Научный вестник НГТУ – 2004. – № 3(18)

УДК 519.171: 371.263

Использование графов для построения генераторов вопросов в компьютерных учебных программах*

В.В. КРУЧИНИН

Рассматриваются вопросы использования двудольных и древовидных графов для построения генераторов вопросов в компьютерных учебных программах. Приведены абстрактные модели и конкретные примеры графов, показаны примеры полученных вопросов, которые могут быть использованы как при создании банка вопросов для тестирующих программ, так для создания генераторов вопросов. Получены алгоритмы подсчета количества и генерации вопросов по заданному номеру. Предложенные модели и алгоритмы апробированы в Томском межвузовском центре дистанционного образования (ТМЦДО).

1. ВВЕДЕНИЕ

В настоящее время компьютерные технологии обучения прочно вошли в учебный процесс вуза. Компьютерные учебные программы, такие как мультимедийные учебники, виртуальные лабораторные работы, тесты для контроля знаний, становятся важными элементами современной технологии обучения. В таких программах необходима организация интерактивного режима. Однако зачастую диалог между компьютерной учебной программой и студентом носит тривиальный характер. Вопросы, как правило, фиксированы, а их количество не превышает нескольких десятков.

Выходом их создавшейся ситуации может служить использование генераторов вопросов – специальных программ на основе формализованного представления знаний данной предметной области, синтезирующих вопросы для компьютерного теста [1, 2].

Перечислим основные свойства генератора вопросов [3]: мощность множества генерируемых вопросов, управляемость последовательностью генерацией вопросов, сложность реализации. Все эти свойства существенно зависят от моделей предметной области, используемых для построения генератора.

Важно также отметить, что использование датчика случайных чисел для генерации вопросов нежелательно, так как при достаточно большой мощности генератора можно получить одинаковые вопросы. Поэтому предлагается разрабатывать алгоритмы генерации, основанные на перечислении вопросов.

Отдельные модели и алгоритмы генерации вопросов обсуждались в литературе. Использование шаблонов для генерации задач и меню вопросов рассматривалось в [1, 3, 5], реляционных моделей – в [6], формальных грамматик в – [1].

Ниже предлагаются достаточно простые модели и алгоритмы генерации вопросов, основанные на использовании представления модели предметной области в виде двудольного графа и дерева.

---

* *Статья получена 3 сентября 2004 г.*

2. ГЕНЕРАТОР ВОПРОСОВ ДЛЯ НЕКОТОРОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ДЕЙСТВИЙ (ПРОЦЕССА, ТЕХНОЛОГИИ)

Пусть дана последовательность $\{x_1, x_2, x_3, ..., x_n\}$, например некоторая последовательность действий. Назовем эту последовательность процессом, а элементарное действие – этапом (шагом). Для каждого этапа процесса имеется список начальных условий, необходимых для осуществления данного этапа, список особых условий завершения этапа процесса. Даны также параметры выполнения этапа. По окончании процесса будет получен некоторый результат *r*. На рис. 1. показан граф, описывающий этап процесса. Есть начальный набор условий для выполнения этапа {*u*0, *u*1, *u*2}. После выполнения этапа будет получен промежуточный результат (*r*1).

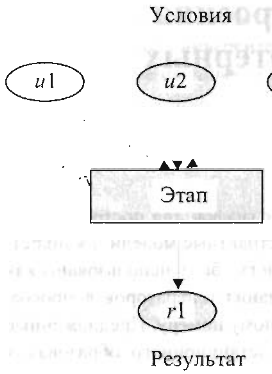

Рис. 1. Описание этапа процесса

Тогда весь процесс можно описать с помощью двудольного графа [7], в котором все вершины разбиваются на два непересекающихся множества. Первое множество – это этапы процесса. Второе множество – условия и результаты этих этапов. На рис. 2 показан пример такого графа. Вершины *u*1, *u*2 и *u*3 необходимы для выполнения первого этапа, вершины *u*4 и *r*1 – условия выполнения второго, а вершины *r*2 и *u*5 – третьего этапов. Для четвертого этапа необходимо получить *r*3 и *r*4 и задать *u*6. Результатом всего процесса будет вершина *r*5.

Примерами описаний подобных процессов могут быть разнообразные технологии приготовления продуктов, химических производств, сборки устройств, разнообразных операций и т.п.

