

В.В. Кручинин, Ю.В. Морозова, Ю.А. Зорин

Томский государственный университет систем управления и радиоэлектроники,
г. Томск, Россия

ПОСТРОЕНИЕ И ИСПОЛЬЗОВАНИЕ ГЕНЕРАТОРОВ ТЕСТОВЫХ ЗАДАНИЙ В СИСТЕМАХ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

Рассматриваются вопросы организации получения тестовых заданий на основе программных генераторов. Описывается опыт разработки и применения таких генераторов в ТУСУРе. Показана обобщенная структура генератора, рассмотрены инструментальные системы построения генераторов и примеры их использования и применения.

Ключевые слова: дистанционное обучение, алгоритмы комбинаторной генерации, генераторы тестовых заданий.

Введение

Опыт организации учебного процесса в Томском государственном университете систем управления и радиоэлектроники (ТУСУР) по дистанционной технологии обучения показал, что ее важнейшей составляющей является система контроля знаний [1]. В конечном счете от качества системы контроля знаний зависит качество выпускаемых специалистов. На факультете дистанционного обучения (ФДО) ТУСУРа создана довольно развитая система контроля знаний, которая включает организацию промежуточного контроля различной формы: выполнение компьютерных контрольных работ, написание письменных работ, рефератов, отчетов, представляемых в электронной форме, а также проведение компьютерного экзамена в форме тестирования [2].

Анализ проведения экзаменационных сессий показал, что:

- 1) хакеры организуют атаки на систему проведения компьютерного тестирования;
 - 2) для многих компьютерных экзаменов подготовлены шпаргалки;
 - 3) шпаргалки и взломанные компьютерные экзамены размещаются на специальных сайтах.
- Вследствие этого была выработана стратегия улучшения системы контроля знаний [3]:

- 1) создание системы защиты от несанкционированного доступа;
- 2) организация защиты от шпаргалок;

- 3) организация работ по закрытию сайтов;
- 4) совершенствование системы проведения экзаменов и контрольных работ.

Одним из возможных направлений решения указанных проблем является внедрение в практику дистанционного контроля генераторов тестовых заданий. Генераторы, с одной стороны, решают проблемы защиты от несанкционированного доступа, так как не имеют заранее заготовленных ответов, с другой стороны, практически каждый студент получает индивидуальное задание. Это решает проблему шпаргалки, потому что программа генерирует правильный ответ в процессе проведения опроса. Следовательно, вместо запоминания правильного ответа необходимо знать алгоритм решения.

Было выявлено два основных направления исследований:

1. Моделирование деятельности преподавателя при приеме экзамена. Применение идей искусственного интеллекта для решения задачи синтеза вопросов. Построение интеллектуальной системы приема экзаменов.

2. Инженерный подход при построении генераторов, основанный на применении эвристических и комбинаторных алгоритмов генерации многовариантных тестовых заданий и вопросов.

Первое направление является наиболее перспективным, так как при его осуществлении будет решена задача качественного приема компьютер-

ного экзамена. Однако первые попытки решения натолкнулись на задачи, которые не имеют удовлетворительного решения на современном уровне развития искусственного интеллекта. Одна из таких задач – понимание текста на естественном языке. Отсюда невозможность решения задачи синтеза корректного вопроса.

Результаты исследований в этом направлении могут привести к появлению интеллектуальных систем, помощников преподавателя при составлении тестовых вопросов.

Второе направление связано с построением алгоритмов, которые генерируют многовариантные тестовые задания и вопросы. Например, в математической задаче можно менять различные входные параметры и тем самым получать множество однотипных задач. Кроме того, имеются определенные результаты в этой области, показывающие возможность использования таких алгоритмов.

В процессе развития второго направления удалось:

- 1) разработать математические основы построения алгоритмов генерации тестовых заданий;
- 2) построить модели и алгоритмы генерации тестовых заданий;
- 3) разработать и внедрить оригинальную технологию создания генераторов;
- 4) разработать серию генераторов по различным дисциплинам и внедрить их в практику дистанционного обучения.

