature Scheme // Advances in Cryptology — ASIACRYPT 2001. — 2001. — Vol. 2248 of LNCS. — P. 157—174.

20. **Guneysu T., Lyubashevsky V., Pöppelmann T.** Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems // CHES 2012: Cryptographic Hardware and Embedded Systems — CHES 2012. NCS. Springer Berlin Heidelberg, 2012. — P. 530—547.

21. **Jiang Zhang, Zhenfeng Zhang, Jintai Ding** et al. Authenticated Key Exchange from Ideal Lattices // IACR.ORG. IACR. 2014. URL: https://eprint.iacr.org/2014/589.pdf

22. **Imai H., Matsumoto T.** Public Quadratic polynomial-tuples for efficient signature-verification and message-encription // Advances in Cryptography — Eurocrypt'88, Springer-Verlag, 1988. — P. 419—453.

23. **Advanced** Encryption Standard (AES). Federal Information. Processing Standards Publication 197, November 26, 2001.

24. **Национальное** бюро стандартов, стандарт шифрования данных, FIPS-Pub.46. Национальное бюро стандартов, Министерство торговли США, Вашингтон, округ Колумбия, январь 1977 г.

25. **Де Фео Л.** Математика криптографии на основе изогении. 2017arXiv:1711.04062 [cs.CR].

26. **Costello C., Jao D., Longa P.** et al. Efficient compression of SIDH public keys. 2016. — 4 October, ePrint 2016/963.

Quantum Computer and Post-Quantum Cryptography

S. A. Bukashkin, sergey.bukashkin@gmail.com, JSC "Concern "Avtomatika", Moscow, 127276, Russian Federation, **M. A. Cherepniov,** cherepniov@gmail.com, MSU, Moscow, 115432, Russian Federation

Corresponding author:

Cherepniov Mikhail A., PhD, Professor, MSU, Moscow, 115432, Russian Federation
E-mail: cherepniov@gmail.com

Received on February 28, 2021
Accepted on April 19, 2021

An overview of the current state of the problem of building a quantum computer and its hypothetical use for breaking cryptographic protocols is presented. The necessary parameters are considered. An overview of existing quantum algorithms and post-quantum cryptographic protocols that are strong with respect to them is presented. The problem of constructing a quantum computer is considered in comparison with the development of the theory and practice of conventional mechanical and electronic computers. The results of contests on the topic of post-quantum cryptography are presented.

Keywords: *quantum computer, quantum algorithms, post-quantum cryptography*

Acknowledgements:

This work of M. A. Cherepniov was supported by the Russian Foundation for Basic Research, project no. 18-29-03124\20.

For citation:

Bukashkin S. A., **Cherepniov M. A.** Quantum Computer and Post-Quantum Cryptography, *Programmnaya Ingeneria,* 2021, vol. 12, no. 4, pp. 171—178.

DOI: 10.17587/prin.12.171-178

References

1. **Manin Y. I.** *Computable and non-computable*, Moscow, Sovetskoe radio, 1980, 130 p. (in Russian).

2. **Feynman R. P.** Simulating Physics with Computers, *International Journal of Theoretical Physics,* 1982, vol. 21, no. 6—7, pp. 467—488. DOI: 10.1007/BF02650179.

3. **Steane A. M., Rieffel E. G.** Beyond Bits: The Future of Quantum, *Information Processing. Computer,* 2000, vol. 33, no. 1, pp. 38—45. DOI: 10.1109/2.816267.

4. **Rieffel E., Wolfgang P.** An Introduction to Quantum Computing for Non-Physicists, *ACM Computing Surveys*, 2000, vol. 32, no. 3, pp. 300—335. DOI: 10.1145/367701.367709.

5. **Deutsch D.** Quantum theory, the Church-Turing principle and the universal quantum computer, *Proceedings of the Royal Society of London; Series A, Mathematical and Physical Sciences,* 1985, vol. 400, no. 1818, pp. 97—117. DOI:10.1098/rspa.1985.0070.

6. **Arute F., Arya K., Babbush R.** et al. Quantum supremacy using a programmable superconducting processor, *Nature,* 2019, vol. 574, pp. 505—510, available at: https://www.nature.com/articles/s41586-019-1666-5

7. **Ras R., Tal A.** Oracle Separation of BQP and PH. Electronic Colloquium on Computational Complexity, Report No. 107, 2018, 22 p.

8. **Simon D. R.** On the Power of Quantum Computation, *35th Annual Symposium on Foundations of Computer Science — FOCS 1994*, IEEE Computer Society, 1994, pp. 116—123.

9. **Chi-Chih Yao A.** Quantum circuit complexity, *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science,* Palo Alto, CA, USA, 1993, pp. 352-361. DOI: 10.1109/SFCS.1993.366852.

10. **Deutsch D., Jozsa R.** Rapid solution of problems by quantum computation, *Proceedings of the royal society A math., phyz., eng. sci.,* 1992, vol. 439, no. 1907, pp. 553-558. DOI: 10.1098/rspa.1992.0167.

11. **Bernstein E., Vazirani U.** Quantum Complexity Theory, *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing.* NY, USA, ACM, 1993, pp. 11—20. DOI: 10.1145/167088.167097.

12. **Hidary J. D.** *Quantum Computing: An Applied Approach*, Springer International Publishing, 2019, pp. 104—107. DOI:10.1007/978-3-030-23922-0.

13. **Koiran P., Nesme V., Portier N.** The quantum query complexity of the abelian hidden subgroup problem, *Theoretical Computer Science.* 2007. Vol. 380, Iss. 1, 2. pp. 115—126. DOI: 10.1016/j.tcs.2007.02.057.

14. **Shor P. W.** Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM J. Comput.*, 1997, vol. 26, no. 5, pp. 1484—1509.

15. **Shor P.** Algorithms for Quantum Computation: Discrete Logarithms and Factoring, *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on IEEE*, 1994, pp. 124—134. DOI: 10.1109/SFCS.1994.365700.

16. **Grover L. K.** A fast quantum mechanical algorithm for database search, *Proceedings 28th Annual ACM Symposium on the Theory of Computing*, May 1996, pp. 212—219.

17. **Brassard G. Hoyer P.** An Exact Quantum Polynomial-Time Algorithm for Simon's Problem, *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems (ISTCS'97)*, IEEE Computer Society Press, 1997, pp. 12—23. DOI: 10.1109/ISTCS.1997.595153.

18. **Kachigar G., Tillich J.-P.** Quantum information set decoding algorithms / Eds T. Lange, T. Takagi, *PQCrypto 2017, Lecture Notes in Computer Science,* Springer, 2017, vol. 10346, pp. 69—89.

19. **Courtois N., Finiasz M., Sendrier N.** How to Achieve a McEliece-Based Digital Signature Scheme, *Advances in Cryptology — ASIACRYPT 2001, Lecture Notes in Computer Science,* 2001, vol. 2248, pp. 157—174.

20. **Guneysu T., Lyubashevsky V., Pöppelmann T.** Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems, *CHES 2012: Cryptographic Hardware and Embedded Systems — CHES 2012*, Springer Berlin Heidelberg, 2012, pp. 530—547.

21. **Jiang Zhang, Zhenfeng Zhang, Jintai Ding** et al. Authenticated Key Exchange from Ideal Lattices, *IACR.ORG. IACR*, 2014, available at: https://eprint.iacr.org/2014/589.pdf

22. **Imai H., Matsumoto T.** Public Quadratic polynomial-tuples for efficient signature-verification and message-encription, *Advances in Cryptography — Eurocrypt'88,* Springer-Verlag, 1988, pp. 419—453.

23. **Advanced** Encryption Standard (AES). Federal Information. Processing Standards Publication 197, November 26, 2001.

24. **National Bureau of Standards,** Data Encryption Standard, FIPS-Pub.46. National Bureau of Standards, U. S. Department of Commerce, Washington, D. C., January 1977.

25. **De Feo L.** Mathematics of cryptography based on isogeny. 2017arXiv:1711.04062 [cs.CR].

26. **Costello C., Jao D., Longa P.** et al. *Efficient compression of SIDH public keys*, 2016, 4 October, ePrint 2016/963.

ИНФОРМАЦИЯ

УДК 004.056.55 DOI: 10.17587/prin.12.179-188

Ф. К. Алиев, д-р физ.-мат. наук, консультант отдела, ДИС МО РФ, Москва,
А. В. Корольков, канд. техн. наук, доц., зав. каф., ФГБУ ВО "Российский технологический университет — МИРЭА", начальник НИЦ, АК РФ, Москва,
Е. А. Матвеев, канд. физ.-мат. наук, директор, НТП "Криптософт", Пенза,
И. А. Шеремет, член-корр. РАН, д-р техн. наук, проф., зам. директора, РФФИ, Москва

О чувствительности гаммы квантовой криптографической системы АКМ2017 к изменениям сеансового ключа

Рассмотрена квантовая криптографическая система АКМ2017. Представлены результаты анализа зависимости степени различия гамм зашифрования и расшифрования от степени различия соответствующих сеансовых ключей. Выявлено и обосновано равенство указанных степеней различия. Для произвольно зафиксированного сеансового ключа зашифрования выявлено и описано распределение сеансовых ключей расшифрования по классам в зависимости от значения степени различия гамм зашифрования и расшифрования. Один класс составляют все сеансовые ключи расшифрования, приводящие к одному и тому же значению степени различия гамм зашифрования и расшифрования. Приведена геометрическая интерпретация указанного распределения по классам в виде размещения по окружностям (класс—окружность) на поверхности сферы единичного радиуса с центром в начале евклидовой прямоугольной системы координат в трехмерном линейном пространстве над полем действительных чисел. Изложенные результаты могут быть использованы при решении задач оптимизации значений параметров точности и надежности функционирования вариантов практических реализаций квантовой криптографической системы АКМ2017, например, при настройке сеансового ключа расшифрования, позволяющей гарантировать наперед заданное малое значение математического ожидания числа неправильно расшифрованных двоичных символов открытого текста.

Ключевые слова: *квантовая криптография, криптографическая система, квантовый компьютер, кубит, квантовая система, несепарабельные (запутанные) состояния, теоретическая стойкость, криптографическая техника, состояние "спиновый синглет", измерение компоненты спина вдоль оси*

Введение

Криптографические методы защиты информации основаны на достижениях в области математики, информатики, физики, науки о данных. В настоящее время развитие криптографии характеризуется широким использованием квантовых эффектов для автоматической абонентской (с возможностью децентрализованной) генерации и распределения ключей для симметричных криптографических систем. Возникла, развивается и достигла стадии практических применений так называемая квантовая криптография [7, 11, 13, 14, 22]. Квантовая и постквантовая криптографии [7] объединяют в себе криптографические средства, методы и способы защиты информации, сохраняющие свой защитный потенциал в условиях возможного создания и применения квантовых компьютеров [13, 19] для решения задач дешифрования (взлома).

Классическая криптография предоставляет широкие возможности для защиты информации [3, 4, 10, 11, 16—18, 21, 22], в том числе и от атак с применением квантовых компьютеров. Например, путем применения для защиты информации криптографической техники, имеющей соответствующее заключение регулятора, в которой реализованы теоретически стойкие криптографические алгоритмы [3, 4, 7, 10, 11]. Однако в этом случае возникают вопросы, связанные с высокой ресурсозатратностью и, соответственно, недостаточной эффективностью для массового практического применения такой техники. Более конкретно, теоретически стойкие системы [3, 4, 10], по Шеннону, совершенные шифры [21] на основе классической криптографии имеют ряд недостатков, существенно затрудняющих их практическое применение в области обеспечения информационной безопасности. Самым значимым среди них является сложность подсистемы управления ключами, под которой понимается подсистема генерации, распределения, применения и утилизации ключевой информации. Как правило, по причине сложности

подсистемы управления ключами теоретически стойкие системы классической криптографии громоздки, дóроги по затратам на выработку и распределение ключевой информации, подвержены повышенной опасности компрометации ключевой информации вне контролируемых зон. По этой причине они не могут иметь массового применения при решении задач информационной безопасности [7, 11]. Квантовая криптография позволяет радикально упростить подсистему управления ключами и полностью исключить влияние "человеческого фактора" на жизненный цикл криптографических ключей.

Необходимо отметить еще одно важное различие классической и квантовой криптографии. Классическая криптография базируется на математических алгоритмах, в которых нет места ошибкам. Квантовые криптографические системы разрабатываются на основе эффектов квантовой физики [7, 11, 13, 14, 22]. В криптографической технике, реализующей протоколы квантовой криптографии, существенно возрастает роль физики, физических процессов, чего не было в классической криптографической технике. Настоящая статья посвящена одному из возникающих в связи с этим вопросов, а именно вопросу об ошибках в расшифрованном тексте, связанных с возможным несовпадением ключа (гаммы) зашифрования и ключа расшифрования.

Для классической криптографии в рассмотрении этого вопроса не было необходимости, так как при условии точного выполнения криптографического протокола, исправности техники и т. п., появление указанного типа ошибок исключено. Данное обстоятельство обусловлено тем фактом, что ключи для зашифрования и расшифрования (напомним, что речь идет о симметричных криптографических системах) являются копиями одной и той же выходной последовательности программного или физического генератора случайных символов, по сути, копиями реализации одного и того же физического процесса.

В случае классической криптографии ключи зашифрования и расшифрования являются копиями результата, относящегося к одному эксперименту, одному комплексу экспериментальных условий. В случае квантовых криптографических систем ситуация совершенно иная. Ключи зашифрования и расшифрования являются результатами, относящимися к двум различным экспериментам, к двум различным комплексам экспериментальных условий, т. е. к двум различным физическим процессам.

Обращая внимание, в частности, на квантовые криптографические системы, основанные на использовании несепарабельности (запутанности, сцепленности) [1, 7, 13] квантовых систем, можем заметить, что эти комплексы экспериментальных условий различны, так как для генерации ключей зашифрования и расшифрования измеряют разные подсистемы двухсоставных квантовых систем. Эти подсистемы разнесены в пространстве, а в некоторых квантовых криптографических системах указанные измерения разнесены и во времени, например, в АКМ2017 [2, 11]. Необходимые измерения проводят с использованием физически различной (разные как объекты) аппаратуры. Перечисленным выше не исчерпывается весь спектр причин, обусловливающих различия физических процессов для генерации ключей зашифрования и расшифрования, в результате чего эти ключи могут не совпадать. Таким образом, естественно встает вопрос, обсуждаемый в настоящей статье, о степени различия результатов реализации указанных физических процессов, определяющих ключи зашифрования и расшифрования.

1. Состояние "спиновый синглет" и измерение компоненты спина вдоль оси

В этом разделе представлены некоторые свойства состояния "спиновый синглет" квантовой системы из двух кубитов в связи с измерением компоненты спина вдоль оси. Материал раздела используется далее, как в описании протокола АКМ2017, так и в обосновании основных результатов данной статьи.

Рассмотрим состояние Белла [13]

$$|\psi_{11}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

двухкубитной квантовой системы, состоящей из двух кубитов *A* и *B*. По историческим причинам это состояние принято называть **спиновым синглетом** [13]. Как следует из работы [1], состояние $|\psi_{11}\rangle$ является несепарабельным состоянием квантовой системы из двух кубитов.

Напомним [1, 13], под *измерением компоненты спина вдоль оси* **v**, где $\mathbf{v} = (v_1, v_2, v_3)$ — единичный вектор в трехмерном пространстве над полем действительных чисел **R**, понимается измерение наблюдаемой

$$\mathbf{v}\bullet\boldsymbol{\sigma} = v_1\sigma_1 + v_2\sigma_2 + v_3\sigma_3,$$

где σ_1, σ_2, σ_3 — вентили Паули;

$$\sigma_1 = \sigma_{\mathbf{X}} = \mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

и представляет собой квантовый аналог классического логического элемента NOT; действует на однокубитное состояние $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, где $\alpha, \beta \in \mathbf{C}$, $|\alpha|^2 + |\beta|^2 = 1$ следующим образом

$$\sigma_1|\psi\rangle = \sigma_{\mathbf{X}}|\psi\rangle = \mathbf{X}|\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}|\psi\rangle = \beta|0\rangle + \alpha|1\rangle;$$

$$\sigma_2 = \sigma_{\mathbf{Y}} = \mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

и действует на однокубитное состояние $|\psi\rangle$ следующим образом:

$$\sigma_2|\psi\rangle = \sigma_{\mathbf{Y}}|\psi\rangle = \mathbf{Y}|\psi\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}|\psi\rangle = -\beta i|0\rangle + \alpha i|1\rangle,$$

где $i \in \mathbf{C}$, i — мнимая единица, т. е. $i^2 = -1$;

$$\sigma_3 = \sigma_{\mathbf{Z}} = \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

и действует на однокубитное состояние $|\psi\rangle$ следующим образом:

$$\sigma_3|\psi\rangle = \sigma_{\mathbf{Z}}|\psi\rangle = \mathbf{Z}|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}|\psi\rangle = \alpha|0\rangle - \beta|1\rangle.$$

Для наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$ имеют место равенства:

$$\mathbf{v}\cdot\boldsymbol{\sigma} = v_1\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + v_2\begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} + v_3\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \\ = \begin{pmatrix} v_3 & v_1 - iv_2 \\ v_1 + iv_2 & -v_3 \end{pmatrix}. \quad (1)$$

Вычислим характеристический многочлен $\chi_{\mathbf{v}\cdot\boldsymbol{\sigma}}(\lambda)$ полученной матрицы (учитывая, что $v_1^2 + v_2^2 + v_3^2 = 1$):

$$\chi_{\mathbf{v}\cdot\boldsymbol{\sigma}}(\lambda) = \begin{vmatrix} \lambda - v_3 & -v_1 + iv_2 \\ -v_1 - iv_2 & \lambda + v_3 \end{vmatrix} = \lambda^2 - 1.$$

Отсюда следует, что возможные значения наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$ равны $\lambda_{1,2} = \pm 1$ независимо от значений координат единичного вектора $\mathbf{v} = (v_1, v_2, v_3)$. Таким образом, при выполнении измерения компоненты спина вдоль оси $\mathbf{v} = (v_1, v_2, v_3)$ для обоих кубитов A и B, т. е. измерения наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$ для каждого из кубитов A и B, получим для каждого из них "1" или "−1". Других значений быть не может, так как выше было показано, что возможные значения наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$ равны ±1 независимо от значений координат единичного вектора $\mathbf{v}$.

Более того, имеет место важное утверждение, которое играет существенную роль в исследовании свойств квантовой криптографической системы AKM2017. Перед формулировкой и доказательством этого утверждения проведем необходимые для дальнейшего вычисления.

Выразим через координаты вектора $\mathbf{v} = (v_1, v_2, v_3)$ координаты собственных состояний $|a\rangle$ и $|b\rangle$ наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$, отвечающих собственным значениям "1" и "−1" соответственно. И, кроме того, выразим векторы $|0\rangle$ и $|1\rangle$ через векторы $|a\rangle$ и $|b\rangle$.

Отдельно будем рассматривать три случая в зависимости от значения координаты v_3 вектора $\mathbf{v}$: $v_3 = 1$, $v_3 = -1$, $v_3 \notin \{\pm 1\}$.

Пусть $v_3 = 1$. Тогда из равенства $v_1^2 + v_2^2 + v_3^2 = 1$ следует, что $v_1 = v_2 = 0$. Отсюда и из равенств (1) следует, что

$$\mathbf{v}\cdot\boldsymbol{\sigma} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

что, в свою очередь, влечет справедливость равенств

$$|a\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \ |b\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (2)$$

Из равенств (2) следует, что

$$|0\rangle = |a\rangle, \ |1\rangle = |b\rangle. \quad (3)$$

Пусть $v_3 = -1$. Тогда из равенства $v_1^2 + v_2^2 + v_3^2 = 1$ следует, что $v_1 = v_2 = 0$. Отсюда и из равенств (1) следует, что

$$\mathbf{v}\cdot\boldsymbol{\sigma} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix},$$

что, в свою очередь, влечет справедливость равенств

$$|a\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \ |b\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

из которых следует, что

$$|0\rangle = |b\rangle, \ |1\rangle = |a\rangle. \quad (4)$$

Пусть $v_3 \notin \{\pm 1\}$.Тогда, с учетом равенств (1), из равенств $\mathbf{v}\cdot\boldsymbol{\sigma}|a\rangle = |a\rangle$, $\mathbf{v}\cdot\boldsymbol{\sigma}|b\rangle = -|b\rangle$ получаем:

$$|a\rangle = \begin{pmatrix} \dfrac{v_1 - iv_2}{\sqrt{(1 - v_3)^2 + (v_1^2 + v_2^2)}} \\ \dfrac{1 - v_3}{\sqrt{(1 - v_3)^2 + (v_1^2 + v_2^2)}} \end{pmatrix}, \quad (5)$$

$$|b\rangle = \begin{pmatrix} \dfrac{-v_1 + iv_2}{\sqrt{(1 + v_3)^2 + (v_1^2 + v_2^2)}} \\ \dfrac{1 + v_3}{\sqrt{(1 + v_3)^2 + (v_1^2 + v_2^2)}} \end{pmatrix}. \quad (6)$$

Из (5) и (6) следует, что

$$|0\rangle = \frac{(v_1 + iv_2)(1 + v_3)k_a}{2(v_1^2 + v_2^2)}|a\rangle + \\ + \frac{(v_1 + iv_2)(-1 + v_3)k_b}{2(v_1^2 + v_2^2)}|b\rangle, \ |1\rangle = \frac{k_a}{2}|a\rangle + \frac{k_b}{2}|b\rangle, \quad (7)$$

где

$$k_a = \sqrt{(1 - v_3)^2 + (v_1^2 + v_2^2)} = \sqrt{2 - 2v_3}, \\ k_b = \sqrt{(1 + v_3)^2 + (v_1^2 + v_2^2)} = \sqrt{2 + 2v_3}. \quad (8)$$

Имеет место следующее утверждение.

Утверждение 1. Пусть квантовая система AB из двух кубитов A и B находится в состоянии $|\psi_{11}\rangle$, т. е. в состоянии "спиновый синглет". Тогда для любого единичного вектора $\mathbf{v} = (v_1, v_2, v_3)$ над полем действительных чисел $\mathbf{R}$ измерения наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$ для каждого из кубитов A и B (вне зависимости от того, какой из них подвергается измерению первым, а какой – вторым) дают значение результата первого измерения, равное "1" или "−1" с вероятностью 0,5;

а значение результата второго измерения с вероятностью 1 равно значению результата первого измерения с противоположным знаком.

Доказательство. Пусть $v_3 = 1$. В этом случае искомый результат очевидным образом следует из равенств (3).

Пусть $v_3 = (-1)$. В этом случае искомый результат очевидным образом следует из равенств (4).

Пусть $v_3 \notin \{\pm 1\}$. Тогда из равенств (7) следует, что

$$\frac{|01\rangle - |10\rangle}{\sqrt{2}} = k_{ab}\frac{|ab\rangle - |ba\rangle}{\sqrt{2}}, \qquad (9)$$

где

$$k_{ab} = \frac{(v_1 + iv_2)(1+v_3)k_a}{2\left(v_1^2+v_2^2\right)}\frac{k_b}{2} - \frac{(v_1+iv_2)(-1+v_3)k_b}{2\left(v_1^2+v_2^2\right)}\frac{k_a}{2} =$$

$$= (v_1 + iv_2)\frac{k_a k_b}{2\left(v_1^2+v_2^2\right)}.$$

Вычислим модуль $|k_{ab}|$ величины k_{ab}. Из определения модуля комплексного числа и равенств (8) следует, что

$$|k_{ab}| = \sqrt{v_1^2+v_2^2}\,\frac{k_a k_b}{2\left(v_1^2+v_2^2\right)} = \sqrt{v_1^2+v_2^2}\times$$

$$\times\frac{\sqrt{\left((1-v_3)^2+\left(v_1^2+v_2^2\right)\right)\left((1+v_3)^2+\left(v_1^2+v_2^2\right)\right)}}{2\left(v_1^2+v_2^2\right)} = \qquad (10)$$

$$= \sqrt{v_1^2+v_2^2}\,\frac{\sqrt{4\left(v_1^2+v_2^2\right)}}{2\left(v_1^2+v_2^2\right)} = 1.$$

Из равенств (9) и (10) следует, что состояния $\frac{|01\rangle - |10\rangle}{\sqrt{2}}$ и $\frac{|ab\rangle - |ba\rangle}{\sqrt{2}}$ совпадают с точностью до ненаблюдаемого при измерении общего множителя k_{ab}. Таким образом, если выполнено измерение наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$ для каждого из кубитов A и B (вне зависимости какой из кубитов A и B подвергается измерению первым, а какой — вторым), то результат "1" (или "−1"), полученный при первом измерении, приводит к результату "−1" (или "1") при втором измерении.

Действительно, допустим, что первым измерению подвергался кубит A и получен результат "1". Тогда после этого измерения квантовая система AB из двух кубитов A и B окажется в состоянии $|ab\rangle$, что предопределяет значение "−1" результата второго измерения уже над кубитом B наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$, у которой состояние $|b\rangle$ отвечает собственному значению "−1".

Аналогично, если первым измерению подвергался кубит A и получен результат "−1", то после этого измерения квантовая система AB из двух кубитов A и B окажется в состоянии $|ba\rangle$, что предопределяет значение "1" результата второго измерения уже над кубитом B наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$, у которой состояние $|a\rangle$ отвечает собственному значению "1".

Аналогично, если первым измерению подвергался кубит B и получен результат "1", то после этого измерения квантовая система AB из двух кубитов A и B окажется в состоянии $|ba\rangle$, что предопределяет значение "−1" результата второго измерения уже над кубитом A наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$, у которой состояние $|b\rangle$ отвечает собственному значению "−1".

Аналогично, если первым измерению подвергался кубит B и получен результат "−1", то после этого измерения квантовая система AB из двух кубитов A и B окажется в состоянии $|ab\rangle$, что предопределяет значение "1" результата второго измерения уже над кубитом A наблюдаемой $\mathbf{v}\cdot\boldsymbol{\sigma}$, у которой состояние $|a\rangle$ отвечает собственному значению "1".

Утверждение 1 доказано.

Замечание 1. Утверждение 1 служит математической основой квантовой криптографической системы АКМ2017 [2, 11], краткое описание которой представлено в разд. 2. В АКМ2017 вектор, вдоль которого производится измерение компоненты спина, служит тем, что в криптографии принято называть *синхропосылкой* [11], а результат измерения используется при выработке бита ключа. В данной работе (как и в работе [2]) этот вектор называют *сеансовым* (или *разовым*) ключом квантовой криптографической системы АКМ2017.

Далее в этом разделе пусть

$$\mathbf{v} = (v_1,\ v_2,\ v_3),\ \mathbf{w} = (w_1,\ w_2,\ w_3) \in \mathbf{R}^3,$$

где $v_1^2 + v_2^2 + v_3^2 = 1$, $w_1^2 + w_2^2 + w_3^2 = 1$. Из того, что квадрат матрицы $(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})$ равен единичной матрице, следует, что собственные числа и, соответственно, возможные значения наблюдаемой $(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})$ равны ± 1 независимо от значений координат векторов $\mathbf{v}$ и $\mathbf{w}$. Наблюдаемая $(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})$ определяет проективное измерение [1], и возможные результаты измерения принадлежат множеству чисел $\{+1;\ -1\}$. Результаты этого измерения можно интерпретировать как произведение результатов измерений наблюдаемых $\mathbf{v}\cdot\boldsymbol{\sigma}$ и $\mathbf{w}\cdot\boldsymbol{\sigma}$.

Обозначим через $P_{\mathbf{vw}}(1)$ и $P_{\mathbf{vw}}(-1)$ вероятности получения значений "1" и "−1" при проективном измерении определенной наблюдаемой $(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})$ над квантовой системой AB из двух кубитов A и B, находящейся в состоянии "спиновый синглет" $|\psi_{11}\rangle$.

Если векторы $\mathbf{v}$ и $\mathbf{w}$ равны, то из утверждения 1 следует, что справедливы равенства: $P_{\mathbf{vw}}(1) = 0$ и $P_{\mathbf{vw}}(-1) = 1$. Поставим задачу вычисления значений вероятностей $P_{\mathbf{vw}}(1)$ и $P_{\mathbf{vw}}(-1)$ в общем случае, когда необязательно выполнение условия совпадения векторов $\mathbf{v}$ и $\mathbf{w}$. Решение этой задачи представим в виде следующего утверждения.

Утверждение 2. Справедливы следующие равенства:

$$P_{\mathbf{vw}}(1) = \frac{1}{2}\left(1 - (\mathbf{v},\ \mathbf{w})\right),\ P_{\mathbf{vw}}(-1) = \frac{1}{2}\left(1 + (\mathbf{v},\ \mathbf{w})\right),$$

где $(\mathbf{v},\ \mathbf{w}) = v_1w_1 + v_2w_2 + v_3w_3$ — скалярное произведение векторов $\mathbf{v} = (v_1,\ v_2,\ v_3)$ и $\mathbf{w} = (w_1,\ w_2,\ w_3) \in \mathbf{R}^3$, $v_1^2 + v_2^2 + v_3^2 = 1$, $w_1^2 + w_2^2 + w_3^2 = 1$; $P_{\mathbf{vw}}(1)$

и $P_{\mathbf{vw}}(-1)$ — соответственно, вероятности получения значений "1" и "−1" при проективном измерении определенной наблюдаемой $(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})$ над квантовой системой AB из двух кубитов A и B, находящейся в состоянии "спиновый синглет" $|\psi_{11}\rangle$.

Доказательство. Для среднего значения $\langle(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})\rangle$ наблюдаемой $(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})$ при проведении проективных измерений над квантовой системой AB, находящейся в состоянии "спиновый синглет" $|\psi_{11}\rangle$, справедлива следующая цепочка равенств [1]:

$$\begin{aligned}\langle(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})\rangle &= 1P_{\mathbf{vw}}(1)+(-1)P_{\mathbf{vw}}(-1)=\\ &=\langle\psi_{11}|((\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma}))|\psi_{11}\rangle,\end{aligned}\qquad(11)$$

где $\langle\psi_{11}|$ — вектор-строка, двойственная к $|\psi_{11}\rangle$. Учитывая, что $P_{\mathbf{vw}}(-1)=1-P_{\mathbf{vw}}(1)$, из последнего равенства цепочки равенств (11) получаем

$$2P_{\mathbf{vw}}(1)-1=\langle\psi_{11}|((\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma}))|\psi_{11}\rangle.$$

Отсюда следует, что

$$P_{\mathbf{vw}}(1)=\frac{1}{2}\left(1+\langle\psi_{11}|((\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma}))|\psi_{11}\rangle\right)\qquad(12)$$

и, следовательно,

$$P_{\mathbf{vw}}(-1)=\frac{1}{2}\left(1-\langle\psi_{11}|((\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma}))|\psi_{11}\rangle\right).\qquad(13)$$

Далее, вычисляя, имеем:

$$(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})=\begin{pmatrix}v_3 & v_1-iv_2\\ v_1+iv_2 & -v_3\end{pmatrix}\otimes\begin{pmatrix}w_3 & w_1-iw_2\\ w_1+iw_2 & -w_3\end{pmatrix}=$$

$$=\begin{pmatrix}v_3w_3 & v_3(w_1-iw_2) & (v_1-iv_2)w_3 & (v_1-iv_2)(w_1-iw_2)\\ v_3(w_1+iw_2) & -v_3w_3 & (v_1-iv_2)(w_1+iw_2) & -(v_1-iv_2)w_3\\ (v_1+iv_2)w_3 & (v_1+iv_2)(w_1-iw_2) & -v_3w_3 & -v_3(w_1-iw_2)\\ (v_1+iv_2)(w_1+iw_2) & -(v_1+iv_2)w_3 & -v_3(w_1+iw_2) & v_3w_3\end{pmatrix}.$$

Отсюда получаем

$$\langle\psi_{11}|((\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma}))|\psi_{11}\rangle=\begin{pmatrix}0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0\end{pmatrix}((\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma}))\begin{pmatrix}0\\ \frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\\ 0\end{pmatrix}=\frac{1}{2}\begin{pmatrix}0 & 1 & -1 & 0\end{pmatrix}((\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma}))\begin{pmatrix}0\\ 1\\ -1\\ 0\end{pmatrix}=$$

$$=\frac{1}{2}(-v_3w_3-v_1w_1-iv_1w_2+iv_2w_1-v_2w_2-v_1w_1+iv_1w_2-iv_2w_1-v_2w_2-v_3w_3)=$$

$$=\frac{1}{2}(-2v_1w_1-2v_2w_2-2v_3w_3)=-(v_1w_1+v_2w_2+v_3w_3)=-(\mathbf{v},\mathbf{w}).$$

Сравнивая начальное и конечное выражения предыдущей цепочки равенств, убеждаемся в том, что среднее значение $\langle(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})\rangle$ наблюдаемой $(\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma})$ для квантовой системы, находящейся в состоянии "спиновый синглет" $|\psi_{11}\rangle$, равно взятому со знаком минус скалярному произведению векторов $\mathbf{v}$ и $\mathbf{w}$, т. е. справедливо равенство

$$\langle\psi_{11}|((\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma}))|\psi_{11}\rangle=-(\mathbf{v},\mathbf{w}).\qquad(14)$$

Пользуясь равенством (14), путем замены в правых частях равенств (12) и (13) выражения $\langle\psi_{11}|((\mathbf{v}\cdot\boldsymbol{\sigma})\otimes(\mathbf{w}\cdot\boldsymbol{\sigma}))|\psi_{11}\rangle$ на выражение $(-(\mathbf{v},\mathbf{w}))$ убеждаемся в справедливости равенств для вероятностей $P_{\mathbf{vw}}(1)$ и $P_{\mathbf{vw}}(-1)$. Утверждение 2 доказано.