Данная модель процесса позволяет описать последовательность действий:

1. Какой этап следует после *x*[*i*] … (название) или в форме меню {*x*[*k*1], *x*[*k*2], *x*[*k*3], *x*[*k*4]}.
2. Какой этап следует перед *x*[*i*] … (название) или в форме меню {*x*[*k*1], *x*[*k*2], *x*[*k*3], *x*[*k*4]}.
3. Даны следующие этапы процесса {*x*[*k*1], *x*[*k*2], *x*[*k*3], *x*[*k*4]}, указать правильную последовательность их выполнения.
4. Указать условия, выполнение которых необходимо для осуществления x[k] этапа процесса {*u*[*k*1], *u*[*k*2], *u*[*k*3], *u*[*k*4]}.
5. Дан промежуточный результат *r*[*k*] выполнения процесса. Указать этап, получающий данный результат {*x*[*k*1], *x*[*k*2], *x*[*k*3], *x*[*k*4]}.
6. Указать недостающее условия для осуществления этапа *x*[*k*] {*u*[*k*1], *u*[*k*2], *u*[*k*3], *u*[*k*4]}.
7. Дан список промежуточных результатов {*r*[*k*1], *r*[*k*2], *r*[*k*3], *r*[*k*4]}, упорядочить их в соответствии с заданным процессом.

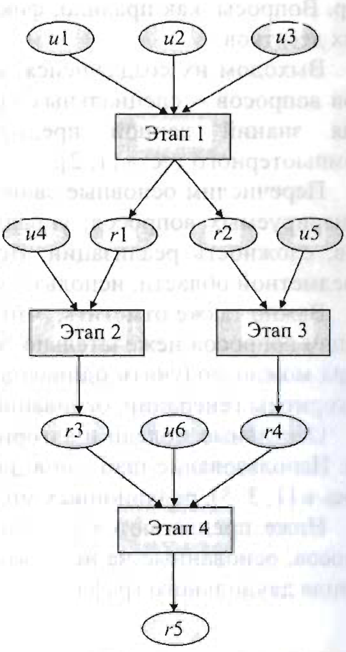

Рис. 2. Описание процесса

Рассмотрим использование данного подхода на конкретном примере. Пусть дана технология (рецепт) приготовления блюда (ботвиньи) [8] (рис. 3), представленная в форме двудольного графа. Прямоугольниками отмечены технологические операции (отварить, перебрать, промыть, нарезать и т.д). Овалами изображены ингредиенты и результаты этих операций (рыба, вода, специи, вареная рыба, бульон и т.д.). Стрелка от овала к прямоугольнику означает, что этот ингредиент необходим для выполнения данной операции. Стрелка от прямоугольника к овалу означает, что данный овал является результатом этой операции.

Тогда можно сформулировать следующие вопросы:

1. *Для выполнения операции «Припустить» необходимы:*

1) бульон;
2) специи;
3) шпинат;
4) смесь.

Укажите верные варианты.

2. *Для выполннения операции «Отварить» необходимы:*

1) сахар;
2) вода;
3) квас;
4) рыба.

Укажите верные варианты.