1. Математические основы построения алгоритмов генерации тестовых заданий

При разработке генератора тестовых заданий необходимо предварительно оценить число возможных тестовых заданий, способы идентификации тестовых заданий, методы управления генерацией заданий, методы получения правильных ответов и оценки правильности ответов обучаемых. Кроме того, в процессе эксплуатации генератора заданий необходимо обеспечить процедуру апелляции, при которой студенту необходимо выдать правильное решение для данного сгенерированного задания.

Оценка мощности генератора. Важной характеристикой генератора является его мощность, которая определяется как мощность множества генерируемых объектов. В общем случае для исследования мощности множества объектов используются методы теории перечислительной

комбинаторики, однако часто грубая оценка мощности позволяет произвести обоснованный выбор для реализации рассматриваемого проекта генератора. Для принятия решения о разработке генератора необходимо оценить численность студентов, изучающих данную дисциплину, для того чтобы каждый студент получил индивидуальное задание. Отсюда следует, что генератор должен иметь мощность, большую, чем число студентов.

Идентификация задания. При получении нового задания генератор должен сформировать некоторый объект (код, число), по которому однозначно можно было бы получить это задание вновь. Назовем этот объект идентификатором задания. Для генератора желательно иметь взаимно однозначное соответствие между идентификатором задания и сгенерированным заданием. Одним из простых решений является хранение выданных заданий и их идентификаторов в некоторой таблице (базе данных). Другим простым решением в случае использования датчика случайных чисел в генераторе является использование начального значения датчика случайных чисел в качестве идентификатора задания. Однако наиболее перспективным является использование алгоритмов комбинаторной генерации, которые обеспечивают генерацию тестового задания по заданному номеру.

Алгоритмы комбинаторной генерации. Подавляющее большинство тестовых заданий можно сформулировать как комбинаторные задачи. Например, задачу на упорядочение можно рассматривать как некоторую перестановку блоков (элементов), задачу выбора некоторого подмножества из множества или множеств можно рассматривать как задачу на сочетания, разбиения, разложения или композиции и т.д.

Кроме того, много заданий можно сформулировать в виде некоторых графов или деревьев [4]. Алгоритмы комбинаторной генерации предназначены для получения элементов комбинаторных множеств и включают три класса алгоритмов:

- 1) алгоритмы генерации элементов комбинаторных множеств по заданному порядковому номеру;
- 2) алгоритмы нумерации элементов комбинаторных множеств;
- 3) алгоритмы последовательной генерации элементов комбинаторных множеств.

Существует огромное число конкретных алгоритмов генерации комбинаторных множеств. Также существуют методы построения алгоритмов комбинаторной генерации, обладающие некоторой универсальностью: генерирующие деревья, методы ограниченного перебора и т.д. Среди этих методов можно выделить методы, основанные на рекурсивной композиции деревьев И / ИЛИ, которые позволяют задать рекуррентную формулу или алгоритм для вычисления мощности комбинаторного множества. Если мощность комбинаторного множества задана в виде функции $f(n)$ алгебры $\{N, +, \times, R\}$, где N – множество натуральных чисел, R – оператор примитивной рекурсии, то для такого множества можно однозначно построить алгоритм последовательной генерации и алгоритмы генерации элемента по номеру и нумерации элемента.

Важно отметить, что использование методов и алгоритмов комбинаторной генерации позволяет решить одну из самых сложных проблем, связанных с генерацией тестовых заданий, – обеспечение правильной работы генератора и упрощение поиска проблемных ситуаций, возникающих при генерации тестовых заданий.

Управление генерацией. В процессе эксплуатации генератора тестовых заданий необходимо управлять генерацией. Например, в тех случаях, когда используется датчик случайных чисел, необходимо избегать генерации одинаковых заданий для студентов, которые находятся в одном классе, или избежать генерации неудачных заданий и т.п. В этом случае необходимо предусмотреть механизм исключения тестового задания из множества возможных. Для этого разрабатывают базу данных с исключенными идентификаторами заданий. Кроме того, как правило, генераторы объединяются в некоторые пакеты, например генераторы задач по одной теме и т.д. Кроме того, генераторы могут выдавать задания разной сложности.

2. Структура генератора тестовых заданий

Рассмотрим обобщенную структуру генератора заданий (рис. 1). Входом для генератора является некоторое задание на генерацию, представленную в виде описания конфигурации, в которой могут быть представлены варианты генерации и некоторые ограничения, число сгенерированных заданий, их варианты, установка некоторых параметров и т.д.

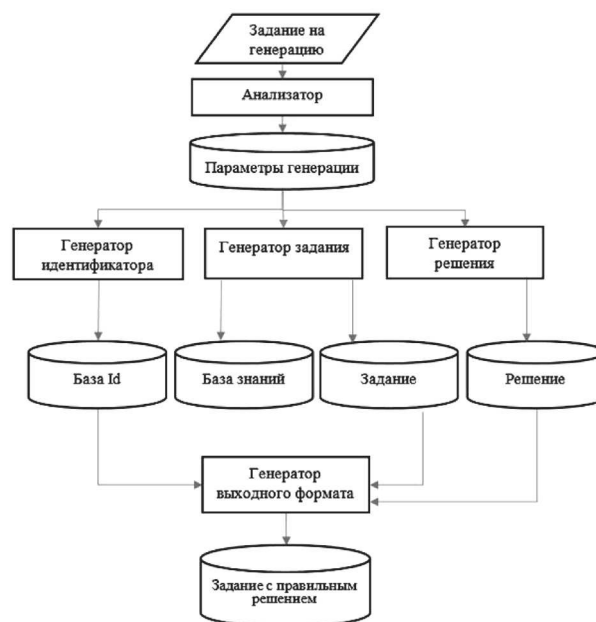


Рис. 1. Структура генератора заданий

Это задание на генерацию поступает на вход анализатора, который сканирует это задание и выявляет ошибки, и если ошибок нет, то формирует внутреннее представление параметров генерации. Если необходимо получить новое тестовое задание, то генерируется новый идентификатор задания, который затем записывается в базу данных. Если идентификатор задан, находится запись в базе данных и генерируется уже известное задание.

После того как стали известны идентификатор и заданные параметры, запускается генератор задания, который использует базу знаний, в которой записаны шаблоны задач, например в виде некоторых фреймов.

Далее тестовое задание поступает на вход генератора решения, генератор решения на основе полученных параметров генерации и параметров тестового задания находит правильный ответ, возможно, также формирует последовательность действий (решение) [5].

3. Основные этапы построения генераторов тестовых заданий

Можно выделить следующие основные этапы при разработке генератора тестовых заданий:

1. Получение технического задания.
2. Этап технического проектирования.

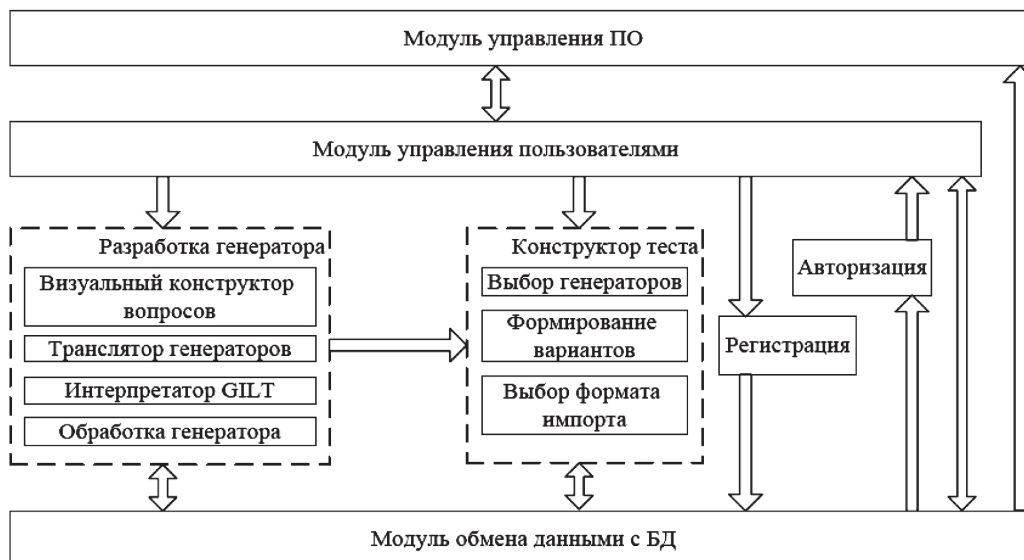


Рис. 2. Структура системы «Платан»

3. Этап программирования и отладки.

4. Этап тестирования и опытной эксплуатации.

4. Инструментальные системы для построения генераторов тестовых заданий

Инструментальные системы построения генераторов тестовых заданий призваны обеспечить снижение затрат на создание и эксплуатацию такого программного обеспечения [6]. Опишем инструментальную систему «Платан», которая реализована на применении алгоритмов деревьев И / ИЛИ. На рис. 2 представлена структура инструментальной системы «Платан».

В этой системе реализован оригинальный язык представления генератора GILT и технологии визуального проектирования генератора заданий.

Технология визуального проектирования генератора заданий заключается в следующем:

1. Формируется основная структура генератора в виде узла (рис. 3).

2. Последовательно раскрывается каждый из узлов: *problem*, *solution*, *answer*.

3. Узел *problem* описывает формулировку задания и его параметры.

4. Узел *solution* описывает решение задачи.

5. Узел *answer* описывает форму и эталонный ответ.

6. В узлах записываются утверждения языка GILT, который содержит переменные и констан-

ты, выражения для записи условий, операции назначения и арифметические операции, операторы для записи выражений в языке Latex и др.

7. В узлах также могут записаны таблицы и рисунки.

Данная технология, внедренная на ФДО ТУ-СУРА, показала:

1) сокращение сроков разработки генераторов;

2) обеспечение надежной технологии контроля знаний в системе дистанционного обучения.

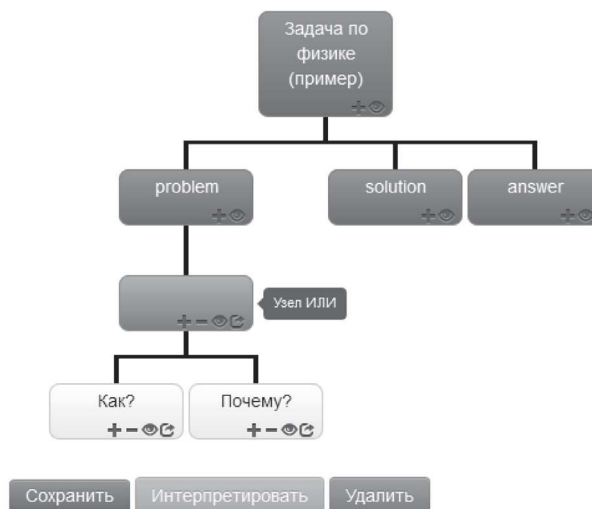


Рис. 3. Основная структура дерева генератора

5. Примеры генераторов тестовых заданий

Рассмотрим пример организации самостоятельной работы на основе генератора тестовых заданий в системе дистанционного обучения ТУСУРа.

Исходные данные для создания генератора по разделу «Дифференциальное исчисление»

Формулировка обобщенной задачи. Найдите $\frac{\partial z}{\partial x}$ и $\frac{\partial^2 z}{\partial y \partial x}$ функции $Z(x, y)$, заданной неявно уравнением:

$$\begin{aligned} \Phi(x, y, z) = & ax^2 + by^2 + cz^3 + mxy + nxz + pyz - \\ & - (1 + 3cv^2 + ns + pu)z + v + \\ & + 2cv^3 - as^2 - bu^2 - msu = 0. \end{aligned}$$

Последовательность ввода значений: в ответ введите сначала значение $\frac{\partial z}{\partial x}$, а затем через точку с запятой значение $\frac{\partial^2 z}{\partial y \partial x}$ в точке $M(s; u; v)$.

Алгоритм формирования параметров:

$$2as + tu + nv; m + nT + 6cvRT + pR, \text{ где } R = 2as + tu + nv, T = 2bu + ms + pv.$$

Генерируемые параметры: $a, b, c, m, n, p, s, u, v$.

Ограничения на параметры:

Параметрам s, u, v придадим значения $0 \pm 1; \pm 2; \pm 3$ независимо друг от друга.

Теоретические указания к задаче:

Указание 1. Если уравнение $\Phi(x, y, z) = 0$ определяет неявно в области D функцию $Z = Z(x, y)$, т.е. в области D выполняется тождество $\Phi(x, y, z(x, y)) \equiv 0$ относительно $(x, y) \in D$ и функция $\Phi(x, y, z)$ имеет частные производные $\Phi'_x, \Phi'_y, \Phi'_z$, причем $\Phi'_z \neq 0$, то справедливы формулы:

$$\frac{\partial z}{\partial x} = -\frac{\Phi'_x}{\Phi'_z}, \quad \frac{\partial z}{\partial y} = -\frac{\Phi'_y}{\Phi'_z}. \quad (1)$$

Частные производные второго порядка $\frac{\partial^2 z}{\partial x^2}, \frac{\partial^2 z}{\partial x \partial y}, \frac{\partial^2 z}{\partial y^2}$ можно получить, дифференцируя соотношения (1) либо по x , либо по y , считая при этом переменную z функцией от (x, y) , частные производные первого порядка от которой уже найдены. Находим:

$$\frac{\partial^2 z}{\partial x^2} = \left(-\frac{\Phi'_x}{\Phi'_z}\right)'_x, \quad \frac{\partial^2 z}{\partial y \partial x} = \left(-\frac{\Phi'_x}{\Phi'_z}\right)'_y, \quad \frac{\partial^2 z}{\partial y^2} = \left(-\frac{\Phi'_y}{\Phi'_z}\right)'_y. \quad (2)$$

Частные производные третьего порядка можно найти, дифференцируя соотношения (2) по соответствующей переменной.

Алгоритм формирования общего решения:

1. Частную производную $\frac{\partial z}{\partial x}$ находим по первой формуле в (1):

$$\begin{aligned} \frac{\partial z}{\partial x} &= -\frac{2ax + my + nz}{3cz^2 + nx + py - (1 + 3cv^2 + ns + pu)} = \\ &= \frac{2ax + my + nz}{1 + 3cv^2 + ns + pu - 3cz^2 - nx - py}. \end{aligned}$$

2. Вычисляем значение $\frac{\partial z}{\partial x}$ при $x = s, y = u, z = v$:

$$\begin{aligned} \frac{\partial z}{\partial x}(s, u, v) &= \frac{2as + tu + nv}{1 + 3cv^2 + ns + pu - 3cv^2 - ns - pu} = \\ &= 2as + tu + nv = R. \end{aligned}$$

3. Для отыскания смешанной частной производной $\frac{\partial^2 z}{\partial y \partial x}$ нам понадобится частная производная $\frac{\partial z}{\partial x}$, которую находим по второй формуле в (1):

$$\frac{\partial z}{\partial y} = -\frac{2by + mx + pz}{3cz^2 + nx + py - (1 + 3cv^2 + ns + pu)}.$$

Вычислим значение $\frac{\partial z}{\partial x}$ в точке $M(s; u; v)$:

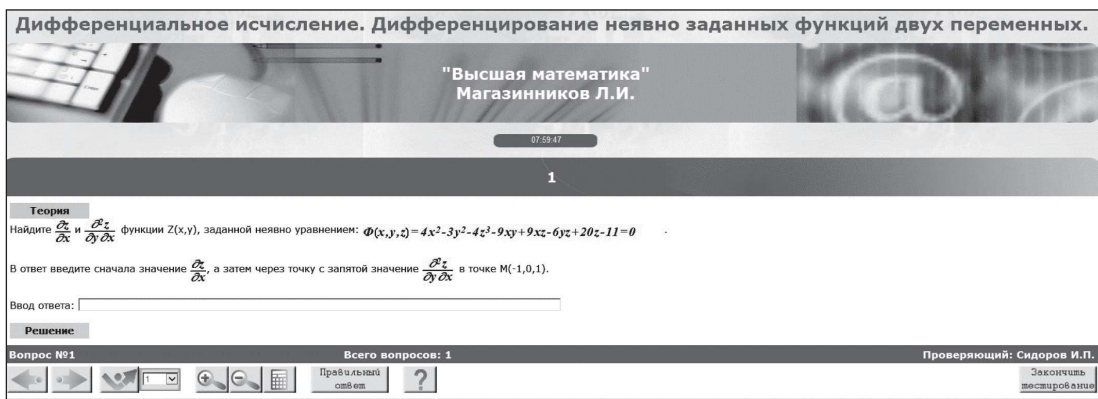


Рис. 4. Задание, сгенерированное из генератора по разделу «Дифференциальное исчисление»

$$\frac{\partial z}{\partial y}(s, u, v) = -\frac{2bu + ms + pv}{3cv^2 + ns + pu - 1 - 3cv^2 - ns - pu} =$$

$$= 2bu + ms + pv = T.$$

4. Для отыскания частной производной $\frac{\partial^2 z}{\partial y \partial x}$ применяем второе соотношение в (2). Выражение для $\frac{\partial z}{\partial x}$ дифференцируем по y , учитывая, что z есть функция от x и y . Можно было дифференцировать выражение для $\frac{\partial z}{\partial y}$ по x .

$$\frac{\partial^2 z}{\partial y \partial x} =$$

$$= \frac{(m + nz'_y)(A - 3cz^2 - nx - py) + (6cz'_y + p)(2ax + my + nz)}{(A - 3cz^2 - nx - py)^2}$$

(Обозначено $A = 1 + 3cv^2 + ns + pu$, заметим, что при $x = s$, $y = u$, $z = v$ величина $A - 3cz^2 - nx - py$ равна единице).

5. Вычислим значение $\frac{\partial^2 z}{\partial y \partial x}$, если

$$x = s, y = u, z = v, z'_y = 2bu + ms + pv.$$

$$\frac{\partial^2 z}{\partial y \partial x}(s, u, v) = m + n(2bu + ms + pv) +$$

$$+ [6cv(2bu + ms + pv) + p](2as + mu + nv) =$$

$$= m + nT + 6cvRT + pR.$$

На рис. 4 показано задание, сгенерированное из заданного шаблона.

Выводы

1. Опыт использования генераторов тестовых заданий показывает их эффективность в решении задачи индивидуального обучения и неизбежность их внедрения в технологии электронного обучения.

2. Развитие генераторов тестовых заданий показывает, что данный класс программных систем относится к системам искусственного интеллекта, которые моделируют деятельность преподавателя.

3. Современные подходы к разработке генераторов говорят о необходимости использования продвинутых инструментальных систем.

ЛИТЕРАТУРА

1. Кручинин В.В. Модели и алгоритмы генерации задач в компьютерном тестировании / В.В. Кручинин, Ю.В. Морозова // Изв. ТПУ. – 2004. – Т. 307, № 5. – С. 127–131.
2. Кручинин В.В. Модели и алгоритмы компьютерных самостоятельных работ на основе генерации тестовых / В.В. Кручинин, Л.И. Магазинников, Ю.В. Морозова // Изв. ТПУ. – 2006. – Т. 309, № 8. – С. 258–262.
3. Кручинин В.В. Генераторы в компьютерных учебных программах / В.В. Кручинин. – Томск : Изд-во Том. ун-та, 2003. – 200 с.
4. Титков А.В. Генерация тестовых заданий на основе грамматик в системе построения генераторов на деревьях И / ИЛИ / А.В. Титков // Материалы Международной научно-методической конференции «Современное образование». – Томск, 2010. – С. 163–164.
5. Кручинин В.В. Организация самостоятельной работы студентов с использованием технологии генерации тестовых заданий / В.В. Кручинин, Л.И. Магазинников, Ю.В. Морозова // Системы автоматизации в образовании, науке и производстве. – Новокузнецк : СибГИУ, 2005. – С. 351–353.
6. Зорин Ю.А. Инструментальные системы построения и получения многовариантных тестовых заданий / Ю.А. Зорин, И.А. Посов // Компьютерные инструменты в образовании. – 2014. – № 1. – С. 14–25.