Замечание 2. Утверждение 2 составляет математическую основу результатов, представленных в разд. 3. Кроме того, представляется уместным указать на то, что равенство (14) приводится с другим доказательством (отличным от того, что приведено в данной статье), в работе [23].

2. Квантовая криптографическая система АКМ2017

Пусть сгенерировано достаточное число $N\in\mathbf{N}$ пар кубитов A_jB_j (где $j=\overline{1,N}$) в состоянии Белла

$$|\psi_{11}\rangle=\frac{|01\rangle-|10\rangle}{\sqrt{2}},$$

где $\mathbf{N}$ — множество натуральных чисел.

Кубиты массива пар $\{A_jB_j|\ j=\overline{1,N}\}$ разделены так, что массив кубитов $\{A_j\}$ составляет исходящий шифрблокнот Алисы, а массив кубитов $\{B_j\}$ составляет входящий шифрблокнот Боба. Алиса и Боб разделены в пространстве.

Может ли быть решена следующая задача?

Алиса должна передать Бобу сообщение, имеющее в двоичном виде представление

$$m=m_1,\ m_2,\ \ldots,\ m_L,$$

длины $L \leqslant N$ бит, зашифровав его с использованием своего исходящего блокнота, а Боб должен получить и расшифровать сообщение с использованием своего входящего блокнота. При этом предполагается, что Алиса и Боб располагают дополнительно еще общедоступным (открытым) классическим каналом связи.

Ответ: да. Для подтверждения истинности этого ответа изложим решение данной задачи, представленное в работах [2, 11].

Алиса выбирает случайным образом (например, используя подходящий генератор случайных чисел) единичный вектор $\mathbf{v} = (v_1, v_2, v_3)$ — (т. е. вектор $\mathbf{v}$ является нормированным вектором [13]) в трехмерном пространстве над полем действительных чисел $\mathbf{R}$.

Вектор $\mathbf{v}$ является **сеансовым (разовым) ключом** и используется для зашифрования только **одного** данного сообщения. Вектор $\mathbf{v}$ будет передан Бобу после завершения процесса зашифрования сообщения m вместе с зашифрованным сообщением (например, по предварительной договоренности в начале криптограммы перед зашифрованным сообщением) по классическому каналу.

Будем полагать, что Алиса осуществляет зашифрование сообщения m последовательно по одному биту. Для зашифрования двоичного символа m_j, где $j = \overline{1, L}$, Алиса осуществляет следующие действия:

1) выполняет измерение наблюдаемой $\mathbf{v} \cdot \boldsymbol{\sigma}$ для кубита A_j и в зависимости от результата измерения "1" или "−1" полагает значение j-го знака γ_j двоичной гаммы $\gamma = \gamma_1, \gamma_2, ..., \gamma_L$ равным 0 или 1 соответственно;

2) вычисляет значение j-го знака s_j криптограммы (зашифрованного сообщения) $s = s_1, s_2, ..., s_L$ через равенство $s_j = m_j \oplus \gamma_j$, где $\oplus$ — знак операции сложения по модулю 2, $j = \overline{1, L}$.

Так как биты сообщения m зашифровываются независимо, то возможно распараллеливание процесса зашифрования без ограничений. После завершения зашифрования сообщения m Алиса передает Бобу пару $(\mathbf{v}, s)$ (т. е. криптограмму) по открытому классическому каналу связи.

Получив криптограмму $(\mathbf{v}, s)$, Боб выполняет процедуру расшифрования.

Для расшифрования двоичного символа s_j, где $j = \overline{1, L}$, Боб осуществляет следующие действия:

1) выполняет измерение наблюдаемой $\mathbf{v} \cdot \boldsymbol{\sigma}$ для кубита B_i и получает результат измерения "1" или "−1", противоположный, в соответствии с Утверждением 1, с результатом, полученным Алисой при зашифровании знака m_j; далее Боб, в зависимости от полученного результата измерения "1" или "−1", полагает значение j-го знака γ_j двоичной гаммы $\gamma = \gamma_1, \gamma_2, ..., \gamma_L$ равным 1 или 0 соответственно (напомним, что у Алисы знак гаммы был равен 0 при получении результата ее измерения "1", а при результате измерения "−1" знак гаммы был равен 1);

2) вычисляет значение j-го знака m_j сообщения $m = m_1, m_2, ..., m_L$ через равенство $m_j = s_j \oplus \gamma_j$.

Так как биты сообщения m расшифровываются независимо, то возможно распараллеливание процесса расшифрования без ограничений.

Еще раз отметим, что в изложенном описании квантовой криптографической системы использовалось то, что **знаки двоичных гамм, сформированных и Алисой, и Бобом, совпадают**. Это следует из утверждения 1.

Описанную квантовую криптографическую систему называют квантовая криптографическая система AKM2017 [2, 11], где буквы A, K и M — первые буквы фамилий Алиев, Корольков и Матвеев, а 2017 — год, в котором криптографическая система была представлена научной общественности в порядке обсуждения.

3. Степень различия гамм зашифрования и расшифрования. Степень различия сеансовых ключей

Из описания криптографической системы AKM2017 (см. разд. 2) следует, что сеансовым ключом является единичный вектор $\mathbf{v} = (v_1, v_2, v_3)$ в трехмерном пространстве над полем действительных чисел $\mathbf{R}$ (т. е. $\mathbf{v} = (v_1, v_2, v_3) \in \mathbf{R}^3$, где $v_1^2 + v_2^2 + v_3^2 = 1$), выбираемый Алисой (стороной, реализующей процесс зашифрования) случайным образом с использованием подходящего генератора случайных чисел. Вектор $\mathbf{v}$ передается в открытом (общедоступном) виде Бобу после завершения процесса зашифрования некоторого сообщения m вместе с зашифрованным сообщением s (например, по предварительной договоренности в начале криптограммы перед зашифрованным сообщением) по классическому каналу. То есть после завершения зашифрования открытого сообщения m Алиса передает Бобу пару $(\mathbf{v}, s)$ (криптограмму) по открытому классическому каналу связи, где $\mathbf{v} = (v_1, v_2, v_3)$, s — зашифрованное сообщение.

Получив криптограмму $(\mathbf{v}, s)$, Боб выполняет процедуру расшифрования. Для этого он использует сеансовый ключ $\mathbf{v} = (v_1, v_2, v_3)$ для формирования ключа расшифрования.

Предположим, что Боб использует для расшифрования ключ $\mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3$, где $w_1^2 + w_2^2 + w_3^2 = 1$. Такая ситуация может возникнуть, например, при неисправности или неточности аппаратуры Боба, или при искажении сеансового ключа в процессе его передачи от Алисы к Бобу в составе криптограммы, или при деструктивных действиях противника и т. д. В результате Боб сгенерирует для расшифрования гамму, отличную от истинной гаммы, сгенерированной и использованной Алисой на сеансовом ключе $\mathbf{v} = (v_1, v_2, v_3)$ при реализации процесса зашифрования. В связи с этим **поставим задачу определения степени различия (или, по другому, расходимости) гаммы расшифрования от гаммы зашифрования в зависимости от степени различия сеансовых ключей зашифрования v и расшифрования w.**

Пусть $\boldsymbol{\gamma}_з = \boldsymbol{\gamma}_{з1}, \boldsymbol{\gamma}_{з2}, ..., \boldsymbol{\gamma}_{зL}$ — гамма зашифрования, сгенерированная Алисой на сеансовом ключе $\mathbf{v} = (v_1, v_2, v_3)$; $\boldsymbol{\gamma}_р = \boldsymbol{\gamma}_{р1}, \boldsymbol{\gamma}_{р2}, ..., \boldsymbol{\gamma}_{рL}$ — гамма расшифрования, сгенерированная Бобом на сеансовом ключе $\mathbf{w} = (w_1, w_2, w_3)$, где L — длина гаммы (число элементов), $L \in \mathbf{N}$.

Положим $P_j(\mathbf{v}, \mathbf{w})$ — это вероятность того, что сумма по модулю два двоичных значений $\gamma_{зj}$ и $\gamma_{рj}$ равна 1, т. е.

$$P_j(\mathbf{v}, \mathbf{w}) = P(\gamma_{3j} \oplus \gamma_{pj} = 1), \ j = \overline{1, L}.$$

Из независимости и тождественности вероятностных процессов в каждом такте $j = \overline{1, L}$ генерации гамм $\gamma_з$ и $\gamma_р$ следует, что $P_j(\mathbf{v}, \mathbf{w})$ не зависит от индекса $j = \overline{1, L}$. Поэтому можно полагать, что

$$P(\mathbf{v}, \mathbf{w}) = P(\gamma_{3j} \oplus \gamma_{pj} = 1), \ j = \overline{1, L}, \qquad (15)$$

т. е. $P(\mathbf{v}, \mathbf{w})$ — вероятность несовпадения на такте с номером j знаков гаммы зашифрования на сеансовом ключе $\mathbf{v}$ и гаммы расшифрования на сеансовом ключе $\mathbf{w}$, где $j = \overline{1, L}$.

Определение 1. Для квантовой криптографической системы AKM2017 число $P(\mathbf{v}, \mathbf{w})$, равное вероятности несовпадения знаков гаммы зашифрования на сеансовом ключе $\mathbf{v}$ и гаммы расшифрования на сеансовом ключе $\mathbf{w}$ (на любом такте) называется *степенью различия* гамм зашифрования $\gamma_з$ и расшифрования $\gamma_р$, сгенерированных соответственно на сеансовых ключах $\mathbf{v}$ и $\mathbf{w}$.

Замечание 3. *Степень различия* гамм зашифрования и расшифрования имеет следующий смысл: если L (как выше) — длина криптограммы (т. е. число символов в зашифрованном тексте), то математическое ожидание числа неправильно расшифрованных символов равно $P(\mathbf{v}, \mathbf{w})L$.

Определение 2. Для любых двух сеансовых ключей

$$\mathbf{v} = (v_1, v_2, v_3), \mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3,$$

(где $v_1^2 + v_2^2 + v_3^2 = 1, \quad w_1^2 + w_2^2 + w_3^2 = 1$) квантовой криптографической системы AKM2017 число $k(\mathbf{v}, \mathbf{w})$, заданное равенством

$$k(\mathbf{v}, \mathbf{w}) = 0{,}5(1 - (v_1w_1 + v_2w_2 + v_3w_3)), \qquad (16)$$

назовем *степенью различия* сеансовых ключей $\mathbf{v}$ и $\mathbf{w}$.

Замечание 4. Приведем геометрическую интерпретацию введенной *степени различия* $k(\mathbf{v}, \mathbf{w})$ сеансовых ключей $\mathbf{v}$ и $\mathbf{w}$. Множество сеансовых ключей $\{\mathbf{v} = (v_1, v_2, v_3) \in \mathbf{R}^3 | \, v_1^2 + v_2^2 + v_3^2 = 1\}$ квантовой криптографической системы AKM2017 образует в трехмерном евклидовом пространстве с декартовой системой координат $OXYZ$ поверхность сферы S радиуса $r = 1$ с центром в точке O с координатами (0, 0, 0), заданной уравнением

$$x^2 + y^2 + z^2 = 1.$$

Тогда для скалярного произведения $(\mathbf{v}, \mathbf{w})$ любых двух сеансовых ключей

$$\mathbf{v} = (v_1, v_2, v_3), \mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3,$$

(где $v_1^2 + v_2^2 + v_3^2 = 1, \quad w_1^2 + w_2^2 + w_3^2 = 1$) квантовой криптографической системы AKM2017 имеет место цепочка равенств

$$(\mathbf{v}, \mathbf{w}) = v_1w_1 + v_2w_2 + v_3w_3 = 1 \cdot 1 \cos\theta = \cos\theta,$$

где θ — угол между векторами $\mathbf{v}$ и $\mathbf{w}$. Отсюда, с учетом равенства (16), получаем, что степень различия $k(\mathbf{v}, \mathbf{w})$ сеансовых ключей $\mathbf{v}$ и $\mathbf{w}$ удовлетворяет равенству

$$k(\mathbf{v}, \mathbf{w}) = 0{,}5(1 - \cos\theta). \qquad (17)$$

Непосредственными следствиями равенства (17) являются: равенство

$$k(\mathbf{v}, \mathbf{w}) = \sin^2\left(\frac{\theta}{2}\right) \qquad (18)$$

и двойное неравенство

$$0 \leqslant k(\mathbf{v}, \mathbf{w}) \leqslant 1.$$

Утверждение 3. Для любых двух сеансовых ключей

$$\mathbf{v} = (v_1, v_2, v_3), \mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3$$

(где $v_1^2 + v_2^2 + v_3^2 = 1, \ w_1^2 + w_2^2 + w_3^2 = 1$) квантовой криптографической системы AKM2017 *степень различия* $P(\mathbf{v}, \mathbf{w})$ гамм зашифрования $\gamma_з$ и расшифрования $\gamma_р$, сгенерированных соответственно на сеансовых ключах $\mathbf{v}$ и $\mathbf{w}$, совпадает со *степенью различия* $k(\mathbf{v}, \mathbf{w})$ сеансовых ключей $\mathbf{v}$ и $\mathbf{w}$, т. е. справедливо равенство

$$P(\mathbf{v}, \mathbf{w}) = k(\mathbf{v}, \mathbf{w}).$$

Доказательство. Из равенства (15) следует, что $P(\mathbf{v}, \mathbf{w}) = P(\gamma_{3j} \oplus \gamma_{pj} = 1)$ для любого номера j такта генерации гаммы. Из описания квантовой криптографической системы AKM2017 (см. разд. 2) следует, что выполнение равенства $\gamma_{3j} \oplus \gamma_{pj} = 1$ равносильно получению значения 1 в результате вычисления значения наблюдаемой $(\mathbf{v} \cdot \boldsymbol{\sigma}) \otimes (\mathbf{w} \cdot \boldsymbol{\sigma})$ путем проведения проективных измерений над соответствующей двухкубитной квантовой системой, находящейся в состоянии "спиновый синглет", т. е. справедливо равенство

$$P(\gamma_{3j} \oplus \gamma_{pj} = 1) = P_{\mathbf{vw}}(1)$$

для вероятностей указанных событий. Кроме этого, из утверждения 2 следует справедливость равенства $P_{\mathbf{vw}}(1) = \frac{1}{2}(1 - (\mathbf{v}, \mathbf{w}))$.

Из последних двух равенств следует справедливость равенства $P(\mathbf{v}, \mathbf{w}) = \frac{1}{2}(1 - (\mathbf{v}, \mathbf{w}))$ для *степени различия* $P(\mathbf{v}, \mathbf{w})$ гамм зашифрования $\gamma_з$ и расшифрования $\gamma_р$.

Отсюда, с учетом равенства (16), получаем искомое равенство. Утверждение доказано.

Непосредственно из утверждения 3, с учетом равенства (18), вытекает следствие 1.

Следствие 1. Для любых двух сеансовых ключей

$$\mathbf{v} = (v_1, v_2, v_3), \mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3$$

(где $v_1^2 + v_2^2 + v_3^2 = 1, \quad w_1^2 + w_2^2 + w_3^2 = 1$) квантовой криптографической системы AKM2017 *степень различия* $P(\mathbf{v}, \mathbf{w})$ гамм зашифрования $\gamma_з$ и расшифрования $\gamma_р$, сгенерированных соответственно на сеансовых ключах $\mathbf{v}$ и $\mathbf{w}$, равна $\sin^2\left(\frac{\theta}{2}\right)$, где θ — угол между векторами $\mathbf{v}$ и $\mathbf{w}$, т. е. справедливо равенство

$$P(\mathbf{v}, \mathbf{w}) = \sin^2\left(\frac{\theta}{2}\right).$$

Непосредственно из утверждения 3, с учетом замечания 3, вытекает следствие 2.

Следствие 2. Для квантовой криптографической системы AKM2017 математическое ожидание числа неправильных (искаженных) элементов открытого текста, полученного после расшифрования криптограммы, равно произведению длины (числа двоичных символов) L криптограммы на числовое значение $k(\mathbf{v}, \mathbf{w})$ степени различия использованных ключей зашифрования $\mathbf{v}$ и расшифрования $\mathbf{w}$.

Замечание 5. При формулировке следствия 2 предполагается, что в зашифрованном тексте (криптограмме) искажения отсутствуют.

Замечание 6. Приведем геометрическую интерпретацию полученных результатов для произвольного сеансового ключа зашифрования $\mathbf{v} = (v_1, v_2, v_3) \in \mathbf{R}^3$ (где $v_1^2 + v_2^2 + v_3^2 = 1$), которую можно представить в виде совокупности следующих 5 позиций.

1. Множество сеансовых ключей расшифрования $\{\mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3 | w_1^2 + w_2^2 + w_3^2 = 1\}$ квантовой криптографической системы AKM2017 составляют [9] в трехмерном евклидовом пространстве с декартовой системой координат $OXYZ$ поверхность сферы S радиуса $r = 1$ с центром в точке O с координатами (0, 0, 0), заданной уравнением $x^2 + y^2 + z^2 = 1$.

2. Вектор $\mathbf{v} = (v_1, v_2, v_3) \in \mathbf{R}^3$ (где $v_1^2 + v_2^2 + v_3^2 = 1$) — это единичный вектор в трехмерном евклидовом пространстве с декартовой системой координат $OXYZ$.

3. Для $\mathbf{v} = (v_1, v_2, v_3)$ множество точек $\mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3$ (где $w_1^2 + w_2^2 + w_3^2 = 1$), дающих одно и то же значение *степени различия* (т. е. класс ключей расшифрования, дающих одно и то же значение T *степени различия*)

$P(\mathbf{v}, \mathbf{w}) = 0{,}5(1 - (v_1w_1 + v_2w_2 + v_3w_3)) = T$, $(0 \leqslant T \leqslant 1)$, гаммы зашифрования на сеансовом ключе $\mathbf{v}$ и гаммы расшифрования на сеансовом ключе $\mathbf{w}$ образуют окружность N на поверхности сферы S (т. е. класс ключей расшифрования, дающих одно и то же значение *степени различия* T, это окружность). При этом N — это окружность радиуса

$$r = \sqrt{1 - (1 - 2T)^2} = \sqrt{4T - 4T^2} = 2\sqrt{T - T^2}$$

с центром в точке $((1 - 2T)v_1, (1 - 2T)v_2, (1 - 2T)v_3)$, заданная системой уравнений

$$\begin{cases} v_1x + v_2y + v_3z = 1 - 2T, \\ x^2 + y^2 + z^2 = 1, \end{cases}$$

где $x = w_1$, $y = w_2$, $z = w_3$.

4. При $\mathbf{v} = (v_1, v_2, v_3) \in \mathbf{R}^3$ (где $v_1^2 + v_2^2 + v_3^2 = 1$) множество точек $\mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3$ (где $w_1^2 + w_2^2 + w_3^2 = 1$), дающих одно и то же значение степени различия $P(\mathbf{v}, \mathbf{w}) = 0$, состоит из одной точки M_1 с координатами (v_1, v_2, v_3).

5. При $\mathbf{v} = (v_1, v_2, v_3) \in \mathbf{R}^3$ (где $v_1^2 + v_2^2 + v_3^2 = 1$) множество точек $\mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3$ (где $w_1^2 + w_2^2 + w_3^2 = 1$), дающих одно и то же значение степени различия $P(\mathbf{v}, \mathbf{w}) = 1$, состоит из одной точки M_2 с координатами $(-v_1, -v_2, -v_3)$.

Из положений, представленных в виде замечания 6, вытекает следующее утверждение.

Утверждение 4. Пусть T — произвольное число, удовлетворяющее двойному неравенству $0 \leqslant T \leqslant 1$. Для любого сеансового ключа $\mathbf{v} = (v_1, v_2, v_3) \in \mathbf{R}^3$ (где $v_1^2 + v_2^2 + v_3^2 = 1$) квантовой криптографической системы AKM2017 вероятность того, что *степень различия* $P(\mathbf{v}, \mathbf{w})$ гамм зашифрования $\gamma_з$ и расшифрования $\gamma_р$, сгенерированных соответственно на сеансовых ключах $\mathbf{v}$ и $\mathbf{w}$ при случайном равномерном выборе сеансового ключа $\mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3$ (где $w_1^2 + w_2^2 + w_3^2 = 1$) из всего множества сеансовых ключей, удовлетворяет неравенству $P(\mathbf{v}, \mathbf{w}) \leqslant T$, равна числу T, т. е. $P(P(\mathbf{v}, \mathbf{w}) \leqslant T) = T$.

Доказательство. Доказательство проведем, обратившись к аналитическому аппарату геометрических вероятностей [6]. Полное множество сеансовых ключей $\mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3$ (где $w_1^2 + w_2^2 + w_3^2 = 1$) совпадает с поверхностью сферы S радиуса $r = 1$ с центром в точке O с координатами (0, 0, 0), заданной уравнением $x^2 + y^2 + z^2 = 1$.

Площадь поверхности сферы S равна 4π.

Из замечания 6 следует, что множество сеансовых ключей $\mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3$ (где $w_1^2 + w_2^2 + w_3^2 = 1$), для которых выполняется неравенство $P(\mathbf{v}, \mathbf{w}) \leqslant T$, совпадает со сферическим сегментом C сферы S, причем высота сегмента C равна $2T$ и, следовательно, площадь сферического сегмента C равна $2\pi 2T = 4\pi T$.

Тогда случайный равномерный выбор (наудачу [6]) сеансового ключа $\mathbf{w} = (w_1, w_2, w_3) \in \mathbf{R}^3$ (где $w_1^2 + w_2^2 + w_3^2 = 1$) из полного множества сеансовых ключей означает выполнение следующих предположений: в качестве $\mathbf{w}$ может оказаться любая точка сферы S, вероятность принадлежности $\mathbf{w}$ любой фигуре [6] g на поверхности сферы S пропорциональна площади g и не зависит ни от ее расположения относительно S, ни от формы g. Тогда в этих предположениях в соответствии с [6] вероятность принадлежности сеансового ключа $\mathbf{w}$ сферическому сегменту $g = C$ определяется равенством

$$P\left(P(\mathbf{v}, \mathbf{w}) \leqslant T\right) = \frac{\text{Площадь } g}{\text{Площадь } S} = \frac{4\pi T}{4\pi} = T,$$

что и требовалось доказать.

Заключение

Представленные в статье применительно к квантовой криптографической системе AKM2017 результаты полностью проясняют в количественном отношении спектр возможных ошибок при расшифровании, связанных с несовпадением сеансовых ключей зашифрования и расшифрования.

В статье даны ответы на вопрос о том, как эти результаты можно использовать при практическом применении AKM2017 для защиты информации. К числу таких применений в контексте данной статьи относится построение механизма исправления возможных ошибок при расшифровании, обусловленных некорректной установкой (по разным причинам) сеансового ключа расшифрования. Такой механизм

может быть основан на применении методов и способов помехоустойчивого кодирования [5, 8, 12, 15, 20]. Например, открытый текст перед зашифрованием закодировать кодом, исправляющим заданное число ошибок. Результаты данной статьи могут быть использованы для установления оптимальных значений параметров этого кода. Такой подход позволит в случае с квантовой криптографической системой АКМ2017 полностью решить задачу исправления ошибок, обусловленных возможной некорректностью при установке сеансовых ключей.

Список литературы

1. **Алиев Ф. К., Корольков А. В., Матвеев Е. А.** Несепарабельные состояния многокубитных квантовых систем. Монография / Под ред. Ф. К. Алиева. — М.: Радиотехника. 2017. — 320 с.
2. **Алиев Ф. К., Корольков А. В., Матвеев Е. А., Орлов С. С., Шеремет И. А.** Квантовая криптографическая система АКМ2017 на основе ресурса несепарабельности состояния спиновой синглет // Системы высокой доступности. — 2018. — Т. 14, № 4. — С. 61—72.
3. **Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В.** Основы криптографии: Учебное пособие. — М.: Гелиос АРВ, 2005. — 480 с.
4. **Бабаш А. В., Шанкин Г. П.** Криптография. — М.: СОЛОН-Р, 2002. — 512 с.
5. **Вернер М.** Основы кодирования. — М.: Техносфера, 2006. — 288 с.
6. **Гмурман В. Е.** Теория вероятностей и математическая статистика. Учеб. пособие для вузов. Изд. 7-е, стер. — М.: Высш. Шк., 2000. — 479 с.
7. **Граймс Р. А.** Апокалипсис криптографии. — М.: ДМК Пресс, 2020. — 290 с.
8. **Духин А. А.** Теория информации. — М.: Гелиос АРВ, 2007. — 248 с.
9. **Корн Г., Корн Т.** Справочник по математике для научных работников и инженеров. — М.: Наука, 1984 — 833 с.
10. **Лось А. Б., Нестеренко А. Ю., Рожков М. И.** Криптографические методы защиты информации. — М.: Юрайт, 2016. — 473 с.
11. **Матвеев Е. А.** Применение квантовомеханических эффектов в системах защиты информации: дис. ... канд. физ.-мат. наук. — Пенза: Научно-техническое предприятие КРИПТОСОФТ, 2019. — 157 с.
12. **Морелос-Сарагоса Р.** Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. — М.: Техносфера, 2005. — 320 с.
13. **Нильсен М., Чанг И.** Квантовые вычисления и квантовая информация. — М.: Мир, 2006. — 824 с.
14. **Прескилл Дж.** Квантовая информация и квантовые вычисления. Том 1. — М. — Ижевск: НИЦ "Регулярная и хаотическая динамика", 2008. — 464 с.
15. **Стратонович Р. Л.** Теория информации. — М.: Сов. радио, 1975. — 424 с.
16. **Фомичев В. М.** Дискретная математика и криптология. — М.: ДИАЛОГ-МИФИ, 2003. — 400 с.
17. **Фомичев В. М.** Методы дискретной математики в криптологии. — М.: ДИАЛОГ-МИФИ, 2010. — 424 с.
18. **Фомичев В. М., Мельников Д. А.** Криптографические методы защиты информации. В 2 ч. Часть 1, Математические аспекты. — М.: Юрайт, 2016. — 200 с. Часть 2, Системные и прикладные аспекты. — М.: Юрайт, 2016. — 245 с.
19. **Хидари Дж. Д.** Квантовые вычисления. — М.: ДМК Пресс, 2021. — 370 с.
20. **Чечета С. И.** Введение в дискретную теорию информации и кодирования. — М.: МЦНМО, 2011. — 224 с.
21. **Шеннон К.** Работы по теории информации и кибернетике. — М.: ИЛ, 1963. — 869 с.
22. **Шнаер Б.** Прикладная криптография. — М.: ТРИУМФ, 2003. — 816 с.
23. **Jordan T. F.** Quantum mechanics in simple matrix form. John Wiley & Sons, 1986, Inc. 271 p.

On the Sensitivity of the Gamma of the Quantum Cryptographic System AKM2017 to Changes in the Session Key

F. K. Aliev, DIS of MD RF, Moscow, Russian Federation,
A. V. Korolkov, Russian Technological University — MIREA, AK RF, Moscow, Russian Federation,
E. A. Matveev, eugene.cs@hotmail.com, Cryptosoft, Penza, 440026, Russian Federation,
I. A. Sheremet, RFBR, Moscow, Russian Federation

Corresponding author:

Matveev Evgeniy A., Director, Cryptosoft, Penza, 440026, Russian Federation
E-mail: eugene.cs@hotmail.com

Received on December 16, 2020
Accepted on February 24, 2021

The quantum cryptographic system AKM2017 is considered. The results of the analysis of the dependence of the degree of difference between the encryption and decryption gamut on the degree of difference between the corresponding session keys are presented. The equality of these degrees of distinction is revealed and substantiated. For an arbitrarily fixed encryption session key, the distribution of session decryption keys by classes is revealed and described, depending on the value of the degree of difference between the encryption and decryption gamuts. One class is made up of all session decryption keys, leading to the same value of the degree of difference between the encryption and decryption gamuts. A geometric interpretation of the specified distribution by classes is given in the form of placement along circles (class is a circle) on the surface of a sphere of unit radius centered at the origin of the Euclidean rectangular coordinate system in a three-dimensional linear space over the field of real numbers.

The stated results can be used to solve the problems of optimizing the values of the parameters of accuracy and reliability of the functioning of variants of practical implementations of the quantum cryptographic system AKM2017, for example, when setting up a session decryption key, which makes it possible to guarantee a predetermined small value of the mathematical expectation of the number of incorrectly decrypted binary plain text.

Keywords: *quantum cryptography, cryptographic system, quantum computer, qubit, quantum system, inseparable (synonym — entangled) states, theoretical security, cryptographic technique, spin singlet state, measurement of the spin component along the axis*

For citation:

Aliev F. K., Korolkov A. V., Matveev E. A., Sheremet I. A. On the Sensitivity of the Gamma of the Quantum Cryptographic System AKM2017 to Changes in the Session Key, *Programmnaya Ingeneria*, 2021, vol. 12, no. 4, pp. 179—188.

DOI: 10.17587/prin.12.179-188

References

1. **Aliev F. K., Korolkov A. V., Matveev E. A.** *Inseparable states of multi-qubit quantum systems.* Monograph / Eds. F. K. Aliev, Moscow, Radiotekhnika, 2017, 320 p. (in Russian).
2. **Aliev F. K., Korolkov A. V., Matveev E. A., Orlov S. S., Sheremet I. A.** Quantum cryptographic system AKM2017 based on the resource inseparability of the spin singlet state, *Sistemy vysokoj dostupnosti*, 2018, vol. 14, no. 4, pp. 61—72 (in Russian).
3. **Alferov A. P., Zubov A. Yu., Kuzmin A. S., Cheremushkin A. V**. *Fundamentals of Cryptography: A Tutorial,* Moscow, Helios ARV, 2005, 480 p. (in Russian).
4. **Babash A. V., Shankin G. P.** *Cryptography,* Moscow, SOLON-R, 2002, 512 p. (in Russian).
5. **Werner M.** *Foundations of coding*, Moscow, Technosphere, 2006, 288 p. (in Russian).
6. **Gmurman V. E.** *Theory of Probability and Mathematical Statistics.* Textbook. manual for universities. Ed. 7th, erased. Moscow, Vysshaya Shkola, 2000, 479 p. (in Russian).
7. **Grimes R. A.** *Apocalypse of cryptography*, Moscow, DMK Press, 2020, 290 p. (in Russian).
8. **Dukhin A. A.** *Information theory,* Moscow, Helios ARV, 2007, 248 p. (in Russian).
9. **Korn G., Korn T.** *Handbook of mathematics for scientists and engineers*, Moscow, Nauka, 1984, 833 p. (in Russian).
10. **Los A. B., Nesterenko A. Yu., Rozhkov M. I.** *Cryptographic methods of information protection,* Moscow, Yurayt, 2016, 473 p. (in Russian).
11. **Matveev E. A.** Application of quantum mechanical effects in information security systems. Cand. diss. Penza, Scientific and technical enterprise CRYPTOSOFT, 2019, 157 p. (in Russian).
12. **Morelos-Zaragoza R.** *The art of noise-immune coding. Methods, algorithms, application,* Moscow, Technosphere, 2005, 320 p. (in Russian).
13. **Nielsen M., Chang I.** *Quantum computing and quantum information*, Moscow, Mir, 2006, 824 p. (in Russian).
14. **Preskill J.** *Quantum information and quantum computing.* Volume 1. Moscow, Izhevsk, Research Center "Regular and Chaotic Dynamics", 2008, 464 p. (in Russian).
15. **Stratonovich R. L.** *Information theory,* Moscow, Sov. radio, 1975, 424 p. (in Russian).
16. **Fomichev V. M.** *Discrete mathematics and cryptology*, Moscow, DIALOG-MEPhI, 2003, 400 p. (in Russian).
17. **Fomichev V. M.** *Methods of discrete mathematics in cryptology*, Moscow, DIALOG-MEPhI, 2010, 424 p. (in Russian).
18. **Fomichev V. M., Melnikov D. A.** Cryptographic methods of information protection. In 2 h. *Part 1*, *Mathematical aspects*, Moscow, Yurayt, 2016, 200 p. *Part 2, Systemic and applied aspects.* Moscow, Yurayt, 2016, 245 p. (in Russian).
19. **Khidari J. D.** *Quantum computing,* Moscow, DMK Press, 2021, 370 p. (in Russian).
20. **Checheta S. I.** *An introduction to discrete information and coding theory*, Moscow, MTsNMO, 2011, 224 p. (in Russian).
21. **Shannon K.** *Works on information theory and cybernetics*, Moscow, IL, 1963, 869 p. (in Russian).
22. **Schnaer B.** *Applied cryptography,* Moscow, TRIUMPH, 2003, 816 p. (in Russian).
23. **Jordan T. F.** *Quantum mechanics in simple matrix form*, John Wiley & Sons, Inc., 1986, 271 p.

ИНФОРМАЦИЯ

XIV Всероссийская Мультиконференция по проблемам управления (МКПУ-2021) 27 сентября – 2 октября 2021 г.

с. Дивноморское, Геленджик,
Краснодарский край, Россия

Мультиконференция включает **четыре локальные научно-технические конференции:**

- Робототехника и мехатроника (РиМ-2021)
- Управление в распределенных и сетевых системах (УРСС-2021)
- Управление аэрокосмическими системами (УАКС-2021)
- Управление в перспективных наземных транспортных системах (УПНТС-2021)

Подробности: **https://niimvus.org.ru/**

УДК 004.82 DOI: 10.17587/prin.12.189-199

О. Н. Долинина, д-р техн. наук, директор института, odolinina09@gmail.com,
Саратовский государственный технический университет,
В. А. Кушников, д-р техн. наук, директор, kushnikoff@yandex.ru,
Саратовский научный центр РАН

Методы и технологии обеспечения качества интеллектуальных систем принятия решения

Повышение степени интеллектуализации способов решения задач требует создания методологии повышения качества интеллектуальных систем принятия решения (ИСПР). В статье приведен подробный обзор существующих методов и технологий обеспечения качества ИСПР, описана методология контроля качества баз знаний ИСПР, дан сравнительный анализ методов статического и динамического анализа баз знаний. Дан анализ российских и отечественных работ, посвященных классификации ошибок в базах знаний ИСПР, их отладке. Описан метод генерации тестов, позволяющий обнаруживать класс ошибок "забывание об исключении" в продукционных и нейросетевых базах знаний, основанных на трехслойном персептроне, на основе метода PODEM.

Ключевые слова: *качество интеллектуальных систем, план обеспечения качества, гарантоспособность программы, ошибки баз знаний, ошибка типа "забывание об исключении", отладка баз знаний, противоречивость знаний, тестирование*

Введение

Последнее десятилетие характеризуется значительным развитием интеллектуальных технологий во всех сферах человеческой деятельности. Возможность автоматизации принятия решения в слабо формализуемых областях за счет использования экспертных знаний приводит к возрастанию числа ошибок в программном обеспечении, и как следствие — к возрастанию числа всевозможных источников отказов. Повышение степени интеллектуализации задач, увеличивающаяся сложность управления современными системами в промышленности, экономике, социальной сфере, их взаимная интеграция, увеличение числа слабо формализуемых задач, повышающиеся требования к качеству принимаемых решений требуют создания методологии повышения качества интеллектуальных систем принятия решений (ИСПР). Особое внимание уделяется системам, имеющим повышенные требования к надежности.

Необходимо отметить, что в настоящее время четко наметились два основных направления развития ИСПР: системы, основанные на знаниях, использующие формализованные модели представления экспертных знаний, и системы, основанные на механизме искусственных нейронных сетей (ИНС). Более того, поскольку процесс формирования баз знаний в их традиционном понимании специалистами по искусственному интеллекту требует больших временных и материальных затрат, то наблюдается тенденция замены экспертных знаний принятием решения искусственными нейронными сетями. Для решения достаточно большого числа задач ИНС себя достаточно хорошо зарекомендовали, а поскольку формирование обучающего множества нейронной сети не требует длительной и дорогостоящей работы с экспертами, то появилась тенденция сокращения разработок в области формализации знаний за счет использования ИНС.

Методы обеспечения качества ИСПР

Качество ИСПР является комплексным многокритериальным показателем, учитывающим не только качество работы каждой подсистемы, но и причинно-следственные взаимодействия элементов системы. Сложность решения задачи обеспечения качества ИСПР связана в первую очередь с наличием у ИСПР таких свойств, как противоречивость знаний, эмергентность, активность, неидентичность и др.

При рассмотрении проблемы обеспечения качества ИСПР в данной статье, с одной стороны, рассмотрим интеллектуальную систему (ИС) как программную систему со всеми свойствами, присущими данному классу систем, с другой стороны, рассмотрим специализированные методы обеспечения качества базы знаний, основного компонента ИСПР.

В теории качества программного обеспечения (ПО) существуют прямо противоположные подходы: от формального подхода, описанного в литературе [1—3] и оформленного в международных и отечественных стандартах ГОСТ Р ИСО/МЭК 9126—93. ISO/IEC 25010:2011, ГОСТ Р ИСО/МЭК 25010—2015 до так называемого функционального [4—8], основанного на том, что функциональные возможности

и степень удовлетворенности пользователей программного продукта важнее структурных характеристик при определении качества ПО, что "качество программного продукта является показателем того, насколько он меняет мир к лучшему" [4]; понятия качества ПО как субъективного по своей природе, поскольку зависит от людей, оценивающих качество [8]. Отметим, что до настоящего времени методы обеспечения качества базы знаний, являющейся центральной частью ИС, недостаточно формализованы, не существует единого подхода к решению данной проблемы. Вместе с тем процесс интеллектуализации систем управления, поддержки принятия решений сопровождается существенным увеличением объемов баз знаний, что требует создания эффективных методов контроля качества этой важнейшей составляющей ИСПР.

Развитие теории качества применительно к современным ИС, особенно для решения управленческих задач большой размерности, происходит недостаточно эффективно. Основными причинами является то, что модели и методы классической теории качества не могут описать и дать адекватные оценки объектам, работоспособность которых нарушается не только вследствие отказов физической природы, но и вследствие ошибок проектирования, информационных воздействий, ошибок в базах знаний и др. Для таких объектов трудно определить само понятие отказа и определить множество причин, порождающих его.

В общем случае можно утверждать, что качество является управляемым показателем и может быть представлено в виде древовидного ациклического связного графа G, где по стандарту ГОСТ Р ИСО/МЭК 25010—2015 верхний уровень представлен следующими характеристиками: функциональность, уровень производительности, совместимость, удобство пользования, надежность, защищенность, сопровождаемость, переносимость (мобильность). Необходимо отметить, что каждая из предложенных характеристик может быть детализирована соответствующими метриками, например, по стандарту ISO/IEC 9126. Детализация или уточнение показателей качества зачастую определяется отраслевыми требованиями, требованиями компании или конкретного программного проекта. Необходимо отметить, что в современной литературе имеются разночтения в содержательных определениях характеристик качества. Весьма показательным примером может являться такой показатель, как *dependabilty*, определяемый ГОСТ как надежность, в то время как зарубежные источники, например, работа [1], определяют этот термин как гарантоспособность [9—15], в которую надежность входит как составная часть. В стандартах ITU-T E.800 "Quality of telecommunication services" (качество телекоммуникационных услуг) и IEC 60300-1:2014 "Dependability management" (управление надежностью) гарантоспособность по составу у́же, чем надежность. В стандарте ГОСТ 27.002—89 не выделяется понятие гарантоспособности, а термин "надежность" имеет два англоязычных эквивалента — *reliability* и *dependability*. Согласно стандартам IAEA, ECSS такие свойства, как конфиденциальность и целостность рассматриваются как смежные для гарантоспособности, доступности, безотказности, достоверности, сопровождаемости, конфиденциальности, функциональности, безопасности. Базовый принцип гарантоспособных вычислений (*dependable computing*), сформулированный в работе [13], определяет их как вычисления, устойчивые к отказам аппаратных средств и программных средств, т. е. к отказам, обусловленным проявлением дефектов, внесенных при разработке и не выявленных при отладке. Еще одно ключевое понятие гарантоспособных программных систем — безопасная отказоустойчивость (*secure fault tolerance*) введено в работе [14] и определяется как средство, поддерживающее другие свойства гарантоспособности. Данный подход к трактовке отказоустойчивости обоснован, так как с помощью рассмотренных средств может обеспечиваться и безотказность, и готовность, и безопасность, и живучесть. В этой же работе предложен принцип его реализации для различных программных систем. Таким образом, гарантоспособность программных систем принятия решения представляет собой комплексный критерий, определяющий свойство осуществлять требуемые услуги, которым можно оправданно доверять [13]. Комплексный критерий гарантоспособности состоит из трех основных компонентов: угрозы; атрибуты; средства. Структура критерия гарантоспособности приведена на рис. 1.

Среди работ, посвященных развитию методов обеспечения качества ПО, следует отметить работы, А. И. Лозинского [3], Э. М. Кларка, О. Грамберга, Д. Пеледа [15], П. Йоргенсена [16], М. А. ван дер Линдена [17], А. Маркова [18], Г. Суринараяна [19], Б. Боэма и др. [20], где рассматриваются методы и способы обеспечения отдельных показателей качества.

Наиболее развитыми являются методы тестирования, описанные в работах Г. Майерса [21], В. В. Липаева [22, 23], К. Канера, Д. Фолка, Е. Нгуена [24], Л. Криспин, Д. Грегори [25], Б. Бейзера [26], Р. Калбертсона и др. [27], позволяющие обнаружить ошибки в программном коде. Однако решение проблемы управления качеством ПО как комплексного критерия до сих пор существует исключительно на неформальном уровне.

В общем случае комплексный показатель качества ПО для сложных систем принятия решения содержит большое число как количественных, так и качественных характеристик. Отметим, что разработка ИСПР в настоящее время часто осуществляется специалистами иной инженерной философии, базирующейся, по-прежнему, на подходах к разработке и анализу результатов, скорее, как к искусству, а не как к строгой науке. При этом сохраняется ориентация на качественные, а не на количественные инструментарии оценки, не укладывающиеся в математический аппарат строгих формальных методов.

С точки зрения системного анализа задача достижения требуемого уровня качества ПО в работе [28] представлена как задача оптимизации комплексного критерия $K = f(P_Q(\mathbf{x}(t), \mathbf{u}(t)))$, где $\mathbf{u}(t) \in \{\mathbf{U}(t)\}$ —

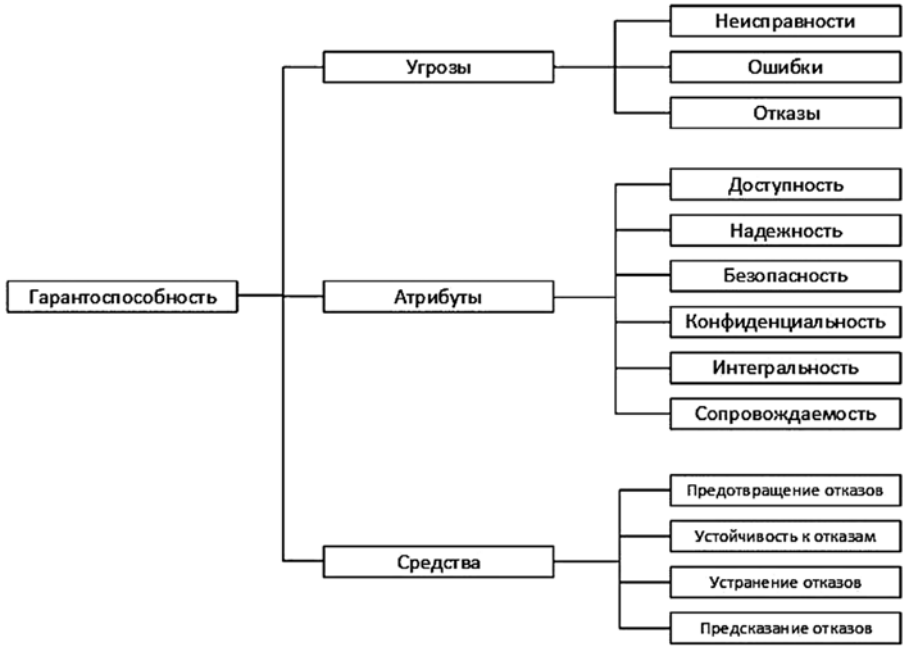


Рис. 1. Структура критерия гарантоспособности программных систем принятия решения

вектор управляющих воздействий; $\mathbf{x}(t) \in \{\mathbf{X}(t)\}$ — вектор состояний окружающей среды.

Критерием эффективности решаемой задачи является функция $P_Q(\mathbf{x}(t), \mathbf{u}(t))$, характеризующая вероятность поддержания на заданном интервале времени требуемого уровня качества ПО при неблагоприятном стечении обстоятельств. Под неблагоприятным стечением обстоятельств в данном случае понимается такое сочетание событий в процессе функционирования программной системы, каждое из которых по-отдельности не оказывает значительного влияния на качество ПО, но в совокупности приводит к значительному снижению его качества.

Учитывая неопределенность ряда скалярных составляющих вектора состояний окружающей среды на временных интервалах значительной длины, наличие в составе векторов **X** и **U** как количественных, так и качественных переменных, имеющих в том числе и нечеткий характер, в основу предлагаемого решения синтеза вектора управляющих воздействий $\mathbf{u}(t)^*$ положено неоднократно проверенное на практике утверждение. Согласно такому утверждению для решения задачи максимизации критерия K достаточно разработать и реализовать подробный комплексный план мероприятий $Pl(t)$ по повышению качества функционирования ПО и использовать для этого эвристическую процедуру $Pl(t)$, основанную на требованиях стандартов ГОСТ Р ИСО/МЭК 9126—93 и ГОСТ Р ИСО/МЭК 25010—2015.

Для определения прогнозного значения основных показателей качества ИСПР, непосредственно влияющих на эксплуатационные характеристики ее ПО, нужно учитывать, что между большинством моделируемых характеристик качества существуют структурно сложные прямые и обратные связи. Такие связи значительно влияют на поведение данных характеристик во времени. Перечисленные факторы существенно затрудняют разработку адекватной математической модели; в качестве формального аппарата, описывающего изменение показателей качества, в работе [28] предлагается математический аппарат системной динамики [29]. Отметим, что для решения поставленной задачи необходимо определить множество наиболее значимых внешних факторов, влияющих на моделируемые переменные, например, опыт разработчиков программного комплекса, трудоемкость разработки ПО, деловая репутация разработчиков программного комплекса и др.

Поскольку ИС являются разновидностью ПО, то для них характерны те же критерии качества, что и для любых программных систем. Однако принимая во внимание, что ядром ИСПР является база знаний, в данной статье остановимся на методах обеспечения качества именно баз знаний.

Модели ошибок в базах знаний и методы отладки интеллектуальных систем

Методы извлечения и формализации знаний в настоящее время довольно хорошо разработаны, однако методология отладки интеллектуальных систем принятия решений, основанных на знаниях, до сих пор остается недостаточно формализованной, базирующейся в основном на экспертном подходе, требующем больших временных и финансовых затрат, но не гарантирующим отсутствие ошибок в базах знаний. Более того, отсутствие методологии отладки баз знаний привело в конце 1990-х годов к тенденции снижения использования ИСПР за счет мнения ряда специалистов по искусственному интеллекту о том, что этапы создания ИСПР не подчиняются требованиям программной индустрии, ориентированной

на жизненный цикл и стандарты. В настоящее время наблюдается обратный процесс — повышение интереса к ИС и даже наделение их человеческими свойствами.

Завершающим этапом разработки ИС является этап отладки. Целью процесса отладки баз знаний (БЗ) является обнаружение и устранение максимального числа ошибок. В данной статье будем использовать понятие отладки БЗ в узком смысле [30].

Отладкой БЗ в узком смысле называется процесс обнаружения, локализации и устранения ошибок в БЗ и соответствующей коррекции БЗ, не связанной с выбором нового способа представления знаний.

Ошибки БЗ можно разделить на следующие группы:

- ошибки, связанные с нарушением внутренней структуры БЗ (структурные ошибки);
- ошибки, связанные с внешними противоречиями предметной области.

Для устранения противоречий в терминологии в области отладки БЗ ИСПР разобьем семейство методов на следующие классы:

- статические методы (не требующие запуска ИСПР на выполнение);
- динамические методы.

Несмотря на рост числа гибридных моделей знаний, в современных ИС наиболее распространенным способом представления БЗ являются продукции. Продукционную базу знаний (ПБЗ) можно представить в следующем виде:

$$P = (F,\ R,\ G,\ C,\ I),$$

где F — множество фактов о решаемой проблеме; R — множество правил вида

$$r_m\text{: если } f_i \text{ и } f_j \text{ ... и } f_n \text{ то } f_k; \qquad (1)$$

G — множество целей; C — множество разрешенных комбинаций фактов; I — интерпретатор правил, реализующий вывод.

Пусть S — множество входных фактов, т. е. фактов, устанавливаемых пользователем в ИС; $S \subset F$. База знаний может быть представлена в виде И/ИЛИ-графа. Например, пусть база знаний содержит следующие правила:

r_1: если s_1 и s_2, то f_1; r_2: если s_2 и s_3, то f_2; r_3: если s_3 и s_4 и s_5, то f_3;

r_4: если f_1, то g_1; r_5: если f_2 и f_3, то g_2.

Тогда приведенным правилам соответствует И/ИЛИ-граф на рис. 2.

Структурной ошибкой e_i в ПБЗ называется ошибка, выявляемая в ходе анализа эквивалентного базе знаний И/ИЛИ-графа, $e_i \in \{E\}$, где множество E — множество всех структурных ошибок в ПБЗ [31—35]. Классификация структурных ошибок приведена на рис. 3.

Приведенные классы структурных ошибок БЗ формализованы и могут быть обнаружены методами статического анализа, к которым относятся анализ таблиц решений, попарное сравнение правил, разбиение правил на смысловые группы, методы теории графов. Наиболее развиты и хорошо представлены в литературе методы, основанные на анализе графовой структуры БЗ. Отметим уже классические работы Х. Сувы, А. Скотта и др. [31], Т. Нгуэна, В. Перкинса и др. [32], Б. Крагуна, Х. Стендела [33], современные работы О. Н. Долининой и др. [34], Р. Кнауфа и др. [35], С. Миллера и др. [36], Е. Пира и др. [37],

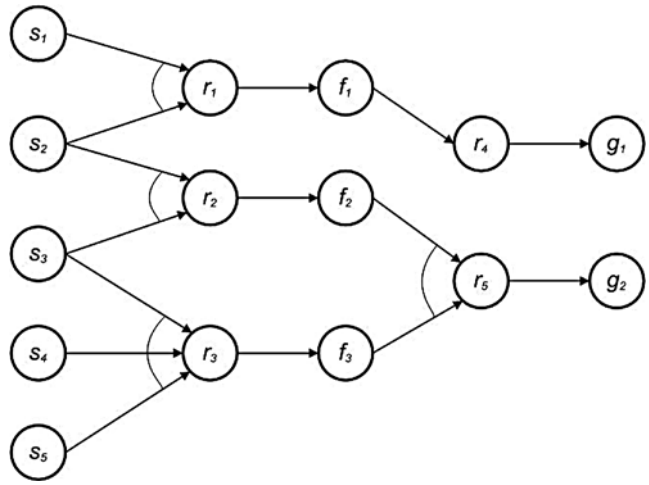


Рис. 2. Пример И/ИЛИ-графа базы знаний

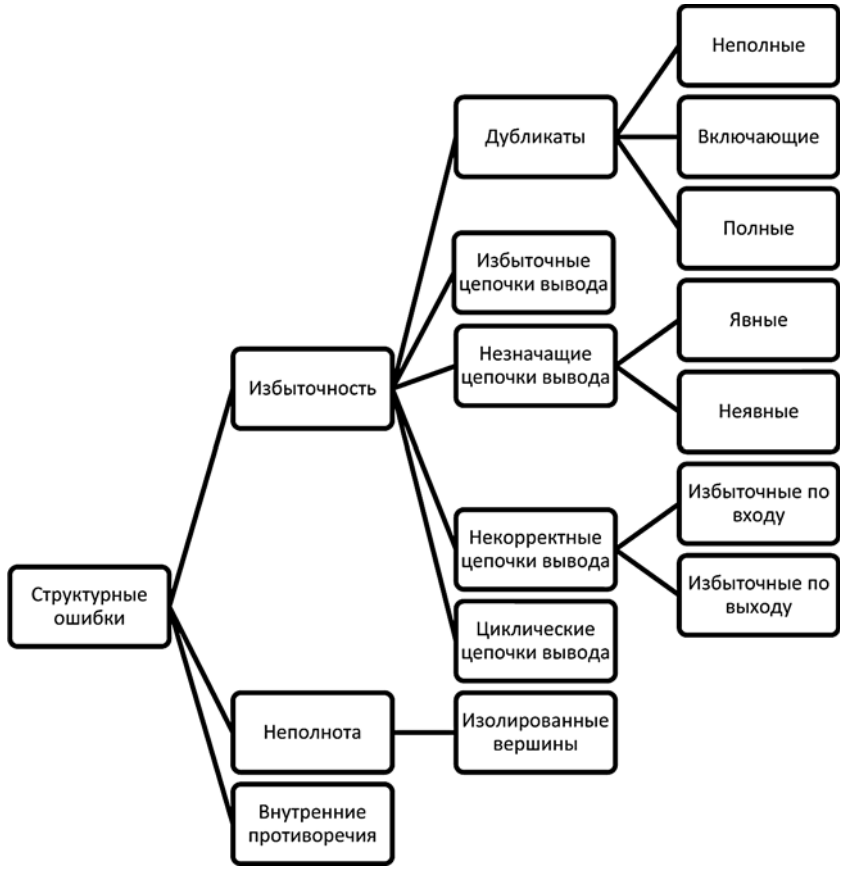


Рис. 3. Классификация структурных ошибок продукционных баз знаний

Сравнительная характеристика наиболее распространенных методов статического анализа продукционных баз знаний

Классы ошибок ПБЗ	Методы				
	Попарное сравнение правил	Логика и доказательство теорем	Таблицы решений	Исследование графов	Сети Петри
Избыточность	+	+	+	+	+ Кроме циклических цепочек вывода
Неполнота	–	–	+	+	+
Внутренняя противоречивость	+	+	+	+	+
Внешняя противоречивость	–	–	–	–	–
Примечание: "+" означает выявление указанного класса ошибок; "–" — невыявление.					

Д. Ксу и др. [38], А. Киматти и др. [39], а также работу К. Ву, Ч. Жу и др., использующую аппарат сетей Петри [40]. Сравнительная характеристика методов статического анализа ПБЗ представлена в таблице.

База знаний, в которой отсутствуют структурные ошибки, считается статически корректной [30]. Приведение БЗ в состояние статической корректности является необходимым, но недостаточным условием отладки. Статическая корректность БЗ не гарантирует ошибок иного рода, причиной которых является противоречивость самой предметной области (так называемая внешняя противоречивость), а также ошибки в самих правилах.

Формально понятие неполноты может быть определено с помощью так называемой сильной теоремы Геделя [41] о неполноте (логическая полнота (или неполнота) любой системы аксиом не может быть доказана в рамках этой системы, для ее доказательства или опровержения требуются дополнительные аксиомы (усиление системы)). С точки зрения ПБЗ неполнота может означать невозможность вывода какого-либо факта в исследуемой БЗ. В работе [42], например, предлагается рассматривать проблему неполноты более узко в разрезе неполных знаний.

В системе формальной логики набор определенных в системе аксиом (представляющих знания) полон, и используется свойство монотонности, заключающееся в том, что вывод остается правильным при добавлении каждой новой аксиомы, соответствующей правилам формальной системы. Однако свойство монотонности в большинстве случаев несправедливо для БЗ. Поэтому решение проблемы немонотонных выводов имеет важное значение. Во время разработки БЗ часто применяют гипотезу "закрытого мира", согласно которой заведомо ложными считаются все неопределенные в данной базе знания, поэтому при небольших размерах БЗ в системе выводимо достаточно малое число фактов (целевых утверждений) и такую БЗ можно считать неполной. Отметим, что решение проблемы устранения неполноты знаний в данной статье не рассматривается.

Будем считать, что ПБЗ P является внешне противоречивой в узком смысле слова, если существует множество допустимых сочетаний фактов, ведущих к неверному решению в предметной области. Большинство предметных областей является внешне противоречивыми и содержит ошибки, связанные с исключениями в предметной области или с критическим сочетанием событий, повлекшим ошибки в принятии решения. Исследование противоречивости предметной области является актуальной темой современных исследований. Если все факты, которые использует ИСПР, могут быть верны одновременно, то никаких противоречий при принятии решения не возникнет. Противоречия формируются в силу наличия содержательных связей между фактами, запрещающих определенные связи. В некоторых предметных областях существуют несовместимые системы взглядов, причем эта несовместимость может проявляться не сразу. В работах Д. А. Поспелова, Г. С. Поспелова, И. Г. Поспелова [43—45], А. С. Нариньяни [46, 47], О. Н. Долининой [48] содержательно показана возможность неверного решения, принимаемого экспертной системой по причине противоречий самой предметной области. Рассмотрим классический пример внешней противоречивости, где следующие правила (2) являются вполне корректными, пока в качестве рассматриваемого объекта не будет введена "летучая мышь" и ПС (2) для летучей мыши делает вывод "птица", который противоречит естественному отношению между фактами "птица — это не зверь", без которого рассматриваемая классификация вообще бессмысленна:

$$\begin{gathered} r_1\text{: ЕСЛИ животное теплокровное И имеет крылья, ТО птица} \\ r_2\text{: ЕСЛИ животное теплокровное И имеет 4 лапы, ТО зверь} \end{gathered} \qquad (2)$$

Этот пример характеризует комбинацию фактов предметной области, связанную с забыванием об исключении. Приведем сформулированную авторами в работе [48] формальную модель ошибки "забывание об исключении" для ПБЗ.

Ошибка типа "забывание об исключении" в ПБЗ имеет место, если в правиле r_m: если f_i и f_j ... и f_n то f_y выполняется всегда, за исключением того случая, когда в ПБЗ установлен набор фактов

$$\{f_1, f_2, ..., f_k\} = S \in C. \quad (3)$$

Число фактов в исключительной комбинации называется степенью ошибки типа "забывание об исключении". Ошибка вида (3) является наиболее общей и покрывает другие виды ошибок в статически корректной ПБЗ.

Субъективность экспертных знаний, сложность их формализации, учета семантических связей привели также к выделению класса ошибок, называемого НЕ-факторами, введенному А. С. Нариньяни в работах [46, 47]. К НЕ-факторам принято относить такие ошибки, как неточность, неопределенность, неоднозначность.

Неточность представляет собой величину, которая может быть получена с точностью, не превышающей некоторый порог α, определенный природой соответствующего параметра. Отметим, что в настоящее время понятие неточности наименее изучено и не имеет формального определения.

Недоопределенность является интервальной оценкой величины, которая по своей природе является более точной, чем в данный момент позволяет установить доступная для разработки информация.

Неоднозначность представляет собой множество альтернатив, оцениваемых неравномерно с точки зрения определенной семантики, например, уверенность, возможность, желательность и т. п. Неоднозначное значение включает недоопределенный интервал альтернатив и заданное на нем распределение, отражает оценку каждого значения интервала представляемым переменной смыслом. Неоднозначность является более сложным фактором, чем используемая в нем недоопределенность. При этом сужение интервала приводит к корректировке распределения. Очевидно, что изменения интервала и распределения зависят от поступления новой информации, дополняющей знания о представляемом параметре, правила изменения определяются семантикой определяемой оценки. Нечеткие модели, предложенные Л. Заде [49], могут считаться разновидностью неоднозначности.

Отметим, что класс ошибок "НЕ-факторы" обусловлен самой природой ИС, т. е. использованием для принятия решений экспертных знаний, в том числе субъективных, а также сложностью и противоречивостью самой предметной области. В настоящее время не существует единства в определении НЕ-факторов в научной литературе.

Поскольку содержательные ошибки не могут быть обнаружены графовыми или иными методами статического анализа, так как не связаны со структурой знаний, то они могут быть выявлены только путем тестирования БЗ. Тестирование предполагает запуск интерпретатора ИСПР на выполнение на тестовом множестве, представляющим определенное множество входных данных, и сравнение полученных результатов вывода с эталонными, сформированными экспертом. Эффективность тестирования зависит от разработанного тестового множества. Процедуру тестирования можно проводить на той стадии разработки, как только ИСПР может выполнить несколько контрольных примеров от начала до конца. Последовательность отладки ИСПР приведена на рис. 4.

Статически корректную базу знаний, не содержащую ошибок типа "забывание об исключении", будем называть корректной базой знаний.

Существующий опыт тестирования традиционного ПО позволяет лишь частично использовать методы тестирования программ для тестирования БЗ. Тестирование начинается с предположения, что система содержит ошибки, а затем уже диагностируется их максимальное число. В настоящее время чаще всего для тестирования БЗ продолжает использоваться стратегия "черного ящика", когда содержимое БЗ неизвестно. Экспертным путем составляется множество тестов *T*. Результаты выполнения ИСПР на множестве *T* сравниваются с эталонными, определенными экспертным способом, в результате делается вывод о факте наличия ошибок в БЗ. После принятия решения о корректности или некорректности БЗ для оценки разброса между мнениями экспертов и результатами вывода ИСПР, а также для выявления противоречивости мнений самих экспертов могут использоваться различные статистические методы.

Критерием исчерпывающего входного тестирования БЗ является обнаружение всех ошибок. Если в качестве тестовых наборов использовать полный перебор всех возможных наборов входных данных, то тестирование может удовлетворить данному кри-

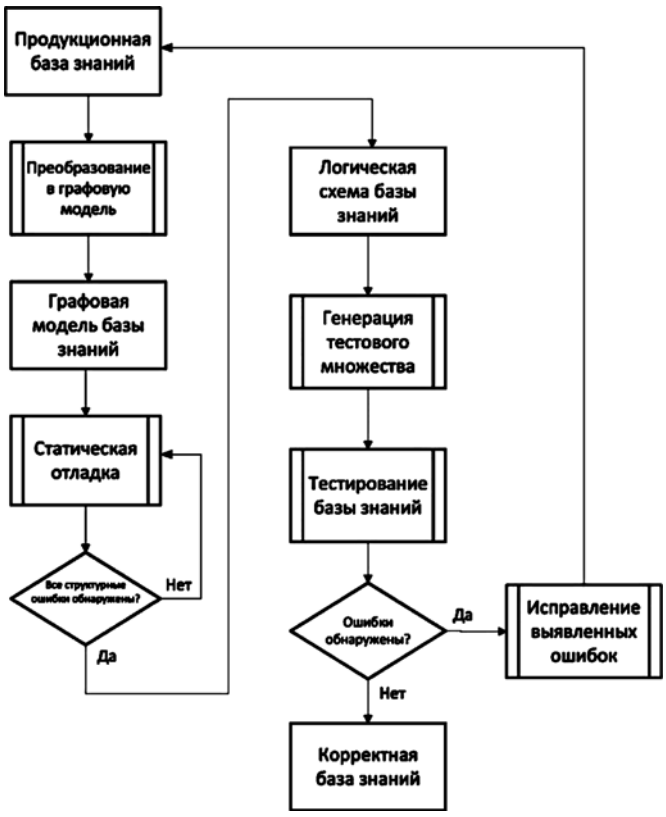


Рис. 4. Последовательность этапов отладки продукционной базы знаний

терию. Однако использование таких тестов противоречит экономическим, физическим, психологическим требованиям, так как связано с привлечением экспертов к построению исчерпывающего входного теста. Для лучшего тестирования БЗ предполагается использовать модульное тестирование, подразумевающее разбиение ИСПР на подсистемы, для того чтобы проследить принятие решения каждой подсистемой на определенном экспертом/экспертами тестовом наборе. Обычно целью модульного тестирования является упрощение процесса оценивания программного обеспечения ИСПР целиком, но достоверность каждой подсистемы ИСПР не может являться достаточным условием достоверности всей ИСПР, поскольку в случае принятия гипотезы о достоверности наличие незначительных ошибок в отдельных подсистемах при их накоплении может привести к невыполнимости гипотезы о достоверности в целом всей системы.