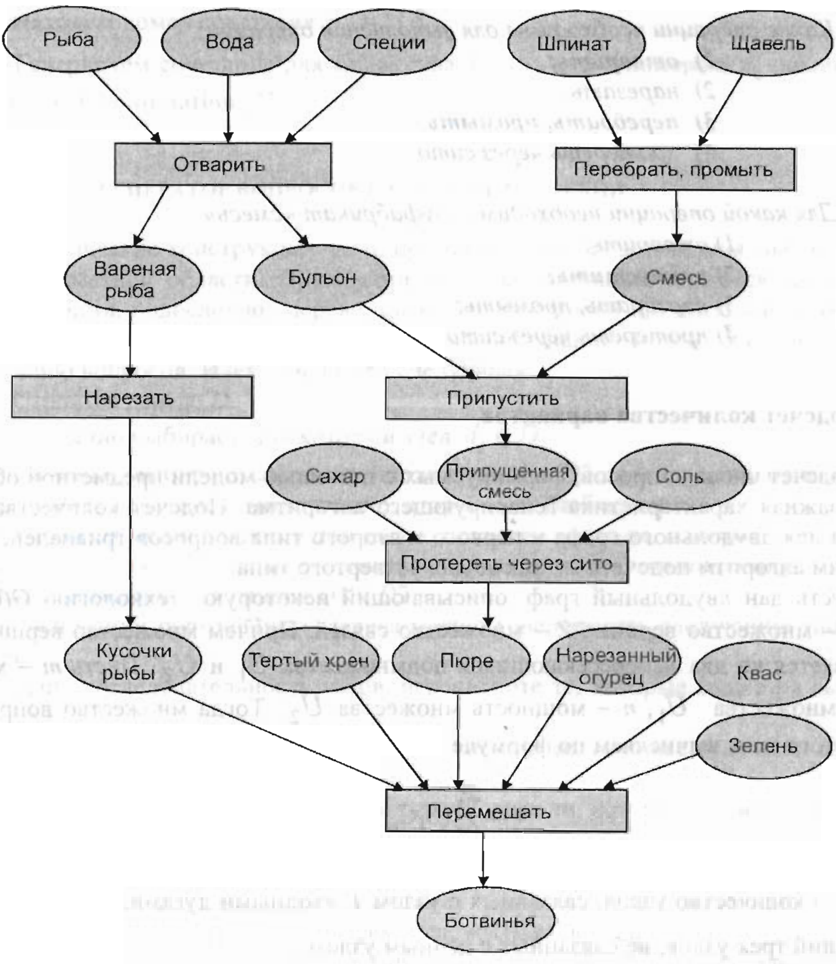

Рис. 3. Описание технологии приготовления ботвиньи

3. *Результатом операции «Припустить» будет:*
 1) пюре;
 2) припущенная смесь;
 3) тертый хрен;
 4) нарезанный огурец.

4. *Указать операцию, результатом которой будет бульон:*
 1) нарезать;
 2) перебрать, промыть;
 3) припустить;
 4) отварить.

5. *Даны следующие операции:*
 1) припустить;
 2) перебрать, промыть;
 3) перемешать;
 4) протереть через сито.

Укажите порядковыми номерами правильную последовательность выполнения операций.

Правильный ответ: 2, 1, 4, 3.

6. *Какие операции необходимы для выполнения операции:*
 1) отварить;
 2) нарезать;
 3) перебрать, промыть;
 4) протереть через сито.

7. *Для какой операции необходим полуфабрикат «смесь»:*
 1) отварить;
 2) припустить;
 3) перебрать, промыть;
 4) протереть через сито.

Подсчет количества вариантов

Подсчет числа вопросов, формируемых с помощью модели предметной области, – важная характеристика генерирующего алгоритма. Подсчет количества вопросов для двудольного графа и первого и второго типа вопросов тривиален. Рассмотрим алгоритм подсчета вопросов для четвертого типа.

Пусть дан двудольный граф, описывающий некоторую технологию $G(U,Z)$, где U – множество вершин, Z – множество связей. Причем множество вершин U разбивается на два непересекающихся подмножества U_1 и U_2. Пусть m – мощность множества U_1, n – мощность множества U_2. Тогда множество вопросов четвертого типа вычисляем по формуле

$$v = \sum_{i=1}^{m} (S_i C_{n-S_i}^3),$$

где S_i – количество узлов, связанных с узлом i входными дугами, $C_{n-S_i}^3$ – число сочетаний трех узлов, не связанных с данным узлом.

Алгоритм генерации вопроса

Пусть дан двудольный граф, описывающий некоторую технологию $G(U,Z)$ и некоторое целое число j, удовлетворяющее условию $0 \le j \le v$.

Необходимо получить j-й вопрос заданного типа.

1. Вычисляем параметры l и k по формуле

$$l = \min_k\left(\sum_{i=1}^{k}(S_i C_{n-S_i}^3) - j\right) > 0 .$$

2. Находим k-ю вершину из U_1, определяем подмножество U_2^k вершин, имеющих выходные дуги к k-й вершине. Удаляем указанное подмножество из U_2:

$$U_- = U_2 - U_2^k .$$

3. Находим номер элемента множества U_2^k – это будет правильный вариант ответа:

$$n_v = l \bmod S_i .$$

4. Находим номер сочетания $n_c = l / S_i$.

5. Генерируем сочетание для множества U_- по п. 3 с номером n_c, используя алгоритм ***G* Combunation** [3].

3. ГЕНЕРАТОР ВОПРОСОВ НА ОСНОВЕ ИЕРАРХИИ

Иерархические конструкции часто применяют для представления знаний некоторой предметной области. Примерами таких конструкций являются различные классификации, родословное дерево, структура некоторой организации, устройства и др. Используя иерархическое представление знаний, можно построить генератор меню вопросов. Идея генератора следующая:

1. Строится дерево D.

2. Случайно выбирается некоторый узел $d_i \in D$.

3. Случайно выбираются два подмножества $D1$ и $D2$ других узлов, одно из них удовлетворяет некоторому отношению, второе – не удовлетворяет.

4. Строится меню вопрос: *«Дан узел d, укажите те узлы, которые удовлетворяют (или не удовлетворяют) отношению <**формулировка отношения**>»*.

Далее записываются список $D1$ и $D2$.

На основании построенного дерева можно генерировать следующие типы вопросов:

1. Дана последовательность узлов, перечислите те, которые являются сыновьями узