

ПРИМЕНЕНИЕ ГЕНЕРАТОРОВ В КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЯХ ОБУЧЕНИЯ

С.И. Борисов, ведущий программист лаборатории инструментальных систем моделирования и обучения Томского университета систем управления и радиоэлектроники,

В.В. Кручинин, зав. лабораторией инструментальных систем моделирования и обучения Томского университета систем управления и радиоэлектроники, зам. директора по научной работе Томского межвузовского центра дистанционного образования

В статье рассматривается класс программ учебного назначения, предназначенных для генерации вопросов. Описываются обобщенная структура и алгоритм работы. Приводятся основные свойства генераторов и конкретные примеры генераторов для организации тестового контроля. Показана технология использования генераторов в компьютерных тренажерах и учебниках на основе опыта Томского межвузовского центра дистанционного образования.

A class of education programs, designed to use for test questions generation is considered. Generalized structure and operation algorithm are described. Basic properties of the generator program(s) are listed. Examples of generator programs are provided, ready to apply for educational knowledge testing process organization. A tested technology of generator programs as applied in computer training programs and textbooks is considered. Experience of generator(s) application in a real educational institution in Tomsk interuniversity center of distance education is given for consideration.

Одним из важнейших элементов технологии обучения является действенный контроль знаний. Организация контроля знаний (входного, промежуточного и итогового) не обходится без выдачи некоторого тестового задания. Если количество этих заданий сравнительно невелико, то за короткое время может быть создан компьютерный банк ответов, решений и т.п. Подтверждением тому служат многочисленные сайты, распростра-

няющие рефераты. Имеются сайты, которые предоставляют услуги по решению задач, курсовых проектов и пр., что также приводит к созданию соответствующего банка ответов и решений.

Проблема плагиата в Интернете — одна из сложнейших¹. Некоторые фирмы предлагают программные системы, которые могут анализировать текст на предмет заимствования². Можно также использовать поисковые сервера, вста-

© С.И. Борисов, В.В. Кручинин, 2004

вив в поля окна поиска требуемый фрагмент. Однако это требует существенных затрат на организацию соответствующей проверки.

Опыт, накопленный в Томском университете систем управления и радиоэлектроники (ТУСУР) при организации дистанционного обучения, показывает, что наиболее действенной является выдача уникального, индивидуального задания каждому студенту. Это достигается за счет применения программных генераторов тестовых заданий.

Обобщенная структура генератора

Генератор — это программная система, которая синтезирует тестовое задание. Обобщенная структура такой системы показана на рис. 1.

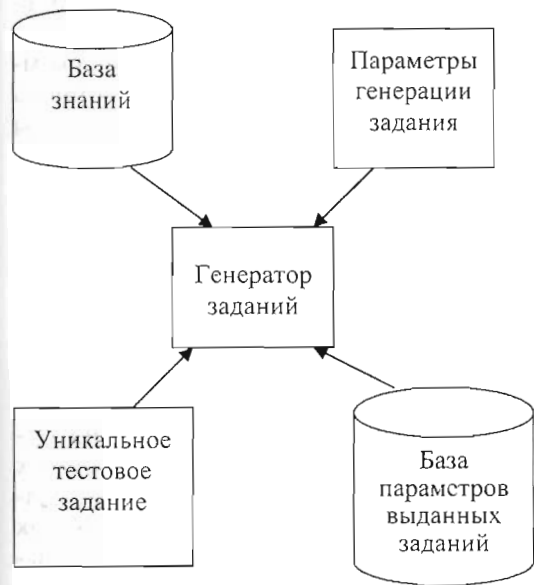


Рис. 1

База знаний содержит полную информацию для генерации задания. Если задание — задача, то база знаний содержит банк шаблонов формулировок задач, области изменения параметров задач, алгоритм нахождения решения или проверки наличия решения, алгоритм изменения значений параметров. Если задание — последовательность вопросов, то

могут использоваться разнообразные модели и алгоритмы генерации вопросов³.

Параметры генерации задания задают начальные значения параметров генерации и ограничения на тип генерируемых заданий. Начальные значения параметров генерации могут быть получены на основании: 1) датчика случайных чисел; 2) индивидуальных параметров студента: фамилия, имя, отчество, место и дата рождения, семестр и пр.; 3) некоторого унифицированного кода, присваиваемого студенту при поступлении в вуз.

Уникальное тестовое задание является конечной целью работы генератора и представляется на некотором декларативном языке для последующего этапа визуализации. Например, генераторы в системе «Фея» формируют тестовое задание на языке представления теста этой системы⁴. Можно также представлять тестовые задания на языке Tex⁵ или XML⁶.

База параметров выданных заданий предназначается для двух основных целей:

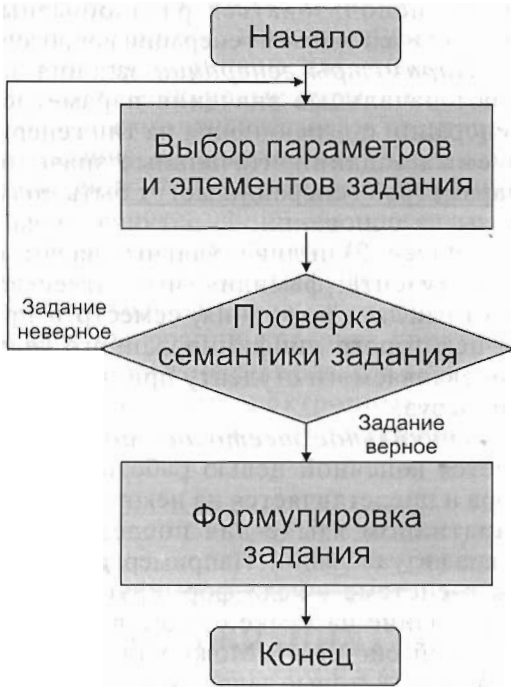
1) идентификации задания, выполненного студентом. Опыт, накопленный в Томском межвузовском центре дистанционного образования (ТМЦДО), показывает, что некоторые студенты присылают ответы или решения не на те задания, которые им были выданы;

2) оперативного анализа и исправления ошибок. Генераторы — сложные программы, отладка которых является достаточно сложным делом. Проверить его работоспособность во всех случаях практически невозможно.

Алгоритм работы генератора (рис. 2) зависит от типа задания и моделей представления знаний.

Рассмотрим основные шаги работы алгоритма.

1. Выбор параметров и элементов задания — это, собственно говоря, шаг, на котором производится синтез задания. Выбор осуществляется, как правило, на основе использования датчика случайных чисел, что в некоторых случаях существенно снижаст параметры



Р и с. 2

генерирующего алгоритма. Если параметры генерировать от входных параметров студента, то необходимо подбирать хеш-функции, которые вычисляли бы значения ключей. Хорошие результаты дает использование алгоритмов нумерации, которые перечисляют все варианты заданий данного алгоритма генерации⁷.

2. Проверка семантики задания — важный шаг, поскольку не всякая комбинация параметров может привести к решению задачи. В простейшем случае следует проверить некоторую формулу, например: подкоренное выражение не должно быть меньше нуля. В более сложных случаях — разработать специальный алгоритм семантического контроля.

3. Формулировка задания — шаг, на котором производится преобразование внутреннего представления тестового задания на некоторый декларативный язык. Поскольку текст задания может содержать формулы, рисунки, таблицы и пр., то нужно использовать язык описания документа для какого-либо конкретного пакета.

Отметим, что если задание можно разбить на части, то приведенный обобщенный алгоритм (см. рис. 2) применим к каждой отдельной части, поэтому весь генератор может представлять собой многократную итерационную схему.

Перечислим основные свойства генератора:

- 1) мощность;
- 2) управление последовательностью генерации;
- 3) управление сложностью тестового задания.

Мощность генератора — это количество возможных вариантов заданий, которые он может получить. При разработке генератора нужно оценить его мощность, в некоторых случаях это трудно сделать, поскольку связь между отдельными генерируемыми элементами задания может оказаться довольно сложной.

При проведении тестирования в компьютерных классах тестовые задания на двух соседних компьютерах должны быть, по крайней мере, не похожими друг на друга. В результате встает задача установки семантического расстояния между двумя сгенерированными заданиями и управления генерацией с учетом этого расстояния.

Еще одним важным свойством генератора является возможность управления сложностью задания. В простейших случаях генератор выдает задания одной и той же сложности. В некоторых технологиях требуется генерировать задания с разной сложностью, начиная, как правило, с простых и заканчивая самыми сложными⁸.

Генераторы для тестового контроля знаний

Задача построения генераторов тестовых заданий и вопросов возникла при организации тестового контроля знаний в дистанционной технологии обучения. Опыт создания и использования компьютерных контрольных работ и экзаменаторов, накопленный в Томском межвузовском центре дистанционного образования⁹, показал, что:

1) студенты быстро приспособиваются к небольшому (100—300 вопросов) компьютерному экзамену, заготавливают шпаргалки;

2) происходит простое механическое запоминание ответов на данный вопрос, поскольку в ответ нужно ввести конкретное число или выбрать конкретный вариант;

3) при одновременной сдаче экзамена в компьютерном классе вопросы для разных студентов могут быть одни и те же.

Таким образом, уменьшается эффективность проведения экзаменационных сессий и контрольных точек. Для устранения указанных недостатков было предложено использовать генераторы тестовых заданий и вопросов.

Рассмотрим пример, в котором нужно вычислить определитель 5-го порядка¹⁰.

В общем случае для формулировки данной задачи необходимо сгенерировать соответствующую матрицу, дискриминант которой должен быть не равен нулю. Далее, применяя алгоритм вычисления определителя, найти его значение. Использовать эту задачу для тестирования студентов автор нашел нецелесообразным и предложил вариант матрицы специального вида:

$$\begin{vmatrix} 0 & b & c & a_1 & a_2 \\ 1 & x & y & z & u \\ 0 & b & c+1 & a_3 & a_4 \\ 0 & b & c+2 & a_5 & a_6 \\ 0 & b & c+3 & a_7 & a_8 \end{vmatrix}$$

Для такого вида матрицы определитель считается по формуле

$$D = -b((a_5 + a_1 - 2a_3)(a_8 + a_4 - 2a_6) - (a_7 + a_3 - 2a_5)(a_6 + a_2 - 2a_4)), \quad (1)$$

где генерируются параметры $b, c, x, y, z, u, \{a\}$, причем параметры c, x, y, z, u могут быть любыми, b не равно нулю. Множество параметров генерируется из условия неравенства нулю определителя. Зная свойства разложения матрицы при вычислении определителя, сравни-

тельно легко найти решение поставленной задачи. Тогда алгоритм генератора будет следующий.

Шаг 1. Генерируем любые значения для параметров c, x, y, z, u .

Шаг 2. Генерируем значения для b и множества параметров $\{a\}$.

Шаг 3. Вычисляем значение D по формуле (1).

Шаг 4. Если D равно нулю, то повторяем шаг 2.

Шаг 5. Формулируем задание с полученными параметрами и записываем правильное решение.

Данный пример показывает, что алгоритм решения задачи может быть существенно упрощен. Но количество вариантов заданий является огромным числом, даже при таком упрощении. Например, если каждый значимый параметр будет иметь по 10 значений, то общее число вариантов может быть порядка 10^9 .

Другой пример взят из генератора компьютерных экзаменов по информатике¹¹.

Дан целочисленный массив длиной 8. Массив необходимо отсортировать по возрастанию. Массив дан после n шагов пузырьковой сортировки. Введите через пробел элементы массива после $n + 1$ шага.

n изменяется от 1 до 5, первые n элементов уже отсортированы. Элементы массива случайным образом задаются в диапазоне от 1 до 99.

Ниже приведен алгоритм сортировки на языке Паскаль.

{сортировка пузырьковым методом}
procedure Bubble;

var

i, j: index;

x: item;

begin

for i := 2 to n do

begin

for j := n downto i do

if a[j-1].key > a[j].key then

begin

x := a[j-1];

a[j-1] := a[j];

a[j] := x;

end;

end;

end; {конец сортировки пузырьковым методом}



Генерируя случайно значения для элементов массива и значения для n , получим конкретное тестовое задание. Количество конкретных тестовых заданий будет вычисляться по формуле

$$v = 99^8 \cdot 5.$$

Это позволяет практически каждому студенту дать индивидуальное задание.

Важным дополнительным требованием к генераторам для тестовых заданий является необходимость получения правильного решения или ответа. Это накладывает новые ограничения на генераторы такого вида, а с другой стороны, позволяет автоматизировать контроль. Например, студент присылает по электронной почте протокол контрольной работы в форме компьютерного тестирования, который получает генератор. Этот протокол содержит параметры студента, контрольной работы и ответы на сгенерированные тестовые задания, введенные студентом. В центре протокол может быть открыт и просмотрен программой — почтовым роботом.

Генераторы заданий в тренажерах

Компьютерные тренажеры являются необходимым инструментом, позволяющим студентам закреплять знания по различным дисциплинам.

Генераторы в тренажерах также играют существенную роль, однако цели их применения другие. Они позволяют:

1) каждый раз получать текст задачи с новыми числовыми данными. Благодаря этому студент не привыкает к конкретному числу, а учит алгоритм решения;

2) управляя классами задач и их сложностью, определить слабые места студента и тем самым способствовать более полному изучению предмета;

3) существенно переработать методическое обеспечение курса, что приводит к более глубокому пониманию сути предмета и методам написания и представления методического обеспечения.

Рассмотрим генерацию задания для тренажера «Найти предел функции» в компьютерном учебнике по математике¹².

В общем виде функция может выглядеть, например, следующим образом:

$$\lim_{x \rightarrow g-0} \frac{\sin(ax^n + b)}{cx^m + d}.$$

В данном случае присутствуют 7 переменных, для которых генерируются значения в различных диапазонах, например:

$$b, g \in [-9, 9]; \quad a, c \in [1, 9]; \quad n, m \in [1, 3].$$

Отличие от генерации аналогичных тестовых заданий заключается в том, что пользователь (студент) может сам задать начальные значения (например, если ему встретилась аналогичная задача в его контрольной работе) либо сгенерировать их автоматически. На рис. 3 показана форма для запроса у студента начальных данных.

Присвойте параметрам значения:

The form consists of seven input fields arranged in three rows. The first row contains fields for $a=1$, $n=1$, and $b=0$. The second row contains fields for $c=1$, $m=1$, and $d=0$. The third row contains a field for $g=0$. Below these fields are two buttons: 'Ok' and 'Генерировать'.

Рис. 3

Разумеется, студент может задать различные данные, в том числе такие, для которых решение задачи не имеет смысла или невозможно. Чтобы избежать этого, предусмотрены специальные способы контроля.

Для указанного примера введем функции $P(x) = \sin(ax^n + b)$ и $Q(x) = cx^m + d$. Для контроля введем правило: если $P(g) \cdot Q(g) \neq 0$.

Пусть, например, были введены следующие начальные данные: $a = 3$, $n = 1$, $b = 0$, $c = 1$, $m = 1$, $d = 0$, $g = 0$. Тогда функция принимает вид:

$$\lim_{x \rightarrow 1+0} \frac{\sin(3x)}{x^3 - 1}.$$



Заключение

В настоящее время в ТМЦДО внедрены и эксплуатируются свыше 20 генераторов тестовых заданий общим объемом более 2 000 шаблонов и алгоритмов, для различных дисциплин, таких как математика, физика, иностранный язык, микроэлектроника, цифровая схемотехника, информатика. Опыт показывает, что различные базы правильных ответов и решений, в том числе и шпаргалок, становятся ненужными. В результате студенты не ищут готовых решений, а основательно готовятся к указанным предметам.

Разработаны и внедрены 2 компьютерных учебника по высшей математике, охватывающие соответственно 2 семестра изучения этой дисциплины в техническом вузе. В этих компьютерных учебниках реализованы около 180 тренажеров. Во всех тренажерах присутствует генерация начальных данных по указанной технологии.

Генераторы становятся одним из важных классов компьютерных учебных программ, поднимающих компьютерные технологии обучения на более высокий уровень.

ПРИМЕЧАНИЯ

¹ См.: Кузьменков О.И. Проблемы сетевого плагиата (<http://www.library.by/shpargalka/belarus/007/657.htm>).

² См.: Groark M., Oblinger D., Choa M. Term paper Mills, Anti-Plagiarism Tools, and Academic Integrity // EDUCOUSE Review / 2001. Sept./Oct.

P. 40—48; Mike H., Friesner T. Plagiarism and Poor Academic Practice — A Threat to the Extension of e-Learning in Higher Education? // Electronic Journal of e-Learning. V. 2. Iss. 1 (Febr. 2004). P. 89—96 (<http://www.ejel.org/volume-2/vol2-issue1/issue1-art25-hart-friesner.pdf>).

³ См.: Башмаков А.И., Башмаков И.А. Разработка компьютерных и обучающих систем. М., 2003; Кручинин В.В. Разработка компьютерных учебных программ. Томск, 1998.

⁴ См.: Кнут Д. Компьютерная типография. М., 2003.

⁵ См.: Mathematical Markup Language (MathML) Version 2.0 (Second Edition) (<http://www.w3.org/TR/MathML2>).

⁶ См.: Дмитриева Н.М., Кручинин В.В., Ситникова Е.А. Технология разработки компьютерных учебных работ и экзаменаторов в Томском межвузовском центре дистанционного образования // Тез. докл. семинара «Электронные учебники и учебно-методические разработки в открытом образовании». М., 2000. С. 69—70.

⁷ См.: Кручинин В.В. Генераторы в компьютерных учебных программах. Томск, 2003.

⁸ См.: Аванесов В.С. Теория и методика педагогических измерений (<http://testolog.narod.ru>).

⁹ См.: Организация всестороннего контроля знаний студентов в дистанционной технологии обучения / Кручинин В.В. и др. // Новые информационные технологии в университетском образовании: Тез. докл. междунар. науч.-метод. конф. Новосибирск, 2003. С. 72—73.

¹⁰ См.: Кручинин В.В., Магазинников Л.И. О повышении качества контролируемых материалов // Материалы науч.-метод. конф. Томск, 2002. С. 69.

¹¹ См.: Молочко М.В., Тилченко С.В. Генераторы по информатике в компьютерных экзаменах // Материалы регион. науч.-техн. конф. «Современное образование: инновации и конкурентоспособность». Томск, 2004. С. 105.

¹² См.: Борисов С.И., Дмитриева Н.М., Долматов А.В., Кручинин В.В., Ситникова Е.А., Томиленко В.А. Компьютерный учебник по математике. Ч. 1 // Тез. докл. семинара «Электронные учебники и учебно-методические разработки в открытом образовании». С. 46—48.

Поступила 30.08.04.