В литературе описаны попытки распространить основные критерии и методы тестирования программ для тестирования БЗ, т. е. конкретных продукций, составляющих ее ядро [50—52]. Отметим, что часто используемый критерий покрытия операторов (КПО), подразумевающий выполнение каждого оператора программы хотя бы один раз, в случае БЗ требующий выполнения каждой продукции, входящей в БЗ, является необходимым, но недостаточным условием для приемлемого тестирования. В качестве более сильного критерия можно использовать критерий покрытия решений, требующий, чтобы каждое решение имело результатом "истину" или "ложь", и при этом каждый оператор выполнялся по крайней мере один раз.

Используемый часто критерий комбинаторного покрытия решений всех решений и путей в программе в большинстве случаев очень трудоемок как для обычных программ, так и для ИСПР. Отметим, что требования к критериям тестирования справедливы для БЗ, содержащих детерминированные продукции. Для БЗ, содержащих недетерминированные (вероятностные, неточные или нечеткие) продукции, принятие решения обычно осуществляется на основе аккумуляции всей используемой информации. Например, продукция может выполняться на заданных тестовых данных, но ее вклад в процесс принятия решения, возможно, будет минимальным, и, следовательно, она практически не влияет на принятие решения. В этом случае целесообразно рассмотреть достижение каждой цели в БЗ, и, следовательно, КПО является явно недостаточным критерием. Наиболее явная ошибка в продукции вида (1) имеет место, если факт f_i никогда не бывает истинным, если установлены факты $f_1, f_2, ..., f_k$.

Тесты, полученные по КПО, обнаруживают именно ошибки такого рода. В случае, если правило может выполняться или не выполняться в зависимости от состояния некоторых других фактов системы, то обнаружение подобных ошибок не гарантируется тестами КПО. В качестве примера подобной ситуации приведем рассмотренный выше пример ошибки типа "забывание об исключении" (3). Очевидно, что решение данного вопроса связано с более детальным изучением предметной области, понимания ее внешней противоречивости и формированием тестового множества, которое необходимо, чтобы выявить противоречивость вида (3) в БЗ.

В работе [52] показано, что логика ПБЗ, представляемая И/ИЛИ-графом Г, после выполнения статического анализа и приведения базы знаний к состоянию статической корректности может быть задана соответствующей графу Г связной логической схемой (ЛС). Переход к схемотехническому представлению ПБЗ дает возможность применить для построения тестов методы, используемые в технической диагностике цифровых устройств.

В диагностике цифровых устройств одной из моделей неисправностей является константная неисправность (см., например, [53]). Ошибка типа (3) считается неисправностью "константный ноль", проявляющейся только при одном наборе значений сигналов в логической сети. Особенностью данного подхода является задание экспертами множества запрещенных комбинаций входных фактов, что психологически проще, чем формирование множества разрешенных комбинаций. В работах авторов [48, 52] показано, что для генерации тестов ПБЗ наиболее целесообразно применить алгоритм технической диагностики PODEM [54], гарантирующий 100%-ное покрытие неисправностей. Неполный логический базис (отсутствие "не" в продукциях) позволяет определять в том числе и кратные ошибки, когда неверными могут быть сразу несколько правил.

Интеллектуальные системы, использующие для принятия решения механизм искусственных нейронных сетей (ИНС), выделим в отдельный класс. Сети ИНС используют в тех случаях, когда трудно или невозможно провести формализацию знаний по принятым моделям, но существует возможность сформировать обучающее множество, которое будет являться основой для принятия решения.

Важное отличие жизненного цикла нейросетевых экспертных систем от жизненного цикла систем, использующих классические модели представления экспертной информации, состоит в том, что этап выполнения для классических моделей, заключающийся в наполнении базы знаний экспертным способом, заменяется автоматическим процессом обучения для нейросетевого механизма принятия решений и обработки экспертной информации. В течение процесса обучения экспертная информация предметной области формализуется в виде множества весовых коэффициентов связей между нейронами, таким образом, обучение является информационным процессом, связанным с обнаружением закономерностей в данных. Однако полученное в ходе обучения представление экспертной информации носит неявный характер, его интерпретация и описание на естественном языке крайне затруднительно и представляет сложность для последующего анализа. В работах Д. Родволда [55, 56] была предложена следующая последовательность этапов жизненного цикла нейросетевых ИСПР, которую на настоящее время можно считать стандартом "де факто".

1. Идентификация: определение требований к системе, целей и ограничений, составляющих спецификацию разработки нейросети.

2. Формализация: сбор данных, которые будут использоваться для обучения нейросети, формиро-

вание исходного формата данных и составляющих документа по анализу данных.

3. Обучение нейросети как итерационный процесс.

4. Выполнение, предусматривающее развертывание и принятие решения нейросетью. Отладка: независимое тестирование и верификация нейросети.

Нужно сказать, что до сих пор отладка большинства ИНС происходит практически вручную путем подбора архитектуры ИНС (в том числе числа входных нейронов, слоев, эпох обучения и т. д.), что не гарантирует корректное решение задачи для всех входных данных, в том числе для описанного выше класса "забывание об исключении". Очевидно, что использовать методы структурного тестирования для ИНС, фактически представляющую "черный ящик" для разработчика и тестировщика, невозможно без преобразования ИНС. В работе [57] для трехслойного персептрона, решающего задачу классификации, предлагается использовать прореживание нейронной сети с получением множества решающих правил, а затем провести тестирование при помощи метода, использующего алгоритм PODEM, описанного выше для ПБЗ. Отметим, однако, что поскольку основной целью прореживания является определение наиболее значимых связей, то возможно, что в результирующем множестве будут потеряны данные, которые ведут к ошибкам типа (3). В работах [58, 59] предлагается использовать четыре взаимодополняемых критерия к формированию множества тестов для сетей глубокого обучения на основе подхода MC/DC (модифицированное условие/покрытие решения — *modified condition/decision coverage*), являющегося вариантом конколического тестирования (гибридной методики верификации ПО, которая обрабатывает программные переменные вдоль конкретного пути выполнения). Для каждого критерия предложен алгоритм генерации тестов-кейсов на основе линейного программирования. Тесты строятся для всех нейронов активации, фиксируются полученные паттерны активации. Алгоритмы генерации тестов были апробированы на наборе данных MNIST (образцов рукописного написания цифр).

В статье [60] рассмотрены вопросы проверки адекватности сверточной нейронной сети, осуществляемой с использованием процедуры мутационного тестирования. Из анализа специальной литературы известны трудности, связанные с применением традиционных критериев адекватности тестирования при измерении степени адекватности сверточных нейросетевых приложений. Только небольшое число тестовых случаев, примененных к модели сверточных нейросетевых приложений, позволяет обеспечить охват нейронов почти на 100 %. В работе предлагается критерий покрытия, основанный на тестировании мутаций для сверточных нейросетевых приложений, и используется критерий покрытия для общих моделей классификации изображений (LeNet-5). Описанный метод позволяет построить локальную оптимальную модель, что не гарантирует обнаружения всех возможных ошибок, в том числе связанных с противоречивостью предметной области.

Необходимо также сказать, что в настоящее время активно развиваются исследования, направленные на использование ИНС для разработки новых подходов к отладке традиционных моделей БЗ, в том числе для тестирования методом черного ящика или поведенческого тестирования СБИС [61], функционального тестирования [62], формирования приоритетного направленного теста для функциональной проверки [63, 64].

Заключение

Качество ИСПР является комплексным многокритериальным понятием, процессы формализации которого до сих пор продолжаются. Наибольшую сложность представляет отладка БЗ, являющихся центральным звеном ИС. Наиболее развитыми и формализованными являются методы поиска структурных ошибок в ПБЗ. Интеллектуальные системы, основанные на искусственных нейронных сетях, несмотря на их растущую популярность тем не менее продолжают оставаться "черным ящиком" для процессов отладки, в особенности для тестирования. Наибольшую сложность для всех видов ИСПР представляет внешняя противоречивость предметной области. Именно она является источником наиболее трудно обнаруживаемых ошибок, в том числе ошибок типа "забывание об исключении", а также НЕ-факторов, выявляемых методами тестирования, которые, несмотря на исследования в данной области, продолжают оставаться не до конца формализованными и являются предметом дальнейших исследований.

Подготовка данной статьи поддержана проектом "Экспансия" РФФИ № 20-17-50028/20.

Список литературы

1. **Avizienis A., Laprie J.-C., Randell B.** Fundamental Concepts of Dependability // Proc. of the 3rd IEEE Information Survivability Workshop (ISW-2000), Boston, Massachusetts, USA, 2000. October 24—26, 2000. — P. 7—12.

2. **Бахмач Е. С., Герасименко А. Д., Головир В. А.** и др. Отказобезопасные информационно-управляющие системы на программируемой логике / под ред. В. С. Харченко, В. В. Скляра. Национальный аэрокосмический университет "ХАИ", Научно-производственное предприятие "Радий", 2008. — 380 с.

3. **Лозинин А. И., Шубинский И. Б.** Определение требований к программному обеспечению. URL: https://docplayer.ru/52636998-A-i-lozinin-i-b-shubinskiy.html (дата обращения 06.05.2016).

4. **DeMarco T. S.** Getting Past Burnout, Busywork and the Myth of Total Efficiency. — New York: Broadway Books, 2001. 248 p.

5. **Mandeville W. A.** Software costs of quality // IEEE Journal on Selected Areas in Communications. — 1990. — Vol. 8, No. 2. — P. 315—318.

6. **Galin D.** Towards an inclusive model for the cost of software quality // Software quality Professional. — 2004. — Vol. 6, No. 4. — P. 25—31.

7. **Knox S. T.** Modeling the costs of software quality // Digital Technical Journal. — 1993. — Vol. 5, No. 4. — P. 9—16.

8. **Weinberg G. M.** Quality Software Management. First-Order Measurement. — NY: Dorset House Publishing, 1993. — 108 p.

9. **ISO/IEC** 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. ГОСТ Р ИСО/МЭК 25010—2015 Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов: URL: https://iso.org (дата обращения 01.03.2021).

10. **Besnard D., Gacek C., Jones Cliff B.** Structure for Dependability: Computer Based Systems from Interdisciplinary Perspective. Springel Verlag London Limited, 2006. — 304 p.

11. **Computer** Safety, Reliability and Security / ed. by R. Winther, B. A. Gran, G. Dahll. Proc. 24th Intern. Conf. SAFECOMP 2005, Friedrikstadt, Norway, September 28—30, 2005. — 409 p.

12. **Laprie J. C.** Dependability Handbook. LAAS Report no 98-346. Toulouse: Laboratory for Dependability Engineering, 1998. — 365 p.

13. **Avizienis A., Laprie J.-C**. Dependable Computing From Concepts to Application // IEEE Trans. On Computers. — 1986. — Vol. 74, No. 5. — P. 629—638.

14. **Dobson I. E., Randell B.** Building Reliable Secure Computing Systems out of Unreliable Insecure Components // Proc. Of the IEEE Conf. on Security & Privacy, Oakland, USA, 1986. — P. 187—193.

15. **Кларк Э. М., Грамберг О., Пелед Д.** Верификация моделей программ: Model Checking. М.: МЦНМО, 2002. — 416 с.

16. **Jorgensen P. C.** Software Testing — A Craftsman's Approach. 3d Edition. Auerbach Pub., 2013. — 440 p.

17. **Maura A. van der Linden** Testing Code Security. Auerbach Pub., 2007. — 328 p.

18. **Markov A., Barabanov A., Tsirlov V.** Models for Testing Modifiable Systems // Probabilistic Modeling in System Engineering / ed. A. Kostogryzov. IntechOpen, 2018. — P. 147—168.

19. **Suryanarayana G.** Software Process versus Design Quality: Tug of War? // IEEE Software. — 2015. — Vol. 32, No. 4. — P. 7—11.

20. **Боэм Б., Браун Д., Каспар Х.** Характеристики качества программного обеспечения / пер. с англ. Е. К. Масловского. — М.: Мир, 1981. — 208 с.

21. **Майерс Г.** Качество программного обеспечения. — М.: Мир, 1980. — 360 с.

22. **Липаев В. В.** Надежность и функциональная безопасность комплексов программ реального времени. — М.: ИСП РАН, 2013. — 207 с.

23. **Липаев В. В.** Основные понятия, факторы и стандарты, определяющие качество крупномасштабных программных средств. — М.: Берлин: Директ-Медиа, 2015. — 237 с.

24. **Канер К., Фолк Д., Нгуен Е. К.** Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. — Киев: ДиаСофт, 2001. — 544 с.

25. **Криспин Л., Грегори Д.** Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд = Agile Testing: A Practical Guide for Testers and Agile Teams. М.: Вильямс, 2010. — 464 с.

26. **Бейзер Б.** Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. СПб.: Питер, 2004. — 320 с.

27. **Калбертсон Р., Браун К., Кобб Г.** Быстрое тестирование. — М.: Вильямс, 2002. — 374 с.

28. **Долинина О. Н.** Системный анализ, методы и модели построения интеллектуальных систем принятия решений при управлении сложными организационно-техническими комплексами: дис. ... д-ра техн. наук: 05.13.01. Саратов: СГТУ имени Гагарина Ю. А., 2018. — 563 с.

29. **Форрестер Д.** Мировая динамика. — М.: АСТ: СПб.: Terra Fantastica, 2003. — 379 с.

30. **Долинина О. Н.** Алгоритмы и методы разработки и отладки экспертных систем: монография. — Саратов: Саратовский гос. технический ун-т, 2015. — 225 с.

31. **Suwa H., Scott A. C., Shortliffe A.** An Approach to Veryfing Consistency and Completeness in a Rule-Based Expert System // Rule-Based Expert Systems. London: Addison—Wesley, 1984. — P. 159—170.

32. **Nguen T., Perkins W., Laffey T., Pecora W.** Checking Expert System Knowledge Bases for consistency and completeness // Proc. of the 9th Int. Joint Conf.on AI, Los.Ang, 1985. — P. 375—378.

33. **Cragun B. J., Stendel H. J.** A decision-table-based processor for checking completeness and consistency in rule-based expert systems // Int. J. Man- Mach. Stud. — 1987. — No. 5. — P. 633—648.

34. **Долинина О. Н., Резчиков А. Ф., Сучкова Н. К.** Формальные модели структурных ошибок в базах знаний интеллектуальных систем // Современные наукоемкие технологии. — 2017. — № 3 — С. 7—11. URL: http://www.top-technologies.ru/ru/article/view?id=36607 (дата обращения: 13.04.2017).

35. **Knauf R., Gonzalez A. J., Abel T.** A framework for validation of rule-based systems // IEEE Trans Syst Man Cybern B Cybern. — 2002. — Vol. 32, No. 3. — P. 281—295.

36. **Miller S. P., Whalen M. W., Cofer D. D.** Software model checking takes off // Commun. ACM. — 2010. — Vol. 53, No. 2. — P. 58—64.

37. **Pira E., Reza Zand Miralvand M., Soltani F.** Verification of Confliction and Unreachability in Rule-Based Expert Systems with Model Checking. URL: https://arxiv.org/abs/1404.2768 (дата обращения: 10.04.2014).

38. **Xu D., Kejian X., Zhang D., Zhang H.** Model Checking the Inconsistency and Circularity in Rule-Based Expert Systems // Computer and Information Science. — 2009. — Vol. 2, No. 1. — P. 12—17.

39. **Cimatti A., Corvino R., Lazzaro A., Narasamdya I., Rizzo T., Roveri M., Sanseviero A., Tchaltsev A.** Formal Verification and Validation of ERTMS Industrial Railway Train Spacing System / eds. P. Madhusudan, S. A. Seshia // CAV. — 2012. — Vol. 7358. — P. 378—393.

40. **Wu Q., Zhou C., Wu J., Wang C.** Study on Knowledge Base Verification Based on Petri Nets ICCA2005, June 27—29, 2005 Budapest, Hungery, IEEE, P. 997—1001.

41. **Godel K.** Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I. and On formally undecidable propositions of Principia Mathematica and related systems I in Solomon Feferman // Kurt Gödel Collected works. — Oxford University Press, 1986. — P. 144—195.

42. **Рыбина Г. В.** Модели, методы и программные средства для построения интегрированных экспертных систем: дис. ... д-ра техн. наук: 05.13.11. — М.: МИФИ, 2004. — 363 с.

43. **Поспелов Г. С., Поспелов Д. А.** Проблемы понимания рассуждений, основанных на знаниях // Материалы IV заседания РГ-22 САИИ. — Переяславль-Залесский, 1987. — С. 15—21.

44. **Поспелов И. Г., Поспелова Л. Я.** Динамическое описание систем продукций и проверка непротиворечивости продукционных экспертных систем // Известия АН СССР. Техническая кибернетика. — 1987. — № 1. — С. 184—192.

45. **Поспелов Д. А.** О "человеческих" рассуждениях в интеллектуальных системах // Вопросы кибернетики. Логика рассуждений и ее моделирование. — М., 1983. — С. 5—37.

46. **Нариньяни А. С.** НЕ-факторы и инженерия знаний: от наивной формализации к естественной прагматике // Сб. "Труды IV Национальной конференции "Искусственный интеллект-94". Т. 1. — Рыбинск, 1994. — С. 9—18.

47. **Нариньяни А. С.** НЕ-факторы State of Art // Научная сессия МИФИ-2004. — 2004. — Т. 3. — С. 26—30.

48. **Долинина О. Н.** Методы отладки баз знаний систем искусственного интеллекта // Системы управления и информационные технологии. — 2011. — № 2 (44). — С. 75—81.

49. **Zadeh L. A.** The Concept Of A Linguistic Variable And Its Application To Approximate Reasoning // Information Sciences. 1975. — Vol. 8. — P. 199—249, P. 301—357; — Vol. 9. — P. 43—80.

50. **Tepandi J.** Comparison of Expert System Verification Criteria: Redundancy // Proc. ECAI 90 Conf. Stockholm, 1990. — P. 49—62.

51. **Marcot B.** Testing your knowledge base // AI Expert, 1987. — Vol. 2. — P. 43—47.

52. **Долинина О. Н.** Информационные технологии в управлении современной организацией. — Саратов: Сарат. гос. техн. ун-т, 2006. — 160 с.

53. **Основы** технической диагностики: в 2-х книгах. Кн. 1. Модели объектов, методы и алгоритмы диагноза / под ред. П. П. Пархоменко. — М.: Энергия, 1976. — 464 с.

54. **Goel P., Rasales B. C.** PODEM-X: an automatic test generation system for VISI logic structures // Proc. 18th IEEE Design Automation Conf., 1981. — P. 260—268.

55. **Rodvold D. M.** A Software Development Process Model for Artificial Neural Networks in Critical Applications // Proc. of the 1999 Int. Conf. on Neural Networks (IJCNN'99). — Washington, 1999. — P. 3317—3322.

56. **Rodvold D. M.** Validation and Regulation of Medical Neural Networks // Molecular Urology. — 2011. — Vol. 5, No. 4. — P. 141—145.

57. **Долинина О. Н., Кузьмин А. К.** Метод генерации тестов для отладки нейросетевых экспертных систем // Вестник Тамбовского государственного технического университета. — 2010. — № 3. — С. 519—527.

58. **Sun Y., Huang X., Kroening D., Sharp J.** Structural Test Coverage Criteria for Deep Neural Networks // 2019 IEEE/ACM 41st Int. Conf. on Software Engineering: Companion Proc. (ICSE-Companion), 2019. — P. 320—321.

59. **Sun Y., Huang Z., Kroening D., Sharp J.** DeepConcolic: Testing and Debugging Deep Neural Networks // 2019 IEEE/ACM 41st Int. Conf. on Software Engineering: Companion Proc. — 2019. — P. 111—114.

60. **Yao Y., Liu J., Huang S., Hui Z., Wu Z.** Testing Adequacy of Convolutional Neural Network based on Mutation Testing // IEEE 19th Int. Conf. on Software Quality, Reliability and Security Companion (QRS-C). — 2019. — P. 536—537.

61. **Piuri V., Sami M. G., Sciuto D., Stefanelli R.** A behavioral approach to testability of neural networks // IEEE Xplore. — 1992. — P. II-654—II-659.

62. **Kirkland L. V., Glenn R.** Wright Functioning Testing Philosophies using Neural Networks // 1997 IEEE Xplore, 1997. — P. 88—91.

63. **Shem H., Fu Y.** Priority Directed Test Generation for Functional Verification using Neural Networks // IEEE 2005 ASP-DAC. — 2005. — P. 1052—1055.

64. **Fine S., Ziv A.** Coverage Directed Test Generation for Functional Verification // 40th Design Automation Conference, June 2—6. — 2003. — P. 286—291.

Methods and Technologies for Quality Assurance of Intelligent Decision-Making Systems

O. N. Dolinina, odolinina09@gmail.com, Yuri Gagarin State University of Saratov, Saratov, 410054, Russian Federation, **V. A. Kushnikov,** kushnikoff@yandex.ru, Saratov Scientific Center of Russian Academy of Sciences, Saratov, 410028, Russian Federation

Corresponding author:

Dolinina Olga N., PhD, Director of the Institute, Yuri Gagarin State University of Saratov, Saratov, 410054, Russian Federation
E-mail: odolinina09@gmail.com

Received on February 25, 2021
Accepted on March 25, 2021

An increase in the degree of intellectualization of tasks requires the creation of methodology for improving the quality of intelligent decision-making systems. The possibility of automating decision-making in poorly formalized areas through the using of the expert knowledge leads to increasing of the number of errors in the software, and as a consequence to increasing of the number of various sources of failures.The article provides a detailed overview of existing methods and technologies for quality assurance of intelligent decision systems. The first part of the article describes the methodology for ensuring the quality of the intelligent systems (IS), based on the GOST / ISO standards, where it is proposed to use a multilevel model to describe the quality of the IS software. It is shown that to ensure the required level of quality, an action plan can be formed and the use of a system dynamics model for the implementation of an action plan for ensuring the quality of IS is described. A comparative analysis of the complex criteria of quality and reliability is given. In the second part, the quality of knowledge base (KB) as a special element of the IS software is described, a comparative analysis of methods for static and dynamic analysis of knowledge bases is considered. An overview of research results in the classification of errors in the knowledge bases and their debugging is given. Special attention is given to the "forgetting about exception" type of errors. The concept of a statically correct knowledge base at the level of the knowledge structure is described and it is shown that statically correct knowledge bases can nevertheless give errors due to errors in the rules themselves because of the inconsistency of the field of studies. Neural network knowledge bases are allocated in a separate class, for neural networks methods of debugging are described.

Keywords: *quality of intelligent systems, quality assurance plan, reliability, knowledge base errors, "forgetting about an exception" error, debugging, inconsistency of knowledge, testing*

Acknowledgements:

This work was supported by the Russian Foundation for Basic Research, project nos. 20-17-50028/20

For citation:

Dolinina O. N., Kushnikov V. A. Methods and Technologies for Quality Assurance of Intelligent Decision-Making Systems, *Programmnaya Ingeneria,* 2021, vol. 12, no. 4, pp. 189—199.

DOI: 10.17587/prin.12.189-199

References

1. **Avizienis A., Randell B.** Fundamental Concepts of Dependability, *Proc. of the 3rd IEEE Information Survivability Workshop (ISW-2000)*, Boston, Massachusetts, USA, October 24—26, 2000, pp. 7—12.

2. **Bahmach E. S., Gerasimenko E. S.** Fail-safe information and control systems based on programmable logic / ed. by V. S. Harchenko, V. V. Sklyar National aerospace university, Research—production enterprise "Radium", 2008, 380 p. (in Russian).

3. **Losinin A. I., Shubinski I. B.** Definition of the demands to the software, available at: https://docplayer.ru/52636998-A-i-lozinin-i-b-shubinskiy.html

4. **DeMarco T. S.** *Getting Past Burnout, Busywork and the Myth of Total Efficiency,* New York, Broadway Books, 2001, 248 p.

5. **Mandeville W. A.** Software costs of quality, *IEEE Journal on Selected Areas in Communications,* 1990, vol. 8, no. 2, pp. 315—318.

6. **Galin D.** Towards an inclusive model for the cost of software quality, *Software quality Professional*, 2004, vol. 6, no. 4, pp. 25—31.

7. **Knox S. T.** Modeling the costs of software quality, *Digital Technical Journal*, 1993, vol. 5, no.4, pp. 9—16.

8. **Weinberg G. M.** *Quality Software Management, First-Order Measurement*. NY, Dorset House Publishing, 1993, 108 p.

9. **ISO/IEC 25010:2011** Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE), available at: https://www.iso.org

10. **Besnard D., Gacek C., Jones Cliff B.** *Structure for Dependability: Computer Based Systems from Interdisciplinary Perspective*, Springel Verlag London Limited, 2006, 304 p.

11. **Computer** Safety, Reliability and Security/ ed. by R. Winther, Biorn A. Gran, G. Dahll, *Proc. 24th Intern. Conf. SAFECOMP 2005*, Friedrikstadt, Norway, September 28—30, 2005, 409 p.

12. **Laprie J. C.** Dependability Handbook LAAS Report no 98-346, Toulouse, Laboratory for Dependability Engineering, 1998, 365 p.

13. **Avizienis A., Laprie J.-C.** Dependable Computing From Concepts to Application, *IEEE Trans. On Computers,* 1986, vol. 74, no. 5, pp. 629—638.

14. **Dobson I. E., Randell B.** Building Reliable Secure Computing Systems out of Unreliable Insecure Components, *Proc. of the IEEE Conf. on Security & Privacy,* Oakland, USA, 1986, pp. 187—193.

15. **Clarke E. M., Grumberg O., Peled D.** *Model Checking*, MIT Press, 1999, 314 p.

16. **Jorgensen P. C.** *Software Testing — A Craftsman's Approach.* 3d Edition, Auerbach Pub., 2013, 440 p.

17. **Maura A. van der Linden** *Testing Code Security.* Auerbach Pub., 2007, 328 p.

18. **Markov A., Barabanov A., Tsirlov V.** Models for Testing Modifiable Systems,: *Probabilistic Modeling in System Engineering* / ed. by A. Kostogryzov, IntechOpen, 2018, pp. 147—168.

19. **Suryanarayana G.** Software Process versus Design Quality: Tug of War? *IEEE Software,* 2015, vol. 32, no. 4, pp. 7—11.

20. **Boem B., Brown B., Kaspar H.** *Characteristics of the software quality,* Moscow, MIR, 1981. 208 p. (in Russian).

21. **Mayers G.** *Software Quality*, Moscow, MIR, 1980, 360 p. (in Russian).

22. **Lipaev V. V.** *Reliability and functional safety of real-time software complexes,* Moscow, ISP RAN, 2013, 207 p. (in Russian).

23. **Lipaev V. V.** *Basic concepts, factors and standards that determine the quality of large-scale software tools*, M.-Berlin, Direct-Media, 2015, 237 p. (in Russian).

24. **Kaner K., Folk D., Nguen E. K.** *Software Testing. Fundamental concept of the business-applications management*, Kiev, DiaSoft, 2001, 544 p. (in Russian).

25. **Krispin L., Gregory D.** *Agile Testing: A Practical Guide for Testers and Agile Teams,* Moscow, Wiliams, 2010, 464 p. (in Russian).

26. **Beizer B.** *Testing of the black box. Technology of the functional testing of the software and systems.* Saint-Petersburg, Piter, 2004, 320 p. (in Russian).

27. **Calbertson R., Brown K., Cobb G.** *Rapid Testing*, Moscow, Wiliams, 2002, 374 p.

28. **Dolinina O. N.** System analysis, methods and models for building intelligent decision-making systems in managing complex organizational and technical complexes, dis. Dr. Of Sc., Saratov, SSTU, 2018, 563 p. (in Russian).

29. **Forrester D.** *World Dynamics.* Moscow, AST, Saint-Petersburg, Terra Fantastica, 2003, 379 p. (in Russian).

30. **Dolinina O. N.** Algorithms and methods for developing and debugging expert systems; Saratov: SSTU, 2015, 225 p. (in Russian).

31. **Suwa H., Scott A. C., Shortliffe A.** An Approach to Veryfing Consistency and Completeness in a Rule-Based Expert System, *Rule-Based Expert Systems,* London, Addison—Wesley, 1984, pp. 159—170.

32. **Nguen T., Perkins W., Laffey T., Pecora W.** Checking Expert System Knowledge Bases for consistency and completeness, *Proc. of the 9th Int. Joint Conf. on AI,* Los-Ang., 1985, pp. 375—378.

33. **Cragun B. J., Stendel H. J.** A decision-table-based processor for checking completeness and consistency in rule-based expert systems, *Int. J. Man-Mach. Stud.* 1987, no. 5, pp. 633—648.

34. **Dolinina O. N., Rezchikov A. F., Suchkova N. K.** Formal models of structural errors in knowledge bases of intelligent systems, *Modern high technologies.* 2017, no. 3, pp. 7—11, available at http://www.top-technologies.ru/ru/article/ view?id=36607 (in Russian).

35. **Knauf R., Gonzalez A. J., Abel T. A.** framework for validation of rule-based systems, *IEEE Trans Syst Man Cybern B Cybern.*, 2002, vol. 32, no. 3, pp. 281—295.

36. **Miller S. P., Whalen M. W., Cofer D. D.** Software model checking takes off, *Commun. ACM,* 2010, vol. 53, no. 2, pp. 58—64.

37. **Pira E., Reza Zand Miralvand M., Soltani F.** Verification of Confliction and Unreachability in Rule-Based Expert Systems with Model Checking, available at https://arxiv.org/abs/1404.2768.

38. **Xu D., Kejian X., Zhang D., Zhang H.** Model Checking the Inconsistency and Circularity in Rule-Based Expert Systems, *Computer and Information Science,* 2009, vol. 2, no. 1, pp. 12—17.

39. **Cimatti A., Corvino R., Lazzaro A., Narasamdya I., Rizzo T., Roveri M., Sanseviero A., Tchaltsev A.** Formal Verification and Validation of ERTMS Industrial Railway Train Spacing System / eds. P. Madhusudan, S. A. Seshia, *CAV,* 2012, vol. 7358, pp. 378—393.

40. **Wu Q., Zhou C., Wu J., Wang C.** Study on Knowledge Base Verification Based on Petri Nets ICCA2005, June 27—29, 2005 Budapest, Hungery, *IEEE,* 2005, pp. 997—1001.

41. **Godel K.** Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. and On formally undecidable propositions of Principia Mathematica and related systems I in Solomon Feferman, *Kurt Gödel Collected works,* Oxford University Press, 1986, pp. 144—195.

42. **Rybina G. V.** *Models, methods and software for building integrated expert systems,* dis. Dr. of Sc., Moscow, MIFI, 2004, 363 p. (in Russian).

43. **Pospelov G. S., Pospelov D. A.** Challenges in understanding of knowledge reasoning, *Proc. IV Meeting, WG-22 AIS.*, Pereyaslavl-Zalesky, 1987, pp. 15—21.

44. **Pospelov I. G., Pospelova L. Ya.** Dynamic description of rule-based systems and consistency check of production expert systems, *News USSR Academy of Sc., Techn. Cybernetics*, 1987, no 1, pp. 184—192.

45. **Pospelov D. A.** *About "human" reasoning in intelligent systems, Problems of Cybernetics. Reasoning logic and its modeling*, Moscow, 1983, pp. 5—37.

46. **Nariniani A. S.** Non-factors and knowledge engineering: from naive formalization to natural pragmatics, *Proc. of the IV National Conference "Artificial Intelligence-94"*, vol. 1. Rybinsk, 1994, pp. 9—18 (in Russian).

47. **Nariniani A. S.** Non-factors State of Art, *Science session MIFI-2004*, 2004, vol. 3, pp. 26—30 (in Russian).

48. **Dolinina O. N.** Methods for debugging knowledge bases of artificial intelligence systems, *Control Systems and Information Technologies*, 2011, no. 2 (44), pp. 75—81 (in Russian).

49. **Zadeh L. A.** The Concept Of A Linguistic Variable And Its Application To Approximate Reasoning, *Information Sciences*, 1975, vol. 8, pp. 199—249, 301—357, vol. 9, pp. 43—80.

50. **Tepandi J.** Comparison of Expert System Verification Criteria: Redundancy, *J. Tepandi, Proc. ECAI 90 Conf.*, Stockholm, 1990, pp. 49—62.

51. **Marcot B.** Testing your knowledge base, *AI Expert*, 1987, vol. 2, pp. 43—47.

52. **Dolinina O. N.** *Information technologies in management of modern company,* Saratov, SSTU, 2006. 60 p. (in Russian).

53. **Fundamentals** of technical diagnostics: in 2 books. Book. 1. Models of objects, methods and algorithms of diagnosis/ ed. by P. P. Parkhomenko, Moscow, Energy, 1976, 464 p. (in Russian).

54. **Goel P., Rasales B. C**. PODEM-X: an automatic test generation system for VISI logic structures, *Proc. 18th IEEE Design Automation Conf.*, 1981, pp. 260—268.

55. **Rodvold D. M.** A Software Development Process Model for Artificial Neural Networks in Critical Applications, *Proc. of the 1999 Int. Conf. on Neural Networks (IJCNN'99)*, Washington, 1999, pp. 3317—3322.

56. **Rodvold D. M.** Validation and Regulation of Medical Neural Networks, *Molecular Urology*, 2011. vol. 5, no. 4, pp. 141—145.

57. **Dolinina O. N., Kuzmin A. K.** Test generation method for debugging neural network expert systems, *Vestnik of Tambov State Technical University*, 2010, no. 3, pp. 519—527 (in Russian).

58. **Sun Y., Huang X., Kroening D., Sharp J.** Structural Test Coverage Criteria for Deep Neural Networks, *2019 IEEE/ACM 41st Int. Conf. on Software Engineering: Companion Proc. (ICSE-Companion)*, 2019, pp. 320—321.

59. **Sun Y., Huang Z., Kroening D., Sharp J.** DeepConcolic: Testing and Debugging Deep Neural Networks, *2019 IEEE/ACM 41st Int. Conf. on Software Engineering: Companion Proc.*, 2019, pp. 111—114.

60. **Yao Y., Liu J., Huang S., Hui Z., Wu Z.** Testing Adequacy of Convolutional Neural Network based on Mutation Testing, *IEEE 19th Int. Conf.on Software Quality, Reliability and Security Companion (QRS-C),* 2019, pp. 536—537.

61. **Piuri V., Sami M. G., Sciuto D., Stefanelli R.** A behavioral approach to testability of neural networks, *IEEE Xplore,* 1992, pp. II-654—II-659.

62. **Kirkland L. V., Glenn R.** Wright Functioning Testing Philosophies using Neural Networks, *1997 IEEE Xplore*, 1997, pp. 88—91.

63. **Shem H., Fu Y.** Priority Directed Test Generation for Functional Verification using Neural Networks, *IEEE 2005 ASP-DAC*, 2005, pp. 1052—1055.

64. **Fine S., Ziv A.** Coverage Directed Test Generation for Functional Verification, *40th Design Automation Conference*, June 2—6, 2003, pp. 286—291.

УДК 004.4, 517.5 DOI: 10.17587/prin.12.200-208

Н. С. Астапов, канд. физ.-мат. наук, доц., ст. науч. сотр., nika@hydro.nsc.ru,
Новосибирский государственный университет, Институт гидродинамики
им. М. А. Лаврентьева СО РАН

Алгоритмы разложения на множители полиномов невысоких степеней

Для полиномов третьей степени специального вида представлены найденные автором разложения на линейные множители. Предложены различные способы разложения на множители полиномов четвертой степени общего и частного видов. Для полиномов шестой степени специального вида даны представления в виде произведения полиномов более низких степеней. Особое внимание уделено представлениям через квадратные трехчлены. Приведено разложение обобщенного возвратного полинома шестой степени на квадратные трехчлены.

***Ключевые слова*:** *программное обеспечение, полиномиальные множители, резольвента, возвратные полиномы*

Введение

К проблеме представления полиномов в виде произведения полиномов более низких степеней приводят, например, задачи символьного интегрирования рациональных функций, задачи решения характеристических уравнений, связанных с дифференциальными уравнениями. В монографии [1] отмечена актуальность поиска простых алгоритмов для решения алгебраических уравнений, приведены примеры инженерно-технических задач, в которых возникает необходимость решения алгебраических уравнений третьей—восьмой степеней.

Формулы Кардано являются компьютерно труднореализуемыми для алгебраических уравнений третьей и четвертой степеней с произвольными буквенными коэффициентами, так как не существует алгоритма извлечения кубического корня из произвольного комплексного числа. Для уравнений пятой и более высокой степени в общем виде нет простых алгоритмов для символьного представления корней через коэффициенты. Простые алгоритмы для символьного исследования алгебраических уравнений малых степеней востребованы и в задачах математики и механики [2—9]. Так, например, для проверки свойства сильной эллиптичности уравнений равновесия, которое имеет большое значение в теории упругости, необходимо символьное исследование алгебраических уравнений четвертой, шестой и двенадцатой степеней (см. стр. 690—692, 695, 696 [4]). В работе [4] даны критерии положительности полиномов четвертой и шестой степеней (теоремы 2, 3) и приведены доказанные авторами теоремы 6 и 7 о вещественных нулях полиномов третьей и четвертой степеней. В работах [5, 6] ключевым является символьное исследование характеристического уравнения четвертой степени с комплексными коэффициентами для нахождения неизвестных значений комплексного волнового параметра. В работах [7—9] рассмотрены некоторые частные случаи алгебраических уравнений третьей—восьмой степеней, для решения которых привлекаются формулы Ферро—Тартальи, Феррари, Кардано и отмечаются вычислительные трудности в общем случае.

Однако в частных случаях, когда коэффициенты полиномов связаны какими-либо дополнительными соотношениями, иногда удается получить достаточно простые разложения на множители, что показывается приведенными ниже разложениями. Настоящая работа является естественным продолжением работ [10, 11].

Полиномы третьей степени

Уравнение третьей степени в каноническом виде ($p \neq 0,\ \ q \neq 0$)

$$x^3 + px + q = 0 \tag{1}$$

подстановкой $x = 1/z$ приводится к виду $z^3 + pz^2/q + 1/q = 0$, затем с помощью подстановки $z = t - p/(3q)$ к виду

$$t^3 - \frac{p^2}{3q^2}t + \frac{2p^3 + 27q^2}{27q^3} = 0. \tag{2}$$

1. Уравнение (2) при условии $2p^3 + 27q^2 = 0$ имеет корень $t_1 = 0$, следовательно, $z_1 = -p/(3q)$ и в этом случае уравнение (1) имеет корень $x_1 = -3q/p$. Это решение можно получить из формул Ф. Клейна для "уравнения диэдра" [10, 12, 13]. Итак, если для комплексных коэффициентов p, q выполняется равенство $2p^3 + 27q^2 = 0$, то справедливо разложение на линейные множители

$$x^3 + px + q = \left(x + \frac{3q}{p}\right)\left(x - \frac{3q}{2p}\left(1 + \sqrt{3}\right)\right)\left(x - \frac{3q}{2p}\left(1 - \sqrt{3}\right)\right). \tag{3}$$

2. Запишем уравнение (2) в виде $t^3 + mt + n = 0$, где $m = -p^2/3q^2$, $n = (2p^3 + 27q^2)/27q^3$. Уравнение (2) при условии $2m^3 + 27n^2 = 0$ имеет корни $t_1 = -3n/m$, $t_{2,3} = 3n(1 \pm \sqrt{3})/(2m)$. Возвращаясь к уравнению (1) и переменным p и q, получим следующее разложение. Если для коэффициентов p, q выполняется равенство $q^2 = -(p/3)^3(2 \pm \sqrt{2})$, то справедливо разложение на линейные множители

$$x^3 + px + q = \\ = \left(x - 3p^2q/\left(p^3 + 27q^2\right)\right)(x - x_2)(x - x_3), \tag{4}$$

где $x_{2,3} = \pm 6p^2q/\left(2p^3(\sqrt{3} \mp 2) + 27q^2(\sqrt{3} \mp 1)\right)$. То есть и в этом случае корни уравнения (1) выражаются через коэффициенты рационально, без операции извлечения корней из коэффициентов.

Заметим, что для любых действительных чисел p полином $M_3 = x^3 + px + \sqrt{-p^3(6 + 3\sqrt{2})}/9$ имеет три действительных корня ("неприводимый случай"). Пакет прикладных программ Mathematica 8.0 с помощью команды Solve находит корни полинома M_3 в виде

$$\frac{-2 2^{1/3} p + 2^{2/3}\left(-\sqrt{2+\sqrt{2}}\sqrt{-p^3} + \sqrt{-(-2+\sqrt{2})p^3}\right)^{2/3}}{2\sqrt{3}\left(-\sqrt{2+\sqrt{2}}\sqrt{-p^3} + \sqrt{-(-2+\sqrt{2})p^2}\right)^{1/3}}.$$

Поэтому Mathematica 8.0 не может найти разложение для M_3 и вырабатывает с помощью команды Factor следующее решение:

```
q = Sqrt[-(p/3)^3(2 + Sqrt[2])]
Factor[x^3 + px + q]
```

$$\frac{1}{9}\left(\sqrt{3(2+\sqrt{2})}\sqrt{-p^3} + 9px + 9x^3\right).$$

3. Уравнение (1) при условии $4p^3 + 27q^2 = 0$ имеет двукратный корень $x_{1,2} = -3q/(2p)$. Поэтому имеем разложение на линейные множители

$$x^3 + px + q = \left(x + 3q/(2p)\right)^2 (x - 3q/p). \tag{5}$$

Отметим, что разложения (3)—(5) кубического полинома на линейные множители не используют операции извлечения корней из коэффициентов и справедливы для полиномов с произвольными комплексными коэффициентами.

Полиномы четвертой степени

Разложение на множители полинома четвертой степени в общем виде приводит к необходимости решения кубического уравнения (резольвенты). Так алгоритм Декарта $(q \neq 0)$ можно записать в виде

$$x^4 + px^2 + qx + r = \\ = \left(x^2 + kx + (t - q/k)/2\right)\left(x^2 - kx + (t + q/k)/2\right), \tag{6}$$

где $k = \sqrt{t - p}$, а t — любой корень кубического уравнения (резольвенты)

$$t^3 - pt^2 - 4rt + 4pr - q^2 = 0. \tag{7}$$

Рассмотрим частные случаи, когда резольвента легко разрешима. Отметим, что для приведенных ниже конкретных разложений для краткости не оговариваются очевидные ограничения на значения коэффициентов полиномов. Поэтому при программировании необходимо тщательно отслеживать эти ограничения и запрещать соответствующие разложения. Например, разложения **1**, **3**, **10** неприменимы при $p = 0$, разложение **2** — при $k = 0$, разложения **4**, **5** — при $a = 0$, разложение **6** — при $p = q$.

1. Если $4pr - q^2 = 0$, то $t_1 = 0$ и из равенства (6) получим разложение

$$x^4 + px^2 + qx + \frac{q^2}{4p} = \\ = \left(x^2 + \sqrt{-p}x - \frac{q}{2\sqrt{-p}}\right)\left(x^2 - \sqrt{-p}x + \frac{q}{2\sqrt{-p}}\right),$$

где $p \neq 0$ и q — произвольные комплексные числа. Это разложение для произвольных комплексных чисел p и r можно записать в виде

$$x^4 + px^2 - 2\sqrt{pr}x + r = \\ = \left(x^2 + \sqrt{-p}x + \sqrt{-r}\right)\left(x^2 - \sqrt{-p}x - \sqrt{-r}\right).$$

Резольвента (7) подстановкой $t = z + p/3$ приводится к каноническому виду

$$z^3 - \left(p^2 + 12r\right)z/3 - \left(2p^3 - 72pr + 27q^2\right)/27 = 0. \tag{8}$$

2. Из резольвенты (8) видно, что если $p^2 + 12r = 0$, то $z_1 = \sqrt[3]{\left(8p^3 + 27q^2\right)}/3$ и разложение (6) имеет вид

$$x^4 + px^2 + qx - p^2/12 = \\ = \left(x^2 + kx + (t - q/k)/2\right)\left(x^2 - kx + (t + q/k)/2\right),$$

где $q \neq 0$, $k = \sqrt{t - p}$, $t = z_1 + p/3$. В этом случае можно использовать и разложение

$$x^4 + px^2 + qx - p^2/12 = \\ = \left(x^2 + kx + \left(t - \sqrt{t^2 + p^2/3}\right)/2\right) \times \\ \times \left(x^2 - kx + \left(t + \sqrt{t^2 + p^2/3}\right)/2\right),$$

где $k = \sqrt{t - p}$, если $q \geqslant 0$, и $k = -\sqrt{t - p}$, если $q \leqslant 0$.

3. Если $2p^3 - 72pr + 27q^2 = 0$, т. е. $r = (2p^3 + 27q^2)/72p$, то резольвента Декарта (7) имеет корень $t_1 = p/3$ и справедливо разложение, содержащее лишь квадратные радикалы

$$M_4 = x^4 + px^2 + qx + r =$$
$$= \left(x^2 + \frac{m}{3}x + \frac{p}{6} - \frac{3q}{2m}\right)\left(x^2 - \frac{m}{3}x + \frac{p}{6} + \frac{3q}{2m}\right),$$

где $m = \sqrt{-6p}$.

Пакет прикладных программ Mathematica 8.0 в этом случае не находит разложение для произвольных p и q:

$$\frac{p^2}{36} + \frac{3q^2}{8p} + qx + px^2 + x^4.$$

Даже для частного случая полиномов M_4 такого вида — для полинома $x^4 - 3x^2 + 2\sqrt{2}x - 3/4$, имеющего трехкратный корень $1/\sqrt{2}$, Mathematica 8.0 не находит разложение, хотя с помощью команды Solve находит корни правильно:

```
p = -3
q = 2Sqrt[-6*p]/3
r = (2p^3+27q^2)/(72p)
Factor[x^4+p x^2+q+r]
Solve[x^4+p x^2+q x+r==0,x]
m = Sqrt[-6*p]
(x^2 + m*x/3 + p/6-3*q/(2*m))(x^2-m x/3+p/6+
3q/(2*m))
Simplify[%]
```

$$\frac{1}{4}\left(-3 + 8\sqrt{2}x - 12x^2 + 4x^4\right)$$

$$\left\{x \to -\frac{3}{\sqrt{2}}\right\}, \left\{x \to \frac{1}{\sqrt{2}}\right\}, \left\{x \to \frac{1}{\sqrt{2}}\right\}, \left\{x \to \frac{1}{\sqrt{2}}\right\}.$$

Однако для полинома такого же вида $x^4 - 6x^2 + 4x$ Mathematica 8.0 вырабатывает верное разложение:

$$(-2 + x)x\left(-2 + 2x + x^2\right).$$

Ниже даны разложения, полученные без использования резольвенты.

4. $x^4 + ax^3 + bx^2 + cx + c(ab - c)/a^2 = \left(x^2 + c/a\right) \times$
$\times\left(x^2 + ax + (ab - c)/a\right)$.

5. $x^4 + ax^3 + bx^2 + cx + (c/a)^2 = \left(x^2 - t_1 x + c/a\right) \times$
$\times\left(x^2 - t_2 x + c/a\right)$,

где t_1 и t_2 — корни квадратного уравнения $t^2 + at + b - 2c/a = 0$.

6. Если $r = q^2\left(1 + 1/(p - q)\right)/4$, то справедливо разложение

$$M_4 = \left(x^2 + mx + \frac{q}{2}\left(1 - \frac{1}{m}\right)\right)\left(x^2 - mx + \frac{q}{2}\left(1 + \frac{1}{m}\right)\right),$$

где $m = \sqrt{q - p}$.

Система Mathematica 8.0 не может найти разложение для M_4 и вырабатывает с помощью команды Factor следующее решение:

```
r = q^2(1 + 1/p - q))/4
Factor[x^4+p x^2+q x+r]
FullSimplify[%]
```

$$\frac{(1 + p - q)q^2}{4(p - q)} + qx + px^2 + x^4.$$

7. Если $r = \left((p + q)^2 - q\right)/4$, то справедливо разложение

$$M_4 = \left(x^2 + mx + (p + q - m)/2\right) \times$$
$$\times\left(x^2 - mx + (p + q + m)/2\right),$$

где $m = \sqrt{q}$.

В этом случае Mathematica 8.0 предлагает решение

$$\frac{1}{4}\left(p^2 - q + 2pq + q^2 + 4qx + 4px^2 + 4x^4\right).$$

8. Если $r = \left(p(q + 1)/(2q)\right)^2 - q^3/(4p)$, то справедливо разложение

$$M_4 = \left(x^2 + mx - \frac{q^2 m}{2p} + n\right)\left(x^2 - mx + \frac{q^2 m}{2p} + n\right),$$

где $n = p(q + 1)/(2q)$, $m = \sqrt{p/q}$.

9. Если $r = \left(p(q + 1)/2\right)^2 - q/(4p)$, то справедливо разложение

$$M_4 = \left(x^2 + mx - \frac{m}{2p} + n\right)\left(x^2 - mx + \frac{m}{2p} + n\right),$$

где $n = p(q + 1)/2$, $m = \sqrt{pq}$.

10. Если $r = pq^2/\left(4\left(p^2 - q\right)\right) + \left(q/(2p)\right)^2$, то справедливо разложение

$$M_4 = \left(x^2 + mx + \frac{q}{2}\left(\frac{1}{p} - \frac{1}{m}\right)\right)\left(x^2 - mx + \frac{q}{2}\left(\frac{1}{p} + \frac{1}{m}\right)\right),$$

где $m = \sqrt{q/p - p}$.

Система Mathematica 8.0 не находит разложение и предлагает решение

$$\frac{pq^2}{4p^2 - 4q} + \frac{q^2}{4p^2} + qx + px^2 + x^4.$$

11. Если $r = \left(\left(p^2 + q\right)/(2p)\right)^2 - pq/4$, то справедливо разложение

$$M_4 = \left(x^2 + mx + \left(p^2 + q\right)/(2p) - pm/2\right) \times$$
$$\times\left(x^2 - mx + \left(p^2 + q\right)/(2p) + pm/2\right),$$

где $m = \sqrt{q/p}$.

12. Если $r = \left(\left(p^2 - 3q\right)/(2p)\right)^2 + pq/12$, то справедливо разложение

$$M_4 = \left(x^2 + mx + \frac{p}{2}\left(1 + \frac{m}{3}\right) + \frac{m^2}{2}\right) \times$$
$$\times \left(x^2 - mx + \frac{p}{2}\left(1 - \frac{m}{3}\right) + \frac{m^2}{2}\right),$$

где $m = \sqrt{-3q/p}$.

Система Mathematica 8.0 не находит разложение и предлагает решение

$$\frac{p^2}{4} + \frac{pq}{12} + \frac{9q^2}{4p^2} + q\left(-\frac{3}{2} + x\right) + px^2 + x^4.$$

Так как в равенстве (6) $t = p + k^2$, то разложение (6) можно записать без радикалов следующим образом

$$x^4 + px^2 + qx + r =$$
$$= \left(x^2 + kx + \left(p + k^2 - q/k\right)/2\right) \times \qquad (9)$$
$$\times \left(x^2 - kx + \left(p + k^2 + q/k\right)/2\right),$$

где k любой корень бикубической относительно k резольвенты (7), которую можно представить в виде

$$r = \left(\left(p + k^2\right)/2\right)^2 - \left(q/(2k)\right)^2. \qquad (10)$$

Итак, для представления любого полинома четвертой степени с произвольными комплексными коэффициентами p, q и r в виде произведения квадратичных множителей (9) достаточно найти один корень k уравнения (10). И наоборот, для произвольных комплексных значений параметров p, q и k получим разложение (9) полинома специального вида $x^4 + px^2 + qx + r$, где коэффициенты p, q произвольны, а коэффициент r определяется равенством (10). Ниже даны примеры разложений частного вида, полученных из (9), (10) для некоторых конкретных значений k.

13. Полагая $k = 1$, находим $r = \left((p+1)^2 - q^2\right)/4$ и получаем разложение

$$M_4 = \left(x^2 + x + \frac{p - q + 1}{2}\right)\left(x^2 - x + \frac{p + q + 1}{2}\right).$$

14. Полагая $k = q$, находим $r = \left(\left(p + q^2\right)^2 - 1\right)/4$ и получаем разложение

$$M_4 = \left(x^2 + qx + \left(p + q^2 - 1\right)/2\right) \times$$
$$\times \left(x^2 - qx + \left(p + q^2 + 1\right)/2\right).$$

15. Полагая $k = \sqrt{p}$, находим $r = p^2 - q^2/(4p)$ и получаем разложение

$$M_4 = \left(x^2 + kx + p - \frac{q}{2k}\right)\left(x^2 - kx + p + \frac{q}{2k}\right).$$

Это разложение можно получить и из разложения (6), выбирая в резольвенте (7) корень $t = 2p$ (при $r = p^2 - q^2/(4p)$). Заметим, что Mathematica 8.0 в этом случае не находит разложение и вырабатывает результат в виде $p^2 - \frac{q^2}{4p} + qx + px^2 + x^4$.

Приведем еще два разложения специального вида, когда лишь один коэффициент p произвольный, а коэффициенты q и r зависят от параметра p.

16. Если $r = p^2/12$ и $4p^3 - 27q^2 = 0$, то справедливо разложение

$$x^4 + px^2 \pm \frac{2\sqrt{3p^3}}{9}x + \frac{p^2}{12} =$$
$$= \left(x^2 + \frac{\sqrt{-6p}}{3}x + \left(1 \pm \sqrt{2}i\right)\frac{p}{6}\right) \times$$
$$\times \left(x^2 - \frac{\sqrt{-6p}}{3}x + \left(1 \mp \sqrt{2}i\right)\frac{p}{6}\right),$$

где i — мнимая единица.

17. Если $r = p^2/12$ и $8p^3 - 27q^2 = 0$, то справедливо разложение

$$x^4 + px^2 \pm \frac{2\sqrt{6p^3}}{9}x + \frac{p^2}{12} =$$
$$= \left(x^2 \pm \frac{2\sqrt{-3p}}{3}x - \left(1 - \sqrt{2}i\right)\frac{p}{6}\right) \times$$
$$\times \left(x^2 \mp \frac{2\sqrt{-3p}}{3}x - \left(1 + \sqrt{2}i\right)\frac{p}{6}\right).$$

18. В этом разложении свободным является коэффициент q:

$$x^4 + \frac{q^2 - 1}{2}x^2 + qx + \left(\frac{q^2 - 1}{4}\right)^2 =$$
$$= \left(x^2 + x + \left(\frac{q-1}{2}\right)^2\right)\left(x^2 - x + \left(\frac{q+1}{2}\right)^2\right).$$

19. Кратные корни. Если полином $x^4 + px^2 + qx + r$ имеет корень кратности 2, 3 или 4, то его коэффициенты удовлетворяют условию Ф. Клейна ([12], см. с. 143):

$$\left(\left(p^2 + 12r\right)/12\right)^3 -$$
$$-27\left(\left(2p^3 - 72pr + 27q^2\right)/216\right)^2/4 = 0. \qquad (11)$$

Так как двукратный корень полинома является и корнем производной этого полинома, то для этого корня, когда полином $x^4 + px^2 + qx + r$ имеет ровно один двукратный корень, получим следующее выражение:

$$k = q\left(p^2 + 12r\right)/\left(2p^3 - 8pr + 9q^2\right). \qquad (12)$$

Поэтому соответствующее разложение можно записать в виде

$$x^4 + px^2 + qx + r = (x-k)^2\left(x^2 + 2kx + r/k^2\right) = \\ = (x-k)^2\left(x^2 + 2kx + 3k^2 + p\right), \quad (13)$$

где k определяется равенством (12). Полином $x^4 + px^2 + qx + r$ имеет два двукратных корня только в случае, когда $q = 0$ и $r = p^2/4$. В этом случае разложение можно записать в виде $x^4 + px^2 + p^2/4 = \left(x^2 + p/2\right)^2$. При $r = p^2/4$ условие Ф. Клейна (11) принимает вид $q^2\left(32p^3 - 27q^2\right) = 0$. Поэтому если выполняются равенства $r = p^2/4$ и $27q^2 = 32p^3$, то полином $x^4 + px^2 + qx + r$ имеет ровно один двукратный корень и справедливо разложение (13).

Возможность представления полинома M_4 в виде (13) проверяется просто: если для коэффициентов p, $q \neq 0$ и $r \neq -p^2/12$ полинома M_4 выполняется условие (11), то M_4 имеет ровно один двукратный корень и справедливо разложение (13). Отметим, что уравнение (11) является биквадратным относительно q. Поэтому если для произвольных коэффициентов p и $r \neq -p^2/12$ из уравнения (11) выберем любой корень q, то для полинома $x^4 + px^2 + qx + r$ выполняется равенство (13). Например, если выберем $r = p^2/36$ и из уравнения (11), которое в этом случае принимает вид $(q/4)^4 = \left(p^2/27\right)^3$, выберем любой корень q, то полином M_4 представим в виде (13). Если, например, выберем $p = 0$ и q из равенства $(q/4)^4 = (r/3)^3$, то также имеет место представление (13).

Если $p^2 + 12r = 0$ и выполняется равенство $8p^3 + 27q^2 = 0$, которое следует из условия (11) при $r = -p^2/12$, то полином $x^4 + px^2 + qx + r$ имеет трехкратный корень $k = 2p^2/(9q) = -8r/(3q)$ и справедливо разложение

$$x^4 + px^2 + qx + r = \left(x - 2p^2/(9q)\right)^3 \times \\ \times\left(x + 2p^2/(3q)\right) = \left(x + 8r/(3q)\right)^3 (x - 8r/q).$$

Полином $x^4 + px^2 + qx + r$ имеет четырехкратный корень только при условии $p = q = r = 0$.

Полином пятой степени

Кратные корни. Выражение для двукратного корня полинома пятой степени в каноническом виде $M_5 = x^5 + bx^3 + cx^2 + dx + e$ слишком громоздко и здесь не приводится, однако если $b = 0$, то двукратный корень равен

$$k = \\ = -\left(27c^3d + 375ce^2 - 400d^2e\right)/\left(2\left(27c^4 + 300cde - 160d^3\right)\right),$$

при условии, что такой корень один. Поэтому для полинома M_5, имеющего один двукратный корень, легко построить алгоритм разложения на множители не выше третьей степени. Если полином M_5 имеет два двукратных корня, то они являются корнями квадратного уравнения

$$\left(12b^3 - 40bd + 45c^2\right)x^2 + \\ + 2\left(4b^2c - 25be + 30cd\right)x + 4b^2d + 75ce = 0$$

и M_5 разлагается на полиномы первой степени.

Если полином M_5 имеет трехкратный корень, то он так же как и двукратный корень выражается через коэффициенты в рациональном виде. В случае если $c = 0$, это выражение принимает вид $k_{1,2,3} = 100e/\left(21b^2 - 100d\right)$ и при $21b^2 - 100d \neq 0$ полином M_5 разлагается на полиномы не выше второй степени. Если $21b^2 - 100d = 0$, то M_5 имеет четырехкратный корень и разлагается на полиномы первой степени.

Разложения на множители полиномов пятой степени специального вида можно получить из разложений полиномов шестой степени, которые приводятся ниже. Здесь приведем еще одно представление M_5, когда два коэффициента полинома свободны (произвольны), а два других связаны, т. е. выражаются через свободные:

$$x^5 + bx^3 + cx^2 + b^2x/4 + bc/2 = \left(x^2 + b/2\right)\left(x^3 + bx/2 + c\right).$$

Полином шестой степени

Кратные корни. Необходимым и достаточным условием для того, чтобы у полинома $x^6 + cx^4 + ex^2 + g$ было ровно два двукратных корня, является равенство $4c^3g - c^2e^2 - 18ceg + 4e^3 + 27g^2 = 0$ при условии, что $c^2 \neq 3e$. В этом случае получим разложение $x^6 + cx^4 + ex^2 + g = \left(x^2 - k\right)^2\left(x^2 - m\right)$, где k и m можно найти по формулам $k = (9g - ce)/\left(2c^2 - 6e\right)$, $m = g\left(c^2 - 3e\right)/\left(3cg - e^2\right)$. Если $c^2 = 3e$, то $g = c^3/27$ и полином $x^6 + cx^4 + ex^2 + g$ имеет два трехкратных корня, равных $\pm\sqrt{-c/3}$, которые будут действительными числами только если $c \leqslant 0$.

Приведем различные представления полинома шестой степени $M_6 = x^6 + cx^4 + dx^3 + ex^2 + fx + g$, у которого один или два коэффициента связаны, а остальные свободны, в виде произведения двух полиномов третьей степени.

1. $M_6 = \left(x^3 \pm mx^2 + fx/d + (n+d)/2\right)\left(x^3 \mp mx^2 + \right.$

$\left. + fx/d - (n-d)/2\right)$,

где $m = \sqrt{2f/d - c}$, $n = \sqrt{d^2 - 4g}$. Здесь коэффициенты c, d, f, g свободны, а связанный коэффициент e является любым корнем уравнения $d^4e^2 - 2d^2f^2e + cd^6 - 4cd^4g - 2d^5f + 8d^3fg + f^4 = 0$. В общем виде Mathematica 8.0 не находит разложение, и, например, при $c = -1$, $d = -2$, $f = -4$, $g = -5$ вырабатывает следующий результат:

```
m = Sqrt[2f/d - c]
n = Sqrt[d^2 - 4g]
e = 4-2Sqrt[30]
Factor[x^6+c x^4+d x^3+e x^2+f x+g]
FullSimplify[%]
```

$$-5 + x\left(-4+x\left(4 - 2\sqrt{30} + x\left(-2 - x + x^3\right)\right)\right).$$

2. $M_6 = \left(x^3 + mx^2 + (n+f)\,x/(2s) + s\right)\left(x^3 - mx^2 - \right.$
$\left. -(n-f)\,x/(2s) + s\right)$,

$s = \sqrt{g}$, $m = \sqrt{f/s - c}$, $n = \sqrt{f^2 - 4ge}$. Здесь свободны коэффициенты c, e, f, g, а коэффициент d находится из уравнения

$$s^3d^2 - 4s^4d - 4ces^3 + cf^2s + 4efs^2 - f^3 + 4s^5 = 0.$$

3. $x^6 + cx^4 + dx^3 + ex^2 \pm fx + g = \left(x^3 + kx^2 + mx + s\right) \times$
$\times\left(x^3 - kx^2 + nx + s\right)$,

где

$$m = \left((\pm 2g - ds)\,k - cf + f^2/s\right)/(2t),$$

$$n = \left((\mp 2g + ds)\,k - cf + f^2/s\right)/(2t), \quad s = \sqrt{g}, \quad t = f - cs,$$

$$k = \sqrt{t/s}.$$

Здесь коэффициенты c, d, f, g свободны, а коэффициент e находится из равенства

$$e = \left(-s\,(d \mp 2s)^2/t + f^2/g\right)/4.$$

4. $M_6 = \left(x^3 + kx^2 + \left(\dfrac{k^2}{4} - \dfrac{g}{k^4}\right)x - \dfrac{k^3}{2}\right) \times$
$\times\left(x^3 - kx^2 + \left(\dfrac{3k^2}{4} + \dfrac{g}{k^4} + c\right)x - \dfrac{2g}{k^3}\right)$,

где $k = d/c$, коэффициенты c, d, g свободны, а связанные коэффициенты e и f находятся из равенств

$$e = 11k^4/16 + k^2/4 - g^2/k^8 - cg/k^4 - 5g/(2k^2),$$

$$f = 2g^2/k^7 - d/(2k^2) - g/k - 3k^5/8.$$

5. $x^6 + cx^4 + dx^3 + ex^2 + g = \left(x^3 + kx^2 + (d-n)/2\right) \times$
$\times\left(x^3 - kx^2 + (d+n)/2\right)$,

где один связанный коэффициент $g = \left(cd^2 + e^2\right)/(4c)$, $k = \sqrt{-c}$, $n = e/k$. Mathematica 8.0 не находит разложение и вырабатывает решение

```
k = Sqrt[-c]
n = e/k
g = (c d^2+e^2)/(4c)
Factor[x^6+c x^4+d x^3+e x^2+f x+g]
FullSimplify[%]
```

$$\frac{d^2}{3} + \frac{e^2}{4c} + dx^3 + cx^4 + x(f + ex + x^5)$$

6. $x^6 + cx^4 + dx^3 + fx + g = \left(x^3 + kx^2 + mx + n - \dfrac{m^2}{2k}\right) \times$
$\times\left(x^3 - kx^2 - mx + n + \dfrac{m^2}{2k}\right)$,

где один связанный коэффициент $g = dmk - cm^2 + m^4/(4c) + d^2/4$, $k = \sqrt{-c}$, $m = \sqrt[3]{fk}$, $n = d/2 - cm/k$.

7. $x^6 + bx^5 + cx^4 + dx^3 + ex^2 + fx + g = \left(x^3 + e/b\right) \times$
$\times\left(x^3 + bx^2 + cx + d - e/b\right)$,

где коэффициенты b, c, d, e свободны, а связанные коэффициенты f, g находятся из равенств $f = ce/b$, $g = e\,(bd - e)/b^2$. Полагая $e = bd$, из этого разложения получим разложение, не содержащее операцию извлечения корней, для полинома пятой степени

$$x^5 + bx^4 + cx^3 + dx^2 + bdx + cd = \left(x^3 + d\right)\left(x^2 + bx + c\right),$$

в котором свободны три коэффициента b, c, d.

Рассмотрим теперь разложения полинома шестой степени на полиномы второй и четвертой степеней.

8. $x^6 + bx^5 + cx^4 + dx^3 + ex^2 + fx + g =$
$= \left(x^2 + bx + c - \dfrac{d}{b}\right)\left(x^4 + \dfrac{d}{b}x^2 + n\right)$,

где $n = e - (bc - d)\,d/b^2$, коэффициенты b, c, d, e свободны, а связанные коэффициенты f, g находятся из равенств $f = be - cd + d^2/b$, $g = (bc - d)\,f/b^2$. Если в этом разложении положить $n = 0$, т. е. $e = (bc - d)\,d/b^2$, то получим разложение

$$x^4 + bx^3 + cx^2 + dx + \frac{d\,(bc - d)}{b^2} =$$
$$= \left(x^2 + bx + \frac{(bc - d)}{b}\right)\left(x^2 + \frac{d}{b}\right).$$

9. $M_6 = \left(x^2 - kx + n\right) \times$
$\times\left(x^4 + kx^3 + \left(k^2 + c - n\right)x^2 + \left(nk + f/n\right)x + n^2\right)$,

где $k = n\left(cn - e\right)/f$, $n^3 = g$, коэффициенты c, e, f, g свободны, а связанный коэффициент d находится из равенства $d = f/n + (3n - c)\,k - k^3$. Отсюда легко получить следующие разложения:

$$x^6 + x^4 + x^3 + x^2 + x + 1 = \left(x^2 + 1\right)\left(x^4 + x + 1\right),$$

$$x^6 - x^4 + x^3 - x^2 + x + 1 = \left(x^2 + 1\right)\left(x^4 - 2x^2 + x + 1\right),$$

$$x^6 - x^4 - x^3 + x^2 - x + 1 =$$
$$= (x - 1)^2\left(x^4 + 2x^3 + 2x^2 + x + 1\right).$$

В этом случае Mathematica 8.0 не находит разложение M_6 для произвольных значений коэффициентов c, d, f и g, однако при $c = 4$, $d = 2$, $f = 3$, $g = -1$ и $n = -1$ находит правильное разложение:

```
n = -1
k = n (c n-e)/f
d = f/n+(3n-c)k-k^3
Factor[x^6+c x^4+d x^3+e x^2+f x+g]
Expand[%]
(-1 - 2x + x^2)(1 - 5^x + 9x^2 + 2x^3 + x^4)
-1 + 3x + 2x^2 - 25x^3 +4x^4 + x^6.
```

10. $M_6 = \left(x^2 - kx + n\right)\left(x^4 + kx^3 + \left(k^2 + c - n\right)x^2 + \right.$
$\left. + \left(ck + d + k^3 - 2n^3\right)x + k\right)$,

где $k = n^2$, $n^3 = g$, коэффициенты c, d, g свободны, а связанные коэффициенты находятся из равенств $d = -k^4 + 3n^5 - ck^2 - dk + cn$, $d = n^7 - 3k^2 + cn^3 + dn$.

11. $M_6 = \left(x^2 + kx + n\right)\left(x^4 - kx^3 + \left(e/n + k^2/4 - n/2\right)\times\right.$,

$\left.\times x^2 - fx/n + n^2/2\right)$,

где $k = 4f/n^2$, $n^3 = 2g$, связанные коэффициенты c, d находятся из равенств $c = e/n - 3k^2/4 + n/2$, $d = f\left(2e/g + (2f/g)^2 - 7/n\right)$, а коэффициенты e, f, g свободны.

12. $x^6 + cx^4 + dx^3 + ex^2 + g = \left(x^2 - kx + n\right)\times$

$$\times\left(x^4 + kx^3 + \left(\frac{2e}{n} - c\right)x^2 + \frac{kn}{2}x + \frac{n^2}{2}\right),$$

где $k = \sqrt{n^2 + 2e - 2cn}/\sqrt{n}$, $n^3 = 2g$, связанный коэффициент d находится из равенства $d = k\left(c - 2e/n + 3n/2\right)$, а коэффициенты c, e, g свободны. Система Mathematica 8.0 в этом случае не находит разложение M_6 для произвольных значений коэффициентов c, e и g, однако при $c = 1$, $e = 1$, $g = 1/2$ и $n = 1$ находит правильное разложение:

```
g = 1/2
n = (2g)^(1/3)
n = 1
k = Sqrt[n^2+2 e-2 c n]/Sqrt[n]
d = k(c - 2e/n+3n/2)
Factor[x^6+c x^4+d x^3+e x^2+g]
Expand[%]
```

$$\frac{1}{2}(1 - x + x^2)(1 + x + 2x^2 + 2x^3 + 2x^4)$$

$$\frac{1}{2} + x^2 + \frac{x^3}{2} + x^4 + x^6.$$

13. $x^6 + cx^4 + dx^3 + ex^2 + g = \left(x^2 - kgx + g\right)\times$

$$\times\left(x^4 + kgx^3 + (eg - c)x^2/g^2 + kx + 1\right),$$

где $k = \sqrt{\left(e - cg + g^2 - 1\right)/\left(g\left(g^2 - 1\right)\right)}$, связанный коэффициент d находится из равенства $d = k\left((cg - e)/\left(g^2 - 1\right) - e + g^2 + 1\right)$, а коэффициенты c, e, g свободны.

14. $M_6 = \left(x^2 - kx + n\right)\left(x^4 + kx^3 + mx^2 - ex/k - mn\right)$,

где $k = \pm\sqrt{n - c - g/n}$, $m = k^2 + c - n$, коэффициенты c, e, g свободны, а связанные коэффициенты d и f находятся из равенств $d = -\left(k^4 + (c - 2n)k^2 + e\right)/k$, $f = n\left(k^4 + (c - n)k^2 - e\right)/k$.

15. $M_6 = \left(x^2 + kx + n\right)\left(x^4 - kx^3 - gx^2/n^2 + ex/k + g/n\right)$,

где $k = \pm\sqrt{n^3 - cn^2 - g}/n$, коэффициенты c, e, g свободны, а коэффициенты d, f находятся из равенств $d = \left(en^2 - k^2\left(g + n^3\right)\right)/\left(kn^2\right)$, $f = \left(en^2 + gk^2\right)/(kn)$.

16. $M_6 = \left(x^2 - kx + g\right)\left(x^4 + kx^3 + \left(k^2 + c - g\right)x^2 + \right.$

$\left.+\left(k^3 + ck - 2gk + d\right)x + 1\right)$,

где k произвольно, коэффициенты c, d, g свободны, а связанные коэффициенты e и f находятся из равенств

$$e = -k^4 + (3g - c)k^2 - dk + cg - g^2 + 1,$$

$$f = gk^3 + \left(cg - 2g^2 - 1\right)k + dg.$$

Если удается найти k из одного из этих уравнений (четвертой или третьей степени относительно k), то связанным остается лишь один коэффициент e или f.

17. $M_6 = \left(x^2 - kx + n\right)\times$

$$\times\left(x^4 + kx^3 + \left(n + k^2\right)x^2 + \left(d + k^3\right)x + nk^2\right),$$

где $n = c/2$, $k = \pm 2\sqrt{g}/c$, коэффициенты c, d, g свободны, а коэффициенты e, f находятся из равенств $e = n^2 + ck^2 - dk - k^4$, $f = cd/2$. Интересно отметить, что хотя система Mathematica 8.0 в этом случае находит правильное символьное разложение M_6 для произвольных коэффициентов c, d, g, при $c = \sqrt{Z}$, где Z не является полным квадратом, например, $c = \sqrt{13}$, система не находит разложение:

$$-\frac{1}{4c^4}\left(-c^2 + 4\sqrt{g}x - 2cx^2\right)\left(4c^2g + 2c^3dx + 16g^{3/2}x + \right.$$

$$\left.+c^4x^2 + 8cgx^2 + 4c^2\sqrt{g}x^3 + 2c^3x^4\right).$$

18. $M_6 = \left(x^2 - kx + n\right)\times$

$$\times\left(x^4 + kx^3 + nx^2 + dx + (dn - f)/k\right),$$

где $n = \left(c + k^2\right)/2$, k произвольно, коэффициенты c, d, f свободны, а коэффициенты e и g находятся из равенств

$$g = \left(k^2 + c\right)\left(dk^2 + cd - 2f\right)/(4k),$$

$$e = \left(k^5 + 2ck^3 - 2dk^2 + c^2k + 2cd - 4f\right)/(4k).$$

Если удается найти k из уравнения четвертой степени относительно k, то связанным остается лишь один коэффициент e.

19. $M_6 = \left(x^2 - kx + n\right)\left(x^4 + kx^3 + \left(k^2 + c - n\right)x^2 + \right.$

$\left.+\left(k^3 + ck - 2nk + d\right)x + k^2n\right)$,

где $k = (f - dn)/(n(c - dn))$, параметр n находится из квадратного уравнения

$$\left(d^2 - 4g\right)n^2 + 2(2cg - df)n + f^2 - c^2g = 0.$$

В частности, $n_{1,2} = c/\left(2 \pm d/\sqrt{g}\right)$, если $f = 0$. Коэффициенты c, d, f, g свободны, а один связанный коэффициент e находится из равенства $e = -k^4 + (4n - c)k^2 - dk + cn - n^2$.

20. $M_6 = \left(x^2 - kx + k^2\right)\left(x^4 + kx^3 + cx^2 + \right.$

$\left.+\left(ck + d - k^3\right)x + dk + e - k^4\right)$,

где k является корнем кубического уравнения $ck^3 - ek = f$. Коэффициенты c, d, e, f свободны, а один связанный коэффициент g находится из равенства $g = k^2(dk + e - k^4)$.

21. $M_6 = (x^2 - kx + k^2)(x^4 + kx^3 + cx^2 + (ck + d - k^3)x + g/k^2)$,

где k — произвольный параметр. Коэффициенты c, d, g свободны, а связанные коэффициенты e, f находятся из равенств

$$f = (ck^4 + dk^3 - g - k^6)/k, \quad e = (k^6 + g - dk^3)/k^2.$$

Выбирая параметр k как специальным образом построенную функцию коэффициентов полинома, из разложения **21** можно получить разложения полинома M_6, в которых свободными будут другие коэффициенты. Например, если $k^3 = f/c$, то $e = 0$ и получим разложение

$$x^6 + cx^4 + dx^3 + fx + mn = \\ = (x^2 - kx + k^2)(x^4 + kx^3 + cx^2 + (ck + n)x + kn),$$

где $n = d - m$, $m = f/c$. А если $k = -e/d$, то $g = -k^6$ и разложение **21** принимает вид

$$x^6 + cx^4 + dx^3 + ex^2 + k^2nx - k^6 = \\ = (x^2 - kx + k^2)(x^4 + kx^3 + cx^2 + (n - k^3)x - k^4),$$

где $n = ck + d$.

22. $M_6 = (x^2 - kx + m)(x^4 + kx^3 + (k^2 + 2)x^2 + knx + n^2)$,

где $k = f/(n(m - n))$, $n = \pm\sqrt{g/m}$, $m = c - 2$. Коэффициенты c, f, g свободны, а связанные коэффициенты d, e находятся из равенств $d = -k(k^2 - n - c + 4)$, $e = fk/n + g/m + 2m$. Так как здесь полином четвертой степени является возвратным, то он раскладывается на квадратичные множители и, следовательно, разложение M_6 может быть записано с использованием лишь квадратных радикалов. Система Mathematica 8.0 не находит разложение, и при $c = 1$, $f = 2$, $g = -3$ вырабатывает решение:

```
m = c-2
n = Sqrt[g/m]
k = f/(n(m-n))
e = f k/n+n^2+2m
d = -k(k^2-n-c+4)
Factor[x^6+c x^4+d x^3+e x^2+f x+g]
FullSimplify[%]
```

$$-3 + \frac{1}{9}x(18 + x(15 - 6\sqrt{3} + \\ + x(54 - 28\sqrt{3} + 9(x + x^3)))).$$

23. $M_6 = (x^2 - kx + n)(x^4 + kx^3 + (k^2 + c - n)x^2 + (ck + d + k^3 - 2n^3)x + m)$,

где $m = (ckn + dn - f + k^3n - 2kn^2)/k$, k произвольно, n — любой корень квадратного уравнения

$$3kn^2 - (2ck + d + 4k^3)n + k^5 + ck^3 + dk^2 + ek + f = 0.$$

Коэффициенты c, d, e, f свободны, а один связанный коэффициент g находится из равенства $g = nm$.

24. $M_6 = (x^2 + f/d)(x^2 - mx + (cm + d)/(2m)) \times (x^2 + mx + (cm - d)/(2m))$,

где $m = \sqrt{f/d}$, связанные коэффициенты e, g находятся из равенств

$$e = (c^2d^2f + 4cdf^2 - d^5 - 4f^3)/(4d^2f),$$

$$g = (c^2f - d^3)/(4d),$$

а коэффициенты c, d, f свободны.

25. Возвратный полином шестой степени можно представить в виде произведения квадратного трехчлена и возвратного полинома четвертой степени:

$$x^6 + bx^5 + cx^4 + dx^3 + ncx^2 + n^2bx + n^3 = \\ = (x^2 + (b - k)x + n)(x^4 + kx^3 + mx^2 + knx + n^2),$$

где $m = k^2 - kb - n + c$, коэффициенты n, b, c, d произвольны. Коэффициент k — любой корень кубического уравнения

$$k^3 - 2bk^2 - (3n - b^2 - c)k + b(n - c) + d = 0.$$

Известно, что возвратный полином четвертой степени представим в виде произведения квадратных трехчленов с использованием лишь квадратных радикалов из коэффициентов исходного полинома. Заметим, что полином четвертой степени в разложении **25** является возвратным. Поэтому если параметр k выражается из кубического уравнения через квадратные радикалы из коэффициентов, то и возвратный полином шестой степени также можно выразить лишь через квадратные радикалы. Таким образом, возвратный полином шестой степени всегда можно разложить на квадратичные множители, используя радикалы не выше третьей или даже второй степени. Например,

$$x^6 + x^5 + x^4 + 2x^3 + x^2 + x + 1 = (x^2 + 2x + 1) \times \\ \times (x^4 - x^3 + 2x^2 - x + 1) = (x + 1)^2(x^2 + 1) \times \\ \times (x^2 - x + 1) = (x^3 + 1)(x^3 + x^2 + x + 1).$$

26. Приведем еще одно разложение возвратного полинома шестой степени:

$$x^6 + bx^5 + cx^4 + dx^3 + cnx^2 + bn^2x + n^3 = \\ = (x^2 - t_1x + n)(x^2 - t_2x + n)(x^2 - t_3x + n),$$

где коэффициенты n, b, c, d произвольны, а t_i — корни кубической резольвенты

$$t^3 + bt^2 + (c - 3n)t + d - 2bn = 0.$$

Заключение

Основная часть данной работы посвящена разложениям на множители полиномов четвертой и шестой степеней специального вида. Полученные результаты могут быть полезны при символьном исследовании алгебраических уравнений, возникающих в задачах механики, физики.

Особое внимание уделено разложениям полиномов на квадратные трехчлены и разложениям, в которых не используются радикалы, кроме квадратных, из коэффициентов полиномов. Эффективность предложенных разложений проиллюстрирована сравнением с решениями, генерируемыми пакетом прикладных программ Mathematica 8.0.

Представленные разложения могут служить основой при проектировании программ, дополняющих имеющиеся пакеты прикладных программ простыми алгоритмами в части разложения полиномов на множители, символьного интегрирования, точного символьного решения алгебраических уравнений четвертой и шестой степеней. Результаты работы могут быть использованы на занятиях по информатике и при проведении студенческих олимпиад.

Список литературы

1. **Кутищев Г. П.** Решение алгебраических уравнений произвольной степени: теория, методы, алгоритмы. — М.: URSS, 2010. — 232 с.
2. **Гашков С. Б.** О сложности интегрирования рациональных дробей // Труды МИАН. — 1997. — Т. 218. — С. 122—133.
3. **Mochimaru Y.** Reciprocal solution of a quartic equation // International Journal of Pure and Applied Mathematics. — 2004. — Vol. 14, No. 2. — P. 207—210.
4. **Зубов Л. М., Рудев А. Н.** Критерий сильной эллиптичности уравнений равновесия анизотропного линейно-упругого материала // Прикладная математика и механика. — 2016. — Т. 80, Вып. 6. — С. 686—721.
5. **Акуленко Л. Д., Нестеров С. В.** Изгибные колебания движущегося стержня // Прикладная математика и механика. — 2008. — Т. 72, Вып. 5. — С. 759—774.
6. **Акуленко Л. Д., Георгиевский Д. В., Нестеров С. В.** Спектр колебаний участка движущегося стержня при воздействии продольной нагрузки // МТТ. — 2015. — № 2. — С. 139—144.
7. **Ватульян А. О., Коссович Е. Л., Плотников Д. К.** О некоторых особенностях индентирования трещиноватых слоистых структур // МТТ. — 2017. — № 4. — С. 94—100.
8. **Vasil'ev V. V.** To the Problem of Stability of a Cylindrical Shell under Axial Compression // Mechanics of Solids. — 2011. — Vol. 46, No. 2. — P. 161—169.
9. **Ильгамов М. А.** Влияние давления окружающей среды на изгиб тонкой пластины и пленки // ДАН. — 2017. — Т. 476, № 4. — С. 482—721.
10. **Астапов И. С., Астапов Н. С.** Решение алгебраических уравнений третьей и четвертой степеней с помощью компьютерной алгебры // Программная инженерия. — 2014. — № 10. — С. 33—42.
11. **Астапов И. С., Астапов Н. С.** Алгоритмы символьного решения алгебраических уравнений // Программная инженерия. — 2017. — Т. 8, № 9. — С. 422—432.
12. **Клейн Ф.** Элементарная математика с точки зрения высшей. Т. 1. — М.: Наука, 1987. — 432 с.
13. **Клейн Ф.** Лекции об икосаэдре и решении уравнений пятой степени. — М.: Наука, 1989. — 336 с.

Algorithms for Factorization of Polynomials of Low Degree

N. S. Astapov, e-mail: nika@hydro.nsc.ru, Lavrentyev Institute of Hudrodynamics of Siberian Branch of Russian Academy of Sciences, Novosibirsk, 630090, Russian Federation, Novosibirsk State University, Novosibirsk-90, 630090, Russian Federation

Corresponding author:

Astapov Nikolay S., Senior Researcher, Lavrentyev Institute of Hudrodynamics SB RAS, Novosibirsk, 630090, Russian Federation
E-mail: nika@hydro.nsc.ru

Received on March 03, 2021
Accepted on March 29, 2021

For polynomials of the third degree of a special type, expansions into linear factors are found. Various methods of factorization of fourth-degree polynomials of general and particular types are proposed. For polynomials of the sixth degree of a special kind, representations are given in the form of a product of polynomials of lower degrees. Special attention is paid to representations through square trinomials.

The decomposition of the generalized reciprocal polynomial of the sixth degree into square trinomials is given.

Keywords: *software, polynomial multipliers, resolvent, reciprocal polynomials*

For citation:

Astapov N. S. Algorithms for Factorization of Polynomials of Low Degree, *Programmnaya Ingeneria*, 2021, vol. 12, no. 4, pp. 200—208.

DOI: 10.17587/prin.12.200-208

References

1. **Kutishchev G. P.** *Solution of algebraic equations of arbitrary degree: theory, methods, algorithms*, Moscow, URSS, 2010, 232 p. (in Russian).
2. **Gashkov S. B.** On the complexity of the integration of rational fractions, *Trudy MIAN*, 1997, vol. 218, pp. 122—133 (in Russian).
3. **Mochimaru Y.** Reciprocal solution of a quartic equation, *International Journal of Pure and Applied Mathematics*, 2004, vol. 14, no. 2, pp. 207—210.
4. **Zubov L. M., Rudev A. N.** A criterion for the strong ellipticity of the equilibrium equations for an anisotropic linear elastic material, *Prikladnaya matematika i mekhanika*, 2016, vol. 80, issue 6, pp. 686—721 (in Russian).
5. **Akulenko L. D., Nesterov S. V.** Bending vibrations of a moving rod, *Prikladnaya matematika i mekhanika*, 2008, vol. 72, issue 5, pp. 759—774 (in Russian).
6. **Akulenko L. D., Georgievsky D. V., Nesterov S. V.** Vibration spectrum of a section of a moving rod under the action of a longitudinal load, *MTT*, 2015, no. 2, pp. 139—144 (in Russian).
7. **Vatul'yan A. O., Kossovich Ye. L., Plotnikov D. K.** On some features of indentation of fractured layered structures, *MTT*, 2017, no. 4, pp. 94—100 (in Russian).
8. **Vasil'ev V. V.** To the Problem of Stability of a Cylindrical Shell under Axial Compression, *Mechanics of Solids*, 2011, vol. 46, no. 2, pp. 161—169.
9. **Il'gamov M. A.** Effect of ambient pressure on the bending of a thin plate and film, *DAN*, 2017, vol. 476, no. 4, pp. 482—721 (in Russian).
10. **Astapov I. S., Astapov N. S.** Solving third- and fourth-order algebraic equations by methods of competer algebra, *Programmnaya ingeneria*, 2014, no. 10, pp. 33—42 (in Russian).
11. **Astapov I. S., Astapov N. S.** Algorithms for Symbolic Solving of Algebraic Equations, *Programmnaya ingeneria*, 2017, vol. 8, no. 9, pp. 422—432 (in Russian).
12. **Klejn F.** *Elementary mathematics from a higher point of view.* Vol. 1, Moscow, Nauka, 1987, 432 p. (in Russian).
13. **Klejn F.** *Lectures on the icosahedron and the solution of equations of the fifth degree*, Moscow, Nauka, 1989, 336 p. (in Russian).

УДК 004.5 DOI: 10.17587/prin.12.209-215

Д. И. Читалов, мл. науч. сотр., cdi9@yandex.ru, Федеральное государственное бюджетное учреждение науки Южно-Уральский федеральный научный центр минералогии и геоэкологии Уральского отделения Российской академии наук, Россия, Челябинская обл., г. Миасс, Ильменский заповедник

О разработке модуля для модификации расчетных сеток посредством утилиты dsmcInitialise программной среды OpenFOAM

Исследование, результаты которого представлены в настоящей статье, посвящено разработке программного модуля с графическим пользовательским интерфейсом, обеспечивающего модификацию вычислительной сетки на базе утилиты dsmcInitialise, применяющейся на этапе препроцессинга численного моделирования задач механики сплошных сред с помощью программной среды OpenFOAM. Статья включает диаграммы структуры и логики функционирования приложения, описан используемый стек технологий. Приведены результаты применения программы в процессе одного из численных экспериментов на примере учебной задачи OpenFOAM. Сформулированы итоговые выводы, определена предполагаемая практическая ценность исследования.

Ключевые слова: *численное моделирование, механика сплошных сред, графический интерфейс пользователя, OpenFOAM, язык программирования Python, открытое программное обеспечение, утилита dsmcInitialise, библиотека PyQt, СУБД SQLite*

Введение

Настоящая статья продолжает цикл работ по созданию графической оболочки для консольной программной среды OpenFOAM [1]. Данная среда активно используется для численного моделирования задач механики сплошных сред на предприятиях таких отраслей машиностроения, как автомобилестроение, ракетно-космическое строение, судостроение, станкостроение и др. Популярность этой программной среды обусловлена следующими факторами: ее функциональные возможности позволяют создавать численные модели объектов и процессов в большинстве областей механики сплошных сред; открытый исходный код расширяет возможности программной среды, например, в части разработки новых программ-решателей, а также в направлении создания и подключения дополнительных программных приложений, в частности, графических интерфейсов.

Базовая версия графической оболочки представлена автором в 2016 г. и предусматривает возможность работы с несколькими популярными программами-решателями OpenFOAM, например, с rhoCentralFoam [2]. В последующие годы исследования в данном направлении были продолжены. Они коснулись прежде всего этапа препроцессинга, на котором определяются исходные условия численного эксперимента, генерируются и настраиваются расчетные сетки. Пользователи графической оболочки, созданной автором, получили возможность управления процессом подготовки расчетных сеток посредством привычного графического интерфейса с экранными формами [3—5]. Основными утилитами генерации расчетных сеток являются blockMesh, snappyHexMesh, foamyQuadMesh. Для работы с каждой утилитой разработан и интегрирован в базовую версию графической оболочки соответствующий программный модуль. Модифицированная версия приложения тестируется специалистами АО ГРЦ им. Макеева [6] при работе над проектами ракетно-космической тематики. Исходный код приложения размещен в свободном доступе на сервисе хостинга IT-проектов GitHub [7].

Цель разработки, результаты которой представлены в настоящей статье, заключается в том, чтобы максимально приблизить функции предложенной графической оболочки [2] к функциональным возможностям существующих программ-аналогов, таких как Salome [8], Helyx-OS [9], Visual-CFD [10]. При этом планировалось устранить недостатки перечисленных программных решений, а именно — подготовить подробную русскоязычную документацию, избавить пользователей от необходимости приобретения лицензии и оплаты услуг технической поддержки.

Результаты расширения исходного кода среды OpenFOAM, которые представлены в статье, связаны с разработкой модуля для работы с утилитой dsmcInitialise. Она является одной из утилит для препроцессинга численного эксперимента, а именно —

для модификации расчетной сетки в целях учета в итоговой численной модели большего числа параметров моделируемой задачи механики сплошных сред. От препроцессинга во многом зависит результат эксперимента и степень его соответствия реальному объекту или процессу. Подробнее назначение данной утилиты и особенности ее применения будут представлены далее.

Рассматриваемый в статье модуль позволит пользователю отказаться от трудоемкого и приводящего к возникновению ошибок использования командной строки, когда и создание необходимых служебных файлов-словарей расчетного случая, и заполнение их расчетными параметрами, и запуск утилит осуществляются вручную, через командную строку. В силу отсутствия какого-либо программного контроля над этим процессом и валидации вводимых параметров повышается вероятность допустить ошибку и сформировать итоговую численную модель с большими отклонениями от реальной ситуации и заявленных требований. Модульный принцип реализации предложенного автором приложения [2] повышает гибкость всей системы и позволяет разрабатывать и интегрировать новые компоненты.

Назначение утилиты dsmcInitialise

Утилита dsmcInitialise — это программа, интегрированная в программную среду OpenFOAM и применяемая на этапе препроцессинга численного моделирования задач механики сплошных сред. Утилита отвечает за корректировку геометрии расчетной области путем формирования начальных конфигураций частиц при проведении моделирования методом Монте-Карло. Скорректированная таким образом расчетная область используется в рамках экспериментов по моделированию задач механики сплошных сред на базе программы-решателя dsmcFoam среды OpenFOAM.

Под методом прямого моделирования Монте-Карло (*Direct Simulation Monte Carlo* — DSMC) понимается подход, основанный на стохастических частицах и предназначенный для численного исследования потока разреженного газа. Метод предложен профессором Г. Бердом [11] и считается одним из наиболее популярных решений для моделирования задач газовых потоков в неравновесном режиме числа Кнудсена. В методе DSMC одна частица включает большое число атомов или молекул реального газа, что обеспечивает снижение вычислительных затрат по сравнению с полностью детерминированным подходом, например, молекулярной динамикой. Каждая из указанных частиц может свободно перемещаться в пространстве в соответствии с установленной скоростью, а также взаимодействовать с другими частицами. Столкновения частиц обрабатываются стохастически после того как завершились все движения частиц. Можно говорить об имитации физики реального газа, а не о попытке решить уравнения движения Ньютона для большого числа отдельных атомов и молекул [12].

В основе программы-решателя dsmcFoam лежит механизм построения численных моделей методом Монте-Карло для исследований в области динамики разреженного газа. Программа dsmcFoam представляет собой шаговый решатель, моделирующий процессы со стохастическими столкновениями молекул, разработанный для исследования проблем потока разреженного газа. При работе с указанным решателем осуществляется декомпозиция расчетной сетки в соответствии с числом процессорных ядер, на которых моделируется задача.

При постановке экспериментов на базе решателя dsmcFoam важная роль отводится двум шагам, а именно — генерации расчетной сетки и инициализации частиц. Первый шаг осуществляется с помощью традиционных сеточных утилит blockMesh, snappyHexMesh, foamyQuadMesh. Важное значение имеет шаг инициализации частиц, на который ориентирована утилита dsmcInitialise.

При проведении численных исследований в области динамики разреженного газа специалист выполняет следующие действия:

1) генерацию вычислительной сетки с помощью одной из стандартных утилит;

2) определение граничных условий для выбранной задачи механики сплошных сред;

3) заполнение вычислительной сетки частицами с помощью утилиты dsmcInitialise;

4) запуск процесса численного моделирования на базе утилиты dsmcFoam;

5) визуализацию результатов с помощью пакета постобработки ParaView [13].

Управление утилитой dsmcInitialise осуществляется на основе параметров, задаваемых через файл-словарь dsmcInitialiseDict (см. таблицу). Указанные параметры определяют характер распределения частиц в граничной области. Специалисту при этом следует задать температуру, скорость и числовую плотность.

Параметры файла-словаря dsmcInitialiseDict

Параметр		Описание	Пример
numberDensities	N_2	Числовая плотность частиц N_2 в идеальном газе	0,777e20
	O_2	Числовая плотность частиц O_2 в идеальном газе	0,223e20
temperature		Температура в расчетной области, K	300
velocity		Вектор скорости	(1950 0 0)

Постановка цели и задач разработки

Обсуждаемое в рамках настоящей статьи программное решение направлено на обеспечение возможности пользователя централизованно работать с утилитой dsmcInitialise в рамках проведения этапа препроцессинга численного моделирования задач механики сплошных сред на базе среды OpenFOAM. Предполагается разработка программного модуля, включающего в себя исполняемые скрипты и элементы интерфейса, а также их интеграция в исходный код графической оболочки, разработанной автором [2]. Такой подход позволит расширить возможности специалистов-экспериментаторов и расширит границы применения приложения, представленного в работе [2], в рамках численных экспериментов.

Особенности работы утилиты dsmcInitialise и структура файла-словаря dsmcInitialiseDict исследованы автором с помощью обучающих материалов пользователя среды OpenFOAM [14, 15]. Эффективное использование утилиты зависит от правильности формирования указанного файла-словаря, при этом логика формирования закладывается в скрипты разрабатываемого модуля. К параметрам, задаваемым в файле, относятся: числовая плотность частиц; температура; давление в расчетной области. Для элементов управления экранных форм интерфейса требуется определение валидаторов для защиты от неверных действий пользователя.

Предложенное программное приложение функционирует не как независимый продукт, а дополняет исходный код уже предложенного программного средства [2] и интегрируется в него на основе модульного принципа. Пользователь обновленной версии приложения имеет возможность выполнять численное моделирование в прежнем формате, однако теперь к доступным опциям препроцессинга добавляется утилита dsmcInitialise. У пользователя сохраняется возможность выбора существующего расчетного случая для редактирования параметров либо создания нового. Доработка существующей версии приложения [2] потребовала выполнения перечисленных далее задач.

1. Проектирование макета экранной формы, выполняющей роль связки пользователя с файлом-словарем dsmcInitialiseDict.

2. Разработка и реализация механизма записи параметров из экранной формы в указанный файл.

3. Расширение инструментальной панели интерфейса за счет добавления кнопки открытия реализованной экранной формы.

4. Реализация механизма вывода параметров файла-словаря dsmcInitialiseDict в окно визуализации результатов, встроенного в интерфейс.

5. Разработка и внедрение механизма сериализации параметров файла-словаря dsmcInitialiseDict, задаваемых через соответствующую экранную форму, и их дальнейшего восстановления для обеспечения возможности внесения корректировок.

6. Реализация набора валидаторов для элементов управления экранной формы в целях обеспечения контроля корректности типов и формата вводимых параметров.

Средства разработки

Далее приведены компоненты модуля и применяемые для их реализации программные механизмы, которые имеют статус open-source-проектов и доступны для разработчиков без каких-либо ограничений.

- **Бэкенд.** Это набор программных средств (скриптов), отвечающих за реализацию логики работы приложения, т. е. часть программы, скрытая от пользователя. В данном случае бэкенд реализован на базе языка программирования Python 3.7 [16], который по состоянию на сентябрь 2020 г. занимал третью позицию рейтинга TIOBE [17].
- **Фронтенд.** Визуализируемая часть приложения (интерфейс). Это звено, связывающее пользователя с логикой программы (бэкендом) и предназначенное для визуализации элементов управления (экранных форм, кнопок, надписей и т. д.) и отображения результатов взаимодействия пользователя с программой. Интерфейс рассматриваемого модуля реализуется на базе фреймворка PyQt5 — популярного инструментального средства для разработки интерфейсов десктопных программ [18, 19].
- **Система хранения данных.** Предложено использование реляционной СУБД SQLite [20], ORM-подхода для выполнения запросов и библиотеки SQLAlchemy [21]. Она предполагает выполнение команд на базе привычного Python-синтаксиса вместо SQL-запросов.

Программный код модуля подготовлен (написан, протестирован и отлажен) посредством интегрированной среды разработки PyCharm. Она предусматривает возможность создания виртуального окружения для проекта, подсветку кода, средства приведения кода к стандарту PEP8.

Структура и логика работы модуля

На рис. 1 приведена структурная диаграмма, описывающая взаимосвязь компонентов графической оболочки [2] после интеграции программного модуля, обеспечивающего работу с утилитой dsmcInitialise в рамках численного моделирования задач механики сплошных сред.

Скрипты представленного программного модуля находятся в четырех директориях папки-корня приложения:

- директория windows для файлов исходного кода, определяющих графическую составляющую программы, в которых содержатся команды, отвечающие за визуализацию элементов окон-виджетов интерфейса;
- директория forms содержит файлы с экранными компонентами, соответствующими параметрам файлов-словарей расчетного случая;
- файлы директории threads включают программный код, обеспечивающий многопоточность запуска утилит OpenFOAM;
- файлы директории functions содержат программный код служебных функций приложения;
- главный запускаемый файл run.py приложения отвечает за визуализацию интерфейса.

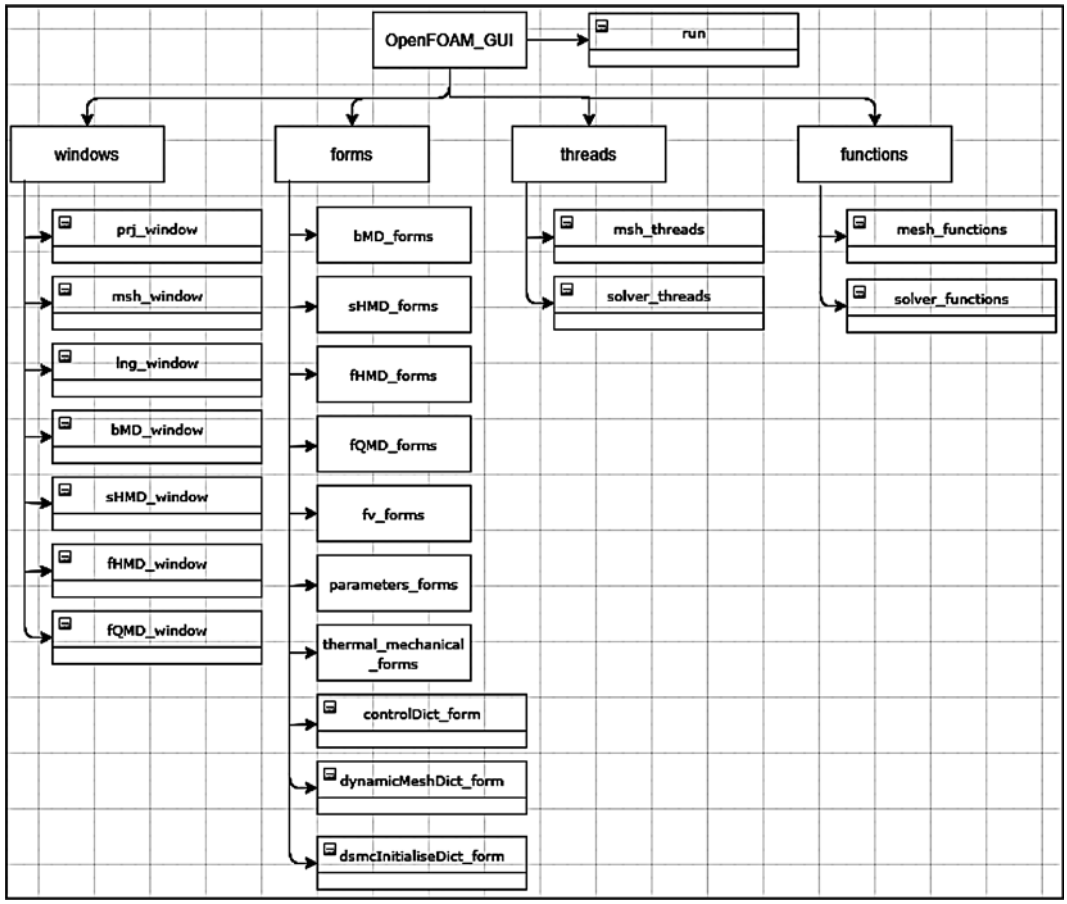


Рис. 1. Структура графической оболочки OpenFOAM_GUI

На рис. 2 приведена диаграмма процессов, отражающая логику работы пользователя с приложением OpenFOAM_GUI в контексте применения утилиты dsmcInitialise.

Модуль для модификации расчетных сеток на базе утилиты dsmcInitialise применяется на этапе препроцессинга численного эксперимента после генерации базовой сеточной модели. Работа с утилитой осуществляется в тех случаях, когда требуется скорректировать расчетную сетку с учетом дополнительных требований задачи механики сплошных сред в области газовой динамики. Запуск утилиты проводится только после построения начальной расчетной сетки, что контролируется реализованными в приложении OpenFOAM_GUI алгоритмами. Исходная сеточная модель формируется традиционными сеточными утилитами, такими как blockMesh, snappyHexMesh, foamyHexMesh, foamyQuadMesh [3—5].

Контроль корректности сгенерированной расчетной сетки выполняется посредством ее визуализации с помощью пакета ParaView. При соответствии расчетной сетки заявленным требованиям эксперимента задача специалиста — переход к следующим шагам препроцессинга или к этапу моделирования задачи механики сплошных сред на базе одного из встроенных решателей. В случае, если расчетная сетка отклоняется от установленных требований эксперимента, возможен вариант возврата к редактированию ее параметров. После этого вновь может быть применена любая из утилит модификации и визуализации сеточной модели.

На рис. 3 (см. третью сторону обложки) представлено главное окно графической оболочки по итогам генерации базовой сетки и ее модификации с помощью утилиты dsmcInitialise и окно пакета ParaView. Возможности графической оболочки протестированы на примере одной из учебных задач репозитория программной среды OpenFOAM. Это задача freeSpacePeriodic, которая моделируется с помощью программы-решателя dsmcFoam и позволяет анализировать динамику разреженного газа.

Результаты исследования

По итогам проведенной автором работы расширен исходный код графической оболочки OpenFOAM_GUI. Список доступных пользователю программных модулей дополнен компонентом, обеспечивающим проведение одного из этапов препроцессинга численного эксперимента — модификации расчетной сетки с помощью утилиты dsmcInitialise. Указанная утилита может применяться ко всему перечню задач механики сплошных сред, решаемых с помощью пакета OpenFOAM. Поэтому пользователями представленного программного модуля могут быть спе-

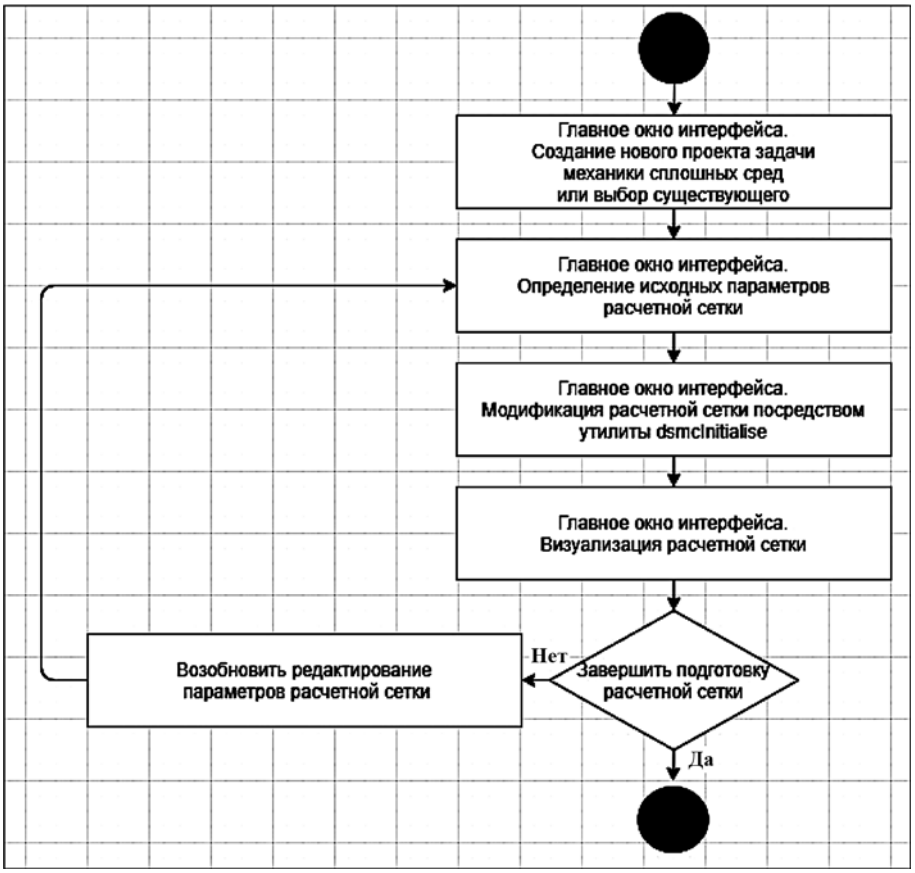


Рис. 2. Логика работы графической оболочки OpenFOAM_GUI в части модификации расчетных сеток с помощью утилиты dsmcInitialiseDict

циалисты различных предприятий, где используется OpenFOAM.

В результате решения сформулированных автором задач выполнена модификация главного окна графической оболочки: добавлены необходимые окна-виджеты, элементы управления, реализованы необходимые программные скрипты, добавлены валидаторы вводимых пользователем данных. Благодаря этому пользователь получает возможность работы с файлом-словарем dsmcInitialiseDict и утилитой dsmcInitialise посредством привычного оконного интерфейса.

В работе представлены диаграммы, визуализирующие структуру расширенной версии графической оболочки OpenFOAM_GUI и логику работы пользователя с утилитой dsmcInitialise. Функциональные возможности разработанного модуля протестированы на примере одного из учебных проектов OpenFOAM. Предполагается дальнейшее тестирование модуля в рамках моделирования задач механики сплошных сред специалистами АО ГРЦ им. Макеева при работе над проектами ракетно-космической отрасли.

Заключение

В настоящей статье представлены результаты разработки программного модуля с графическим интерфейсом для обеспечения работы пользователя с утилитой dsmcInitialise, которая применяется в рамках численного моделирования задач механики сплошных сред с помощью программной среды OpenFOAM для заполнения сеточной модели DSMC-частицами.

Автором разработан программный модуль, обеспечивающий централизованную (через главное окно интерфейса) работу специалиста с утилитой dsmcInitialise и с соответствующим файлом-словарем dsmcInitialiseDict и интегрированный в графическую оболочку [2]. Модифицированы графическая и программная составляющие графической оболочки посредством реализации перечисленных далее компонентов:

- экранная форма-виджет, соответствующая файлу-словарю dsmcInitialiseDict с элементами управления, обеспечивающими возможность определения параметров заполнения вычислительной сетки DSMC-частицами;
- алгоритм и программный скрипт генерации и выполнения bash-скрипта, отвечающего за запуск утилиты dsmcInitialise;
- валидаторы для контроля корректности типа и формата значений, определяемых через виджет-форму для словаря dsmcInitialiseDict;
- алгоритм и программный скрипт для сериализации и последующего восстановления параметров файла dsmcInitialiseDict;

• алгоритм и программный скрипт подготовки нескольких вариантов файла-словаря dsmcInitialiseDict для одного и того же проекта задачи механики сплошных сред, что позволяет выполнять моделирование на различных исходных параметрах.

Потенциальная практическая ценность исследования выражается в том, что для пользователя реализованный модуль может обеспечить экономию рабочего времени при формировании файла-словаря dsmcInitialiseDict и модификации расчетных сеток с помощью утилиты dsmcInitialise. Кроме того, возможна минимизация ошибок этапа препроцессинга численного эксперимента. Графический интерфейс позволяет заменить использование командной строки на работу с привычными экранными формами. Это ускоряет и упрощает проведение численного моделирования на базе программной среды OpenFOAM. Валидаторы, предложенные автором, минимизируют вероятность возникновения ошибок при вводе типов данных на этапе подготовки сеточной модели.

Список литературы

1. **OpenFOAM.** The open source CFD toolbox. URL: https://www.openfoam.com/ (дата обращения: 07.12.2020).
2. **Читалов Д. И., Меркулов Е. С., Калашников С. Т.** Разработка графического интерфейса пользователя для программного комплекса OpenFOAM // Программная инженерия. — 2016. — Т. 7, № 12. — С. 568—574. DOI: 10.17587/prin.7.568-574.
3. **Читалов Д. И., Калашников С. Т.** Разработка приложения для подготовки расчетных сеток с градуирующими и изогнутыми краями для программной среды OpenFOAM // Системы и средства информатики. — 2018. — Т. 28, № 4. — С. 122—135. DOI: 10.14357/08696527180412.
4. **Читалов Д. И., Калашников С. Т.** Разработка приложения для подготовки расчетных сеток посредством утилиты snappyHexMesh программной среды OpenFOAM // Программные продукты и системы. — 2018. — Т. 31, № 4. — С. 715—722. DOI: 10.15827/0236-235X.124.715-722.
5. **Читалов Д. И., Калашников С. Т.** Разработка приложения для подготовки расчетных сеток с помощью утилиты foamyQuadMesh платформы OpenFOAM // Программная инженерия. — 2018. — Т. 9, № 7. — С. 311—317. DOI: 10.17587/prin.9.311-317.
6. **АО** "Государственный ракетный центр имени академика В. П. Макеева". URL: http://www.makeyev.ru/ (дата обращения: 07.12.2020).
7. **OpenFOAM_GUI.** URL: http://github.com/DmitryChitalov/OpenFOAM_GUI (дата обращения: 07.12.2020).
8. **Salome.** The Open Source integration Platform for Numerical Simulation. URL: https://salome-platform.org/ (дата обращения: 07.12.2020).
9. **Helyx-OS.** Open-Source GUI for OpenFOAM. URL: https://engys.com/products/helyx-os (дата обращения: 07.12.2020).
10. **Visual-CFD.** URL: https://www.esi-group.com/products/computational-fluid-dynamics (дата обращения: 07.12.2020).
11. **Берд Г.** Молекулярная газовая динамика. — М.: Мир, 1981. — 319 с.
12. **White C., Borg M., Scanlon T. J.** at al. Jason.dsmcFoam+: An OpenFOAM based direct simulation Monte Carlo solver// Computer Physics Communications. — 2018. — Vol. 224. — P. 22—43.
13. **ParaView.** URL: https://www.paraview.org/ (дата обращения: 07.12.2020).
14. **OpenFOAM.** User Guide. URL: http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf (дата обращения: 07.12.2020).
15. **OpenFOAM.** Tutorial Guide. URL: http://openfoam.com/documentation/tutorial-guide/index.php (дата обращения: 07.12.2020).
16. **Python 3.7** documentation. URL: https://docs.python.org/3.3/ (дата обращения: 07.12.2020).
17. **TIOBE** Index. URL: http://www.tiobe.com/tiobe-index/ (дата обращения: 07.12.2020).
18. **PyQt5** Reference Guide. URL: http://pyqt.sourceforge.net/Docs/PyQt5/ (дата обращения: 07.12.2020).
19. **Прохоренок Н. А.** Python 3 и PyQt. Разработка приложений. — СПб.: БХВ-Петербург, 2012. — 704 с.
20. **SQLite.** URL: https://www.sqlite.org/index.html (дата обращения: 07.12.2020).
21. **SQLAlchemy.** URL: https://www.sqlalchemy.org/ (дата обращения: 07.12.2020).

On the Development of a Module for the Modification of Computational Meshes by the dsmcInitialise Utility

D. I. Chitalov, cdi9@yandex.ru, South Urals Federal Research Centre of Mineralogy and Geoecology of the UB RAS, Chelyabinsk Region, Miass, Ilmen reserve, 456317, Russian Federation

Corresponding author:

Chitalov Dmitry I., Junior Researcher, South Urals Federal Research Centre of Mineralogy and Geoecology of the UB RAS, Chelyabinsk Region, Miass, Ilmen reserve, 456317, Russian Federation
E-mail: cdi9@yandex.ru

Received on January 22, 2021
Accepted on March 23, 2021

The research, the results of which are presented in this article, is devoted to the development of a software module with a graphical user interface that provides a modification of the computational mesh based on the dsmcInitialise utility, which is used at the preprocessing stage of numerical modeling of continuum mechanics problems using the OpenFOAM software environment. The paper describes the existing graphical shells for working with OpenFOAM with an indication of their shortcomings, formulates the relevance of the work, and defines the goals and objectives of the study. The article presents the features of the direct Monte Carlo simulation method, a description of the dsmcInitialise utility integrated into OpenFOAM and designed for such modeling, as well as a description of the corresponding dictionary file with parameters. The article includes diagrams of the structure and logic of the applica-

tion, describes the technology stack used. The results of the application of the program on the example of one of the training problem of OpenFOAM are presented. The final conclusions are formulated, as well as the provisions that determine the scientific novelty of the research, and its intended practical value is determined. A link to the repository with the source code of the presented software module is provided.

Keywords: *numerical simulation, continuum mechanics, graphical user interface, OpenFOAM, Python, open source software, dsmcInitialise utility, PyQt, SQLite*

For citation:

Chitalov D. I. On the Development of a Module for the Modification of Computational Meshes by the dsmcInitialise Utility, *Programmnaya Ingeneria,* 2021, vol. 12, no. 4, pp. 209—215.

DOI: 10.17587/prin.12.209-215

References

1. **OpenFOAM.** The open source CFD toolbox, available at: https://www.openfoam.com/
2. **Chitalov D. I., Merkulov E. S., Kalashnikov S. T.** Development of a graphical user interface for the OpenFOAM toolbox, *Programmnaya ingeneria*, 2016, vol. 7, no. 12, pp. 568—574 (in Russian).
3. **Chitalov D. I., Kalashnikov S. T.** Development of an application for the preparation of computational meshes with graduating and curved edges for the OpenFOAM software, *Systems and Means of Informatics*, 2018, vol. 28, no. 4, pp. 122—135 (in Russian).
4. **Chitalov D. I., Kalashnikov S. T.** Development of an application for preparing calculation grids using the snappyHexMesh utility of the OpenFOAM software environment, *Software products and systems*, 2018, vol. 31, no. 4, pp. 715—722 (in Russian).
5. **Chitalov D. I., Kalashnikov S. T.** Application development for meshes preparation using foamyQuadMesh utility for the OpenFOAM toolbox, *Programmnaya ingeneria*, 2018, vol. 9, no. 7, pp. 311—317 (in Russian).
6. **Makeyev** SRC, available at: http://www.makeyev.ru/
7. **OpenFOAM_GUI,** available at: http://github.com/Dmitry-Chitalov/OpenFOAM_GUI
8. **Salome.** The Open Source Integration Platform for Numerical Simulation, available at: http://www.salome-platform.org
9. **Helyx-OS.** Open Source GUI for OpenFOAM, available at: http://engys.com/products/helyx-os
10. **Visual-CFD** for OpenFOAM. CFD simulation software aimed at solving complex flow applications, available at: http://www.esi-group.com/software-solutions/virtual-environment/cfd-multiphysics/visual-cfd-openfoam
11. **Bird G.** *Molecular gas dynamics,* Moscow, Mir Publishers, 1981, 319 p. (in Russian).
12. **White C., Borg M., Scanlon T. J., Longshaw St., John B., Emerson D. R., Reese J.** dsmcFoam +: An OpenFOAM based direct simulation Monte Carlo solver., *Computer Physics Communications*, 2018, vol. 224, pp. 22—43.
13. **ParaView.** Open-source, multi-platform data analysis and visualization application, available at: http://www.paraview.org/
14. **The OpenFOAM** Foundation. User Guide, available at: http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf
15. **OpenFOAM.** Tutorial Guide, available at: http://openfoam.com/documentation/tutorial-guide/index.php
16. **Python 3.7** documentation, available at: http://docs.python.org/3.7/
17. **Tiobe** Index, available at: http://www.tiobe.com/tiobe-index/
18. **PyQt5** Reference Guide, available at: http://pyqt.sourceforge.net/Docs/PyQt5/
19. **Prohorenok N. A.** *Python 3 and PyQt. Application Development*, St. Petersburg, BHV-Petersburg, 2012, 704 p. (in Russian).
20. **SQLite,** available at: https://www.sqlite.org/index.html
21. **SQLAlchemy,** available at: https://www.sqlalchemy.org/

ИНФОРМАЦИЯ

Международная научная конференция "Суперкомпьютерные дни в России" 27—28 сентября 2021 г.

Тематика конференции охватывает следующие основные направления:

- Проблемы создания экзафлопсных суперкомпьютеров: архитектура, программирование, сопровождение
- Суперкомпьютерные технологии в промышленности
- Конвергенция высокопроизводительных вычислений, машинного обучения и технологий больших данных: теория, практика, перспективы, истории успеха
- Перспективные модели, языки и технологии параллельного программирования
- Теория и практика решения больших и сверхбольших задач
- Эффективность и масштабируемость параллельных программ и вычислительных систем
- Новые принципы организации высокопроизводительных вычислений. Нетрадиционные архитектуры вычислительных систем
- Суперкомпьютерные технологии и защита информации
- Технологии распределенных вычислений и распределенной обработки данных, Grid-технологии, облачные технологии
- Большие данные: хранение, обработка, аналитика
- Визуализация в суперкомпьютерном мире: методы, технологии и системы
- Суперкомпьютерное образование

Подробности: **http://russianscdays.org/**

УДК 004.421 DOI: 10.17587/prin.12.216-222

Н. К. Петрова[1,2], канд. физ.-мат. наук, доц., nk_petrova@mail.ru,
А. П. Мухачев[1], студент, houstondevs@gmail.com,
А. А. Загидуллин[2], аспирант, arhtur.zagidullin@ya.ru,
С. М. Куценко[1], канд. пед. наук, доц., s.koutsenko@mail.ru,
[1] Казанский государственный энергетический университет,
[2] Казанский (Приволжский) Федеральный университет

Реализация электронного курса по программированию на языке Python для платформы Android

Представлено описание и принципы разработки мобильного приложения для платформы Android, обеспечивающего свободный доступ к электронному курсу по обучению базовым структурам языка Python и построению на их основе шаблонных алгоритмов программирования. Содержание курса разработано на основе сравнительного анализа Python с языком C++. Одной из целей такого подхода является разделение задач, для решения которых эффективнее применять интерпретируемый язык Python, либо компилируемый язык C++. Разработанное приложение является бесплатным, логически целостным, допускает возможность дополнения его новыми элементами — примерами, типами алгоритмов.

Ключевые слова: *обучение алгоритмизации, основы программирования, Python, платформа, фреймворк, мобильное приложение, сервер*

Введение

Развитие современного государства, в том числе и России, как информационного общества предполагает массовое внедрение информационных технологий (ИТ) во все сферы деятельности социума, и в первую очередь в сферу образования. Особенностью настоящего времени является тот факт, что почти в каждой семье имеется как минимум один, а чаще и несколько настольных компьютеров или ноутбуков, и практически у каждого члена семьи имеется смартфон или планшет. На этом фоне помимо возрастания интереса к компьютерным играм, общению в соцсетях, дистанционному использованию различных сервисных служб большую популярность набирают различные системы онлайн-обучения [1]. Главное преимущество таких систем в отличие от очных курсов в том, что они позволяют получать новые знания по собственному графику, без привязки к группе, времени и месту занятия.

Существует большое число мобильных приложений для предоставления доступа к онлайн-курсам по различным тематикам. Не стало исключением и программирование, так как сейчас это очень востребованная и высокооплачиваемая отрасль.

Если обратиться к сервису GooglePlay, то можно найти тысячи различных приложений, как развлекательных, так и оказывающих различные услуги. Если отфильтровать их по теме "Программирование", поиск выдает тысячи различных сервисов, предоставляющих возможность получить знания об основах программирования.

Ниже представлены некоторые важные критерии, по которым целесообразно выбирать приложение для обучения программированию [2, 3]:

- *компактность* — компоненты дистанционного обучения должны быть короткими по продолжительности, учитывая то, что они доступны в среде, в которой вероятны потенциальные перерывы в связи — этот критерий важен для тех, кто не хочет инсталлировать приложение на свое устройство;
- *высокий уровень доступности мультимедиа* — высокое качество изображения/звука при малом размере выходных файлов для ускорения загрузки;
- *доступность* — полный и бесплатный доступ к материалам курса;
- *содержание курса* — методологически выверенная структура и логическая завершенность;
- *отвлекающие факторы* — отсутствие любых видов рекламы и др.

Авторами разработано мобильное приложение для платформы Android, которое позволяет получить свободный электронный доступ к курсу, включающему обучение базовым понятиям, основным лексическим элементам, структурам языка Python. Перечисленные выше критерии учтены при разработке мобильного приложения. Также подготовлены методические материалы по разработке базовых алгоритмов средствами этого языка. Основной "мотив" для разработки описанного в статье курса — предоставить его пользователям, занимающимся самообразованием, полный доступ к интерактивным курсам. По этой причине у авторов нет планов делать это приложение платным. При этом разработчиками

гарантируется достоверность информации, внесенной в обучающее приложение, его методологически выверенная структура и логическая завершенность.

В рабочих программах дисциплины "Алгоритмизация и программирование" в технических вузах и особенно на специализированных ИТ-направлениях, как правило, предусматривается изучение языка C++. При этом язык Python многие студенты изучают или самостоятельно, или в рамках специальных дисциплин. Разработанное приложение позволит его пользователям, владеющим базовыми знаниями по языку C++, быстрее освоить более современный и более удобный язык программирования Python, используя представленный в методической части сравнительный анализ двух языков.

1. Содержательная часть курса

В содержании курса кроме перечисленных выше тем предусмотрен раздел, посвященный оценке достоинств и недостатков языка Python как интерпретируемого языка. Большая часть тем курса подается как результат сравнительного анализа с операторами и понятиями языка C++, который, как уже было сказано, согласно требованиям учебных программ высшего образования является обязательным для изучения на всех профилирующих ИТ-специальностях.

При разработке программ на любом языке могут использоваться как типовые (операторы if, while), так и уникальные языковые конструкции (такие как инкремент ++ /декремент --). В итоге, одна и та же задача может быть реализована по-разному, в зависимости от алгоритмического языка [4].

Наглядным примером сказанному является задача по созданию и инициализации одномерного массива x[n] значениями, вычисляемыми от x1 с шагом dx, и последующей печатью результата (см. таблицу).

Именно с этой позиции в предлагаемом онлайн-курсе подается сравнительный анализ языков Python и C++. Он показывает, в частности, что громоздкие языковые конструкции для реализации определенных алгоритмов на языке C++ можно в несколько строчек выразить на Python. Это позволит сэкономить время на разработку программ и на их реализацию. Результаты такого анализа будут способствовать расширению уровня знаний, предоставят возможность взглянуть на программный код с различных сторон. Они позволят оценить возможные достоинства и недостатки языков, установить полезное общее и не менее полезное различное в алгоритмизации с применением того или иного языка. Здесь имеется в виду, что изучение языка Python не означает автоматического отказа от более мощного транслятора C++. Да, Python гибче, это "человекоориентированный" язык, в то время как C++ язык машиноориентированный, он имеет сложный синтаксис, строгую типизацию, сложные отладку и поиск ошибок. При этом Python — интерпретатор и, следовательно, программа на нем работает медленнее, интерпретатор не оптимизирует код. Программы на C++ компилируются, и в результате создаются высокопроизводительные и отказоустойчивые приложения, которые имеют совместимость с аппаратным обеспечением. У C++ большое сообщество разработчиков. Но эти два языка можно "подружить": Python имеет встроенные механизмы для вызова программ на C++, что позволяет разделить алгоритм на программы пользовательского уровня, написанные на языке Python, и вычислительные блоки, реализованные на C++.

По представленным выше причинам немаловажным аспектом сравнительного анализа является оценка производительности рассматриваемых языков Python и C++. Программисту важно знать, как и за какое время выполняются одинаковые алгоритмы на разных языках. Пользователь приложения получит доступ к графикам и отчетам, по которым можно сделать определенные выводы и понять, для каких задач предпочтительнее использовать C++, а для каких — Python.

Программа, написанная на языке сценариев Python, уступает в скорости выполнения программе, реализованной на языке C++. Тем не менее достаточно часто важно не быстродействие программного продукта, а быстрота и эффективность его разработки. Именно эти факторы и следует учитывать, выбирая для программирования Python. С его помощью можно получить результат быстрее, сэкономив время на разработку программы.

Подобная ситуация возникает, например, в задачах компьютерного моделирования, когда программный продукт служит инструментом, формирующим итоговый алгоритм [5]. Для решения таких задач удобен, например, язык Visual Basic for Applications (VBA). Его механизмы позволяют обеспечивать прямой доступ к ячейкам листа Excel и работать одновременно с несколькими Excel-документами. При этом

C++	Python

int i=0; float x[n]; //Выделяем память под вещественный массив размера *n*
for(float x=x1; i<n; i++, x+=dx)
xx[i]=x; //Инициализируем каждый элемент массива в цикле по индексу *i*
	x = [x1 + dx*i for i in range(n)] // Создаем и инициализируем элементы массива с помощью функции-итератора range(*n*)

Вывод массива на консоль	
for(float x=x1; i<n; i++, x+=dx) cout<<xx[i]<<'\t';	print(xx)

нет необходимости заботиться о способах обмена данными между ними, что делает его незаменимым при выборе метода реализации моделируемого процесса или измерения. А встроенный Макрорекордер избавляет разработчика от программирования рутинных стандартных операций, записывая их в виде небольшой программы — макроса. На упрощенных расчетах формируется алгоритм моделируемого процесса, а для реальных трудоемких вычислений полученный алгоритм переводится на скоростные языки [6].

В Python доступен широкий спектр библиотечных функций. Как и в современных стандартах языка C++ при разработке программ на языке Python можно не тратить время на описание переменных, размера массива, заменять массивы кортежами, до минимума сокращать алгоритмы, аккумулируя в одном операторе целые фрагменты программ.

2. Разработка платформы для серверной и клиентской частей приложения

В разработанном авторами приложении серверная часть образовательной платформы реализуется на языке программирования Python с использованием фреймворков Django и Django Rest framework [7, 8].

Django считается лучшим веб-фреймворком, написанным на Python [9]. Этот инструмент удобно использовать для разработки веб-сервисов, которые работают с базами данных. Django написан на Python, он использует принцип DRY (*don't repeat yourself*), благодаря этому сокращается время написания системы.

Серверная часть системы является полноценным REST-сервисом (*Representational State Transfer* — передача состояния представления) — архитектурный стиль взаимодействия компонентов распределенной системы в компьютерной сети. Данный архитектурный стиль представляет собой согласованный набор ограничений, учитываемых при проектировании распределенной системы.

"Общение" между сервером и клиентом проходит http-запросами, в которых данные между системами будут передаваться в формате JSON (*JavaScript Object Notation*) — простого формата обмена данными, удобного для чтения и написания как человеком, так и компьютером. JSON является текстовым форматом данных, полностью независимым от языка реализации прикладных систем, а это очень удобно, поскольку клиентская и серверная части могут быть реализованы на разных языках программирования, большинство языков программирования позволяет работать со строками JSON. Кроме того, JSON использует соглашения, знакомые программистам C-подобных языков, таких как C, C++, C#, Java, JavaScript, Perl, Python и др. Эти свойства делают JSON идеальным языком обмена данными. JSON основан на двух структурах данных — коллекции пар ключ/значение и упорядоченном списке значений.

Для хранения данных на сервере используется объектно-реляционная система управления базами данных PostgreSQL [10]. Это очень гибкая и надежная СУБД, которая может хранить большое количество различных типов данных, включая сложные структуры данных. Этим критериям отвечают и другие СУБД, поэтому главным ее достоинством является свободная лицензия.

При разработке приложений для мобильных устройств особенно актуально стоит вопрос выбора платформы — разработка мобильных приложений iOS или Android. Не менее важен выбор технологии. Статистика StatCounterGlobalStats показывает, что в России 25 % пользователей предпочитают iOS и 74 % — Android. Принимая во внимание этот факт, мы выбрали в первую очередь операционную систему Android, чтобы охватить большее число пользователей.

При всем разнообразии доступных технологий выбор так или иначе сводится к одному из двух видов приложений [10]:

- нативные приложения — разрабатываются для использования на определенной платформе;
- гибридные приложения — сочетают свойства и нативных, и веб-приложений, работающих через веб-браузер на устройстве пользователя.

Рассмотрев большинство кроссплатформенных фреймворков, мы выбрали Flutter [11], так как с учетом всех его достоинств он больше всего подходит под решаемые нами задачи. Кроме того, у нас есть большой опыт работы с данной платформой в разработке мобильных приложений. В результате клиентская часть нашего приложения реализована на языке Dart с использованием фреймворка Flutter, который позволяет создавать производительные нативные прикладные программы для iOS и Android. Такие приложения разрабатываются для использования на определенной платформе или на определенном устройстве [12].

Реализованное разделение клиентской и серверной частей позволит в будущем без особого труда масштабировать платформу, внедряя новые функции в разработанную систему (рис. 1).

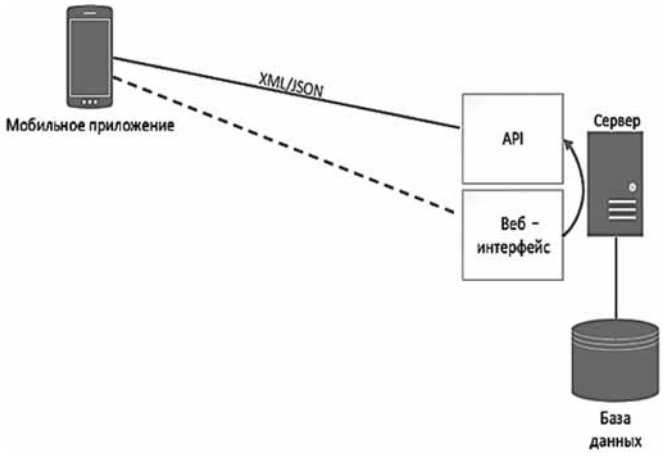


Рис. 1. Взаимодействие клиентской и серверной частей приложения

3. Описание пользовательского интерфейса приложения

Приложения, которые мы используем, имеют функциональные возможности, позволяющие пользователям регистрироваться (рис. 2) и авторизовываться (рис. 3) на платформе Android.

На рис. 4 представлено окно с кнопками навигации, позволяющими перейти к разделам курса, где можно выбрать тему для изучения и начать ознакомление.

Инициализация окна осуществляется через вкладку "Курсы" в нижней навигационной панели, окно при этом является частью алгоритмической структуры приложения. Из фрагмента кода, реализующего нижнюю навигационную панель (рис. 5), можно увидеть описание класса "CoursesScreen", который и является окном, принадлежащим вкладке "Курсы", и находится внутри реализации навигационной панели.

На рис. 6 представлен еще один фрагмент кода, реализующий отображение окна разделов курса. В этом окне находятся вертикально прокручиваемые карточки для каждой темы, внутри которых включены и краткое описание темы, и кнопка, позволяющая перейти на страницу, посвященную заданной теме.

Модерирование контента осуществляется через административный раздел сервера. На него можно попасть, указав учетные данные администратора. Материалы курса, а также просмотр всех записей, внесенных в базу данных, доступны только администратору платформы.

Вся серверная часть платформы, в том числе и база данных, размещены на Heroku — облачной платформе, основанной на управляемой контейнерной системе, с интегрированными службами передачи данных и мощной экосистемой для развертывания и запуска современных приложений.

Разработанное приложение было протестировано в группе студентов первого курса при выполнении контрольной работы на факультативном занятии по программированию на Phyton. Контрольная работа включала в себя как теоретические вопросы об особенностях языка, так и написание необходимых алгоритмических конструкций по соответствующим заданиям. В качестве помощи для поиска необходимой информации в случае затруднения с ответами разрешалось пользоваться мобильными телефонами. Только одной половине студентов была представлена возможность искать необходимую информацию на любых сайтах Интернет, а другой половине разрешалось пользоваться только разработанным нами мобильным приложением. В ходе этого эксперимента удалось проверить надежность работы приложения — сбоев не было. Кроме того, благодаря концентрации только на одном приложении и его адаптированности для мобильного устройства вторая группа студентов справилась с работой быстрее первой. Наконец, опрос мнения студентов о качестве нового курса позволил сделать вывод, что курс удобен и обладает необходимой полнотой информации.

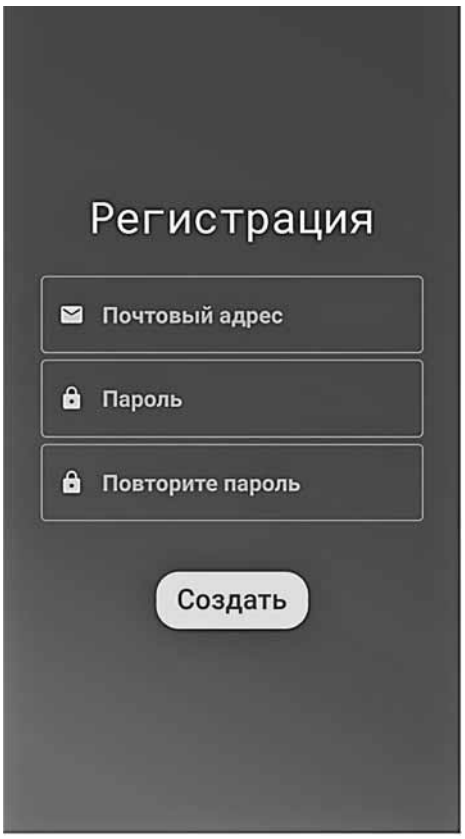


Рис. 2. Окно регистрации

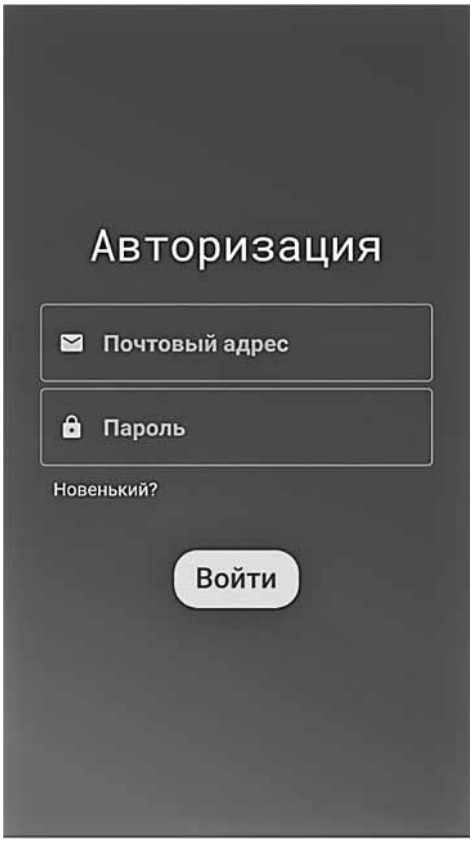


Рис. 3. Окно авторизации

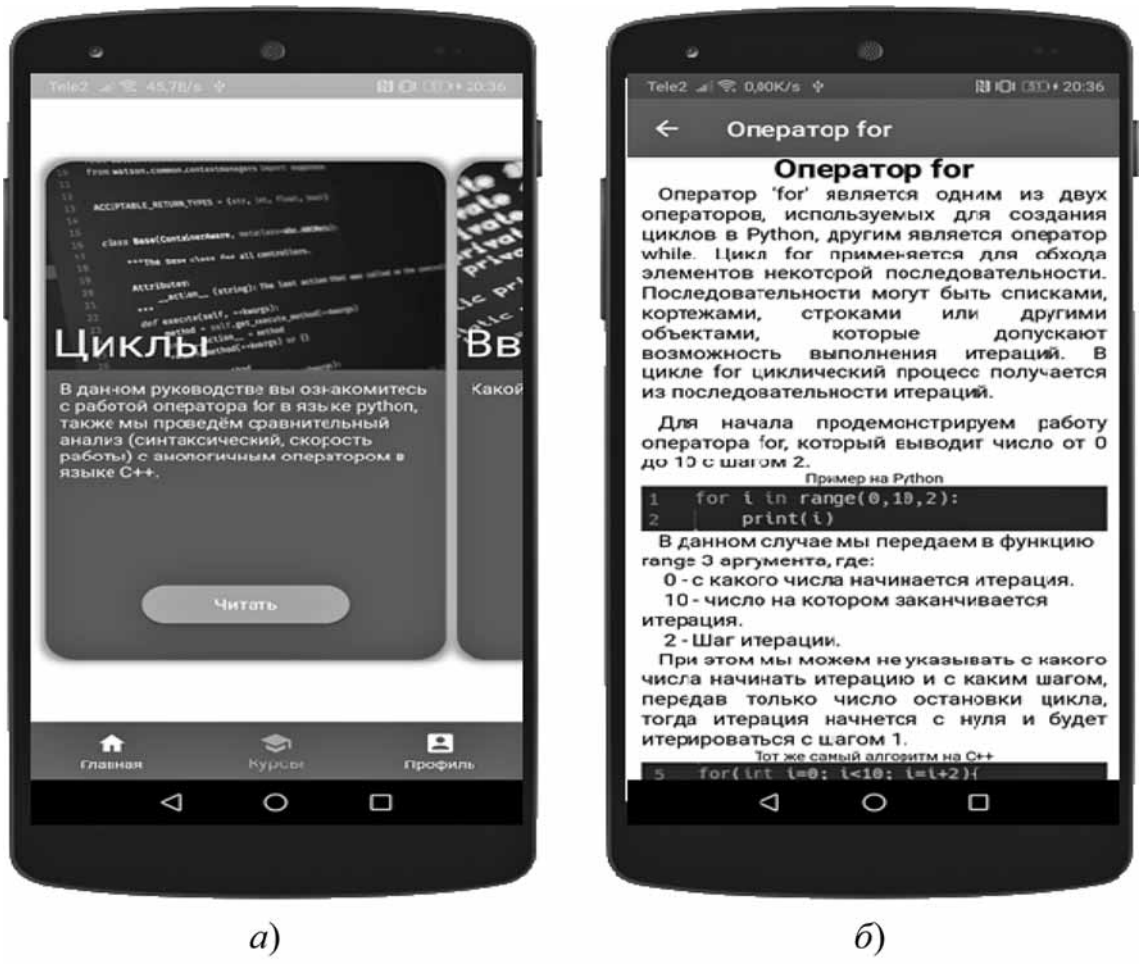


Рис. 4. Окно с кнопками навигации по приложению и одним из выбранных разделов курса (*а*) и содержательная часть этого раздела (*б*)

```
class _BottomBarScreenState extends State<BottomBarScreen>{
  int _selectedIndex = 0;
  final List<Widget> _tabs = [
    HomeScreen(),
    CoursesScreen(),
    ProfileScreen(),
  ];

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }
```

Рис. 5. Фрагмент кода нижней навигационной панели

```
class _CoursesScreenState extends State<CoursesScreen> {
  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Center(
        child: Container(
          height: 500,
          child: ListView(
            shrinkWrap: true,
            padding: const EdgeInsets.all(8.0),
            scrollDirection: Axis.horizontal,
            children: <Widget>[
              MyCard(Link1, title1, text1),
              MyCard(Link2, title2, text2),
              MyCard(Link3, title3, text3),
```

Рис. 6. Исходный код разделов курса

Заключение

В ходе исследования пройдены следующие ключевые этапы построения проекта.

1. Поиск и анализ существующих аналогичных мобильных приложений для выявления достоинств и недостатков каждого из них, чтобы учесть их при разработке своего приложения.

2. Разработка концепта дизайна для удобного использования мобильного приложения.

3. Проектирование и разработка программной части продукта на выбранной технологии.

4. Публикация продукта в магазин приложений для открытого доступа.

Таким образом, результатом проделанного исследования и разработки является завершенное, масштабируемое и стабильно работающее приложение для мобильной операционной системы Android, которое позволяет в режиме онлайн изучать язык программирования Python и способы написания программ с применением базовых алгоритмов на этом языке.

Дальнейшим этапом развития приложения будет его адаптация для платформы iOS. Мы планируем сделать курс интерактивным — включить возможность написания и отладки небольших фрагментов программ, а также разделы по самотестированию обучающихся.

Работа выполнена в рамках федеральной Программы повышения конкурентоспособности Казанского федерального университета и при финансовой поддержке РФФИ (грант 19-32-50108/19мол_нр).

Список литературы

1. **Aresta M., Pedro L., Santos C.** Mobile Learning and Higher Education: A Theoretical Overview // J. Mobile Multimedia. — 2015. — Vol. 11, No. 1—2. — P. 147—156.

2. **Drigas A. S., Angelidakis P.** Mobile Applications within Education // An Overview of Application Paradigms in Specific Categories. — 2017. — Vol. 11, No. 4. — P. 17—26.

3. **Mehdipour Y., Zerehkafi H.** Mobile learning for education: benefits and challenges // Int. J. Comput. Engin. Res. — 2013 — Vol. 03, No. 6. — P. 93—101.

4. **Куценко С. М., Дубовиков И. И.** Сравнительный анализ языков программирования // Ученые записки ИСГЗ. — 2019. — № 2 (17). — С. 170—177.

5. **Petrova N. K., Nefedyev Y. A., Abdulmjanov T., Zagidullin A. A., Andreev A.** The software complex for computer simulating the observation of stars from the lunar surfaces and calculating their selenographical coordinates // Proc. Intern. Conf. SGEM. Albena, Bulgaria. Jun. 28 — Jul. 9, 2017. — Vol. 17 (2,1). — P. 687—694.

6. **Валиуллин К. И., Давлетшин А. Д.** Разработка программного комплекса для компьютерного моделирования эксперимента по измерению координат звезд с поверхности Луны на C++ // Тезисы межд. молод. научн. конф. Тинчуринские чтения-2020. Энергетика и цифровая трансформация. КГЭУ, Казань, 28—29 апреля 2020. — С. 24—27.

7. **Salas-Zárate M., Alor-Hernández G., Valencia-García R.** et al. Analyzing best practices on Web development frameworks: The lift approach // Sci. Computer Programming. — 2015. — Vol. 102. — P. 1—19.

8. **Документация Django.** URL: https://docs.djangoproject.com/en/3.0/ (дата обращения 04.05.2020).

9. **Pedregosa F., Varoquaux G., Gramfort A.** et al. Scikit-learn: Machine learning in Python // J. Machine Learn. Res. — 2011. — Vol. 12. — P. 2825—2830

10. **Документация PostgreSQL.** URL: https://www.postgresql.org/docs/ (дата обращения 30.04.2020).

11. **Rieger Ch., Majchrzak T. A.** Towards the definitive evaluation framework for cross-platform app development approaches // J. Syst. Software. — 2019. — Vol. 153. — P. 175—199.

12. **Документация Flutter.** URL: https://flutter.dev/docs (дата обращения 4.05.2020).

13. **Adinugroho T. Y., Reina, Gautama J. B.** Review of Multi-Platform Mobile Application Development Using WebView: Learning Management System on Mobile Platform // Procedia Computer Science. — 2015. — Vol. 59. — P. 291 — 297.

Creating an Electronic Course on Programming in Python for the Android Platform

N. K. Petrova[1,2], nk_petrova@mail.ru, **A. P. Mukhachev**[1], houstondevs@gmail.com,
A. A. Zagidullin[2], arhtur.zagidullin@ya.ru, **S. M. Koutsenko**[1], s.koutsenko@mail.ru,
[1]Kazan Power Engineering University, Kazan, 420066, Russian Federation,
[2]Kazan Federal University, Kazan, 420008, Russian Federation

Corresponding author:

Petrova Natalia K., Associate Professor, Kazan State Power Engineering University, Kazan, 420066, Russian Federation, Senior Researcher, Institute of Physics of the Kazan Federal University, Kazan, 420008, Russian Federation
E-mail: nk_petrova@mail.ru

Received on September 29, 2020
Accepted on April 19, 2021

The description and principles of developing a mobile application for the Android platform that provides free access to electronic courses on teaching the basic structures of the Python language and the construction of template programming algorithms based on them are presented. The content of the course is based on the principle of comparative analysis with the C++ language, one of the goals of which is to differentiate the tasks for which it is more efficient to use either the Python scripting language or the C++ compiler. The developed application is logically integral, allows the possibility of supplementing with new data — examples, types of algorithms — and, no less important, is free.

Keywords: *training of algorithmization, programming basics, Python, platform, framework, mobile application, server*

Acknowledgements:

This work was performed according to the Russian Government Program of Competitive Growth of Kazan Federal University and was partially supported by the Russian Foundation for Basic Research grant no. 19-32-50108\19мол_нр.

For citation:

Petrova N. K., Mukhachev A. P., Zagidullin A. A., Koutsenko S. M. Creating an Electronic Course on Programming in Python for the Android Platform, *Programmnaya Ingeneria,* 2021, vol. 12, no. 4, pp. 216—222.

DOI: 10.17587/prin.12.216-222

References

1. **Aresta M., Pedro L., Santos C.** Mobile Learning and Higher Education: A Theoretical Overview, *J. Mobile Multimedia,* 2015, vol. 11, no. 1—2, pp. 147—156.
2. **Drigas A. S., Angelidakis P.** Mobile Applications within Education, *An Overview of Application Paradigms in Specific Categories,* 2017, vol. 11, no. 4, pp. 17—26.
3. **Mehdipour Y., Zerehkafi H.** Mobile learning for education: benefits and challenges, *Int. J. Comput. Engin. Res.,* 2013, vol. 03, no.6, pp. 93—101.
4. **Koutsenko S. M., Doubovikov I. I.** Comparative analysis of programming languages, *Uchenye zapiski ISGZ,* 2019, no. 2 (17), pp. 170—177 (in Russian).
5. **Petrova N. K., Nefedyev Y. A., Abdulmjanov T., Zagidullin A. A., Andreev A.** The software complex for computer simulating the observation of stars from the lunar surfaces and calculating their selenographical coordinates, *Proc. Intern. Conf. SGEM.* Albena, Bulgaria. Jun. 28 — Jul. 9, 2017, vol. 17 (2,1), pp. 687—694.
6. **Valiullin K. I., Davletshin A. D.** Razrabotka programmnogo kompleksa dlya komp'yuternogo modelirovaniya eksperimenta po izmereniyu koordinat zvyozd s poverhnosti Luny na S++, *Tezisy mezhd. molod. nauchn. konf. Tinchurinskie chteniya-2020. Energetika i cifrovaya transformaciya*, KGEU, Kazan', 28—29 April 2020, pp. 24—27 (in Russian).
7. **Salas-Zárate M., Alor-Hernández G., Valencia-García R.** et al. Analyzing best practices on Web development frameworks: The lift approach, *Sci. Computer Programming,* 2015, vol. 102, pp. 1-19.
8. **Django Documentation,** available at: https://docs.djangoproject.com/en/3.0/ (date of access 4.05.2020).
9. **Pedregosa F., Varoquaux G., Gramfort A.** et al. Scikit-learn: Machine learning in Python, *J. Machine Learn. Res.*, 2011, vol. 12 (2011), pp. 2825—2830.
10. **PostgreSQL Documentation,** available at: https://www.postgresql.org/docs/ (date of access 30.04.2020).
11. **Rieger Ch., Majchrzak T. A.** Towards the definitive evaluation framework for cross-platform app development approaches, *J. Syst. Software*, 2019, vol. 153, pp. 175—199.
12. **Flutter Documentation,** available at: https://flutter.dev/docs (date of access 04.05.2020).
13. **Adinugroho T. Y., Reina, Gautama J. B.** Review of Multi-Platform Mobile Application Development Using WebView: Learning Management System on Mobile Platform, *Procedia Computer Science*, 2015, vol. 59, pp. 291—297.

X Всероссийская научно-техническая конференция

Проблемы разработки перспективных микро- и наноэлектронных систем –

МЭС-2021

Конференция МЭС посвящена актуальным вопросам автоматизации проектирования МЭС, систем на кристалле, IP-блоков и новой элементной базы микро- и наноэлектроники

Конференция МЭС является крупнейшей конференцией в области САПР микроэлектроники на территории России и стран СНГ

В этом году конференция проходит в виде Интернет-форума с проведением заседаний научных секций разной тематической направленности в online-режиме. Конференция завершится проведением очного Пленарного заседания, на котором будут подведены ее итоги, а также секции «Презентации новых микроэлектронных проектов, САПР и готовых продуктов» на которой партнеры и спонсоры конференции представят доклады о своих разработках.

Прием докладов осуществляется **с 01 марта по 01 августа 2021 г.** Принятые доклады будут опубликованы на web-сайте конференции, а также в четырех выпусках трудов конференции, которые будут издаваться по мере поступления, рецензирования и редакционной подготовки текстов статей. Как и в 2020 году, будут проведены конкурсы на лучшие доклады с призами для победителей.

Участие в конференции МЭС-2021 бесплатное.

Более подробную информацию можно найти на web-сайте конференции МЭС-2021:

mes-conference.ru

Институт Математики им. С. Л. Соболева СО РАН
Новосибирский национальный исследовательский государственный университет
Российская Ассоциация Искусственного Интеллекта
Российская Инженерная Академия
Institute of Electrical and Electronics Engineers (IEEE)

VIII Международная конференция

ЗНАНИЯ – ОНТОЛОГИИ – ТЕОРИИ

8—12 ноября 2021 г., Новосибирск

Целью конференции является ознакомление с новейшими научными достижениями, обмен знаниями и передовым опытом в области математических методов представления и анализа данных, извлечения знаний и построения теорий предметных областей, анализа формальных понятий и извлечения информации из текстов естественного языка. Сборник трудов конференции будет проиндексирован в РИНЦ, избранные статьи будут проиндексированы в Scopus.

Тематика конференции отражает основные стадии процесса познания:

- **Обнаружение закономерностей и извлечение знаний,** скрытых в структурированных и неструктурированных данных. Машинное обучение. Распознавание образов, анализ данных. Прогнозирование. Индуктивный вывод
- **Систематизация знаний.** Инженерия знаний. Управление знаниями. Извлечение знаний из текстов на естественном языке. Разработка онтологий предметных областей, технологии создания и применения онтологий
- **Построение теорий предметных областей.** Разработка семантических и онтологических моделей предметных областей. Анализ формальных понятий. Логическая семантика естественного языка. Нечёткие логики

Работа конференции планируется в виде пленарных, секционных и стендовых докладов и круглых столов по тематике конференции. Рабочие языки конференции — русский и английский

Контактные данные для переписки: **zont@math.nsc.ru**

ООО "Издательство "Новые технологии". 107076, Москва, Стромынский пер., 4
Технический редактор *Е. М. Патрушева.* Корректор *Е. В. Комиссарова*

Сдано в набор 11.05.2021 г. Подписано в печать 30.06.2021 г. Формат 60×88 1/8. Заказ PI421
Цена свободная.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: **www.aov.ru**

Рисунок к статье **С. А. Букашкина, М. А. Черепнёва**

«КВАНТОВЫЙ КОМПЬЮТЕР И ПОСТКВАНТОВАЯ КРИПТОГРАФИЯ»

• Для шифрования с открытым ключом и инкапсуляции ключа – 16 схем: **BIKE**, **Classic McEliece**, **CRYSTALS-KYBER**, **FrodoKEM**, **HQC**, **LAC**, **LEDAcrypt** (производная схема от **LEDAkem/LEDApkc**), **NewHope**, **NTRU** (производная схема от **NTRUEncrypt/NTRU-HRSS-KEM**), **NTRU Prime**, **NTS-KEM**, **ROLLO** (производная схема от **LAKE/LOCKER/Ouroboros-R**), **Round5** (производная схема от **Hila5/Round2**), **RQC**, **SABER**, **SIKE**, **Three Bears**.

• Для цифровой подписи – 9 схем: **CRYSTALS-DILITHIUM**, **FALCON**, **GeMSS**, **LUOV**, **MQDSS**, **Picnic**, **qTESLA**, **Rainbow**, **SPHINCS+**.

Коды — решетки — изогении — многочлены — хеш — прочее

Название	Тип	Прямая операция	Обратная операция
Lepton	Ш	244	282
Three Bears	Ш, ВОК	145	213
qTESLA	П	26063	1310
Classic McEliece	Ш	3000	450
DILITHIUM	П	711	288
FrodoKEM	ВОК	3577	3580
RQC	ВОК, Ш	6460	18000
NTRU	Ш	59	97
NewHope	ВОК	210	220
SIKE	ВОК	14955	17957
Rainbow	П	8688	6174
LUOV	П	216000	124000
SPHINCS	П	164592	3975
pqRSA	Ш,П	22×10^9	$1{,}8\times10^9$
RSA	Ш, П, ВОК	75000	5000
EC-DLP	Ш, П, ВОК	400	600

П – подпись, Ш – шифрование, ВОК – выработка общего ключа

Результаты конкурса

Рисунок к статье **Д. И. Читалова**

«О РАЗРАБОТКЕ МОДУЛЯ ДЛЯ МОДИФИКАЦИИ РАСЧЕТНЫХ СЕТОК ПОСРЕДСТВОМ УТИЛИТЫ dsmcInitialise ПРОГРАММНОЙ СРЕДЫ OpenFOAM»

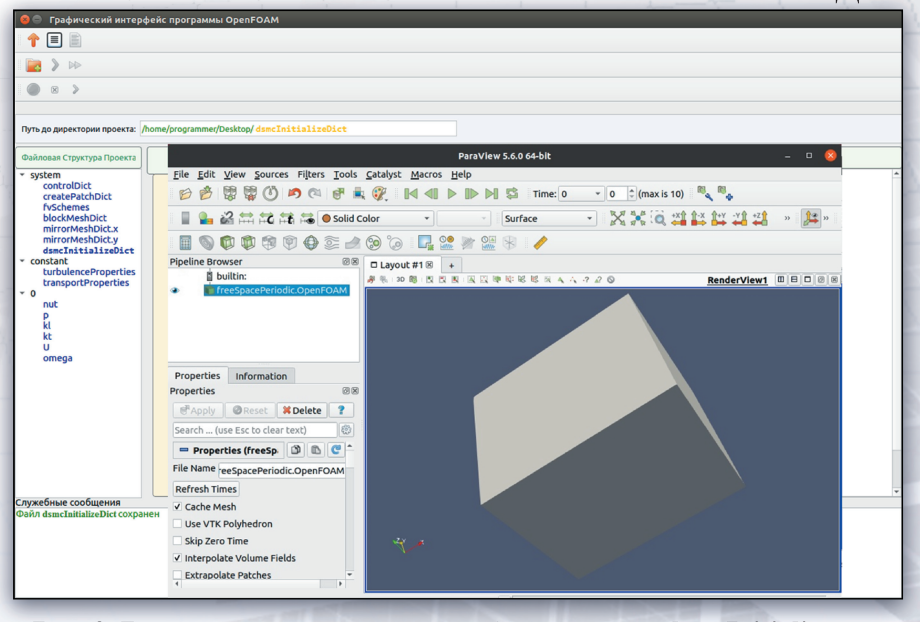


Рис. 3. Визуализация результатов работы утилиты dsmcInitialise на примере одной из учебных задач OpenFOAM

Издательство «НОВЫЕ ТЕХНОЛОГИИ»
выпускает *научно-технические журналы*

Теоретический и прикладной научно-технический журнал

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

В журнале освещаются состояние и тенденции развития основных направлений индустрии программного обеспечения, связанных с проектированием, конструированием, архитектурой, обеспечением качества и сопровождением жизненного цикла программного обеспечения, а также рассматриваются достижения в области создания и эксплуатации прикладных программно-информационных систем во всех областях челове ческой деятельности.

Подписной индекс по Объединенному каталогу
«Пресса России» – 22765

Ежемесячный теоретический и прикладной научно-технический журнал

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В журнале освещаются современное состояние, тенденции и перспективы развития основных направлений в области разработки, производства и применения информационных технологий.

Подписной индекс по Объединенному каталогу
«Пресса России» – 72656

Междисциплинарный теоретический и прикладной научно-технический журнал

НАНО- и МИКРОСИСТЕМНАЯ ТЕХНИКА

В журнале освещаются современное состояние, тенденции и перспективы развития нано- и микросистемной техники, рассматриваются вопрос разботки и внедрения нанои микросистем в различные области науки, технологии и производства.

Подписной индекс по Объединенному каталогу
«Пресса России» – 79493

Ежемесячный теоретический и прикладной научно-технический журнал

МЕХАТРОНИКА, АВТОМАТИЗАЦИЯ, УПРАВЛЕНИЕ

В журнале освещаются достижения в области мехатроники, интегрирующей механику, электронику, автоматику и информатику в целях совершенствования технологий производства и создания техники новых поколений. Рассматриваются актуальные проблемы теории и практики автомати ческого и автоматизированного управления техническими объектами и технологическими процессами в промышленности, энергетике и на транспорте.

Подписной индекс по Объединенному каталогу
«Пресса России» – 79492

Научно-практический и учебно-методический журнал

БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

В журнале освещаются достижения и перспективы в области исследований, обеспечения и совершенствования защиты человека от всех видов опасностей производственной и природной среды, их контроля, мониторинга, предотвращения, ликвидации последствий аварий и катастроф, образования в сфере безопасности жизнедеятельности.

Подписной индекс по Объединенному каталогу
«Пресса России» – 79963

Адрес редакции журналов для авторов и подписчиков:
107076, Москва, Стромынский пер., 4. Издательство "НОВЫЕ ТЕХНОЛОГИИ".
Тел.: (499) 269-55-10, 269-53-97. Факс: (499) 269-55-10. E-mail: antonov@novtex.ru

ISSN 2220-3397. Программная инженерия. 2021. Том 12. № 4. 169 – 224. Подписной индекс по Объединенному каталогу "Пресса России" – 22765