Kruchinin V.V., Morozova Yu.V., Zorin Ya.A.
Tomsk State University of Control Systems and
Radioelectronics, Tomsk, Russia
DEVELOPMENT AND APPLICATION
OF TEST ASSIGNMENT GENERATORS
IN DISTANCE LEARNING SYSTEMS

Keywords: distance learning, combinatorial generation algorithms, test assignment generators.

Experience accumulated by Tomsk State University of Control Systems and Radioelectronics (TUSUR University) in employment of distance learning technologies for delivery of education shows that a system of knowledge assessment is a crucial component of such technologies. Ultimately, the quality of the assessment system determines the quality of graduates. The Faculty of Distance Learning of TUSUR University has created a rather well-established system of knowledge assessment that includes: various forms of midterm assessment computer-aided review works, writing assignments, review papers, reports submitted in digital form; computer testing.

Analysis of experience with examination has shown that:

- 1) hackers target the computer testing system;
- 2) cheat sheet are available for many computer tests;

3) cheat sheets and hacked computer tests are available at special websites.

As a result, the following strategy has been developed in order to improve the assessment system:

- 1) to develop an access security system;
- 2) to organize anti-cheat sheet protection;
- 3) to organize website shutdown;

4) to develop the examination and review system.

One potential way to address these problems is to integrate test assignment generators in the system of distance knowledge assessment. On the one hand, such generators ensure access security as they contain no ready-to-use answers, and on the other hand, virtually each student is given an individual assignment. That solves the problem of cheat sheets because the program generates the right answer during the testing, which means that instead of knowing the right answer, the student must know the right solution algorithm.

While developing such generators, the team has:

- 1) developed the mathematical framework for algorithms of test assignment generation;
- 2) developed models and algorithms for test assignment generation.

3) developed and implemented an original generator technology;

4) developed a series of generators for various academic subjects and implemented them in distance learning practices.

REFERENCES

1. *Kruchinin V.V.* Modeli i algoritmy generacii zadach v komp'yuternom testirovanii / V.V. Kruchinin, Ju.V. Morozova // *Izv. TPU.* – 2004. – Т. 307, № 5. – S. 127–131.
2. *Kruchinin V.V.* Modeli i algoritmy komp'yuternyh samostojatel'nyh rabot na osnove generacii testovyh / V.V. Kruchinin, L.I. Magazinnikov, Ju.V. Morozova // *Izv. TPU.* – 2006. – Т. 309, № 8. – S. 258–262.
3. *Kruchinin V.V.* Generatory v komp'yuternyh uchebnyh programmah / V.V. Kruchinin. – Tomsk : Izd-vo Tom. un-ta, 2003. – 200 s.
4. *Titkov A.V.* Generacija testovyh zadaniy na osnove grammatik v sisteme postroenija generatorov na derev'jah I / ILI / A.V. Titkov // *Materialy Mezhdunarodnoj naucho-metodicheskoj konferencii «Sovremennoe obrazovanie».* – Tomsk, 2010. – S. 163–164.
5. *Kruchinin V.V.* Organizacija samostojatel'noj raboty studentov s ispol'zovaniem tehnologii generacii testovyh zadaniy / V.V. Kruchinin, L.I. Magazinnikov, Ju.V. Morozova // *Sistemy avtomatizacii v obrazovanii, nauke i proizvodstve.* – Novokuzneck : SibGIU, 2005. – S. 351–353.
6. *Zorin Ju.A.* Instrumental'nye sistemy postroenija i poluchenija mnogovariantnyh testovyh zadaniy / Ju.A. Zorin, I.A. Posov // *Komp'yuternye instrumenty v obrazovanii.* – 2014. – № 1. – S. 14–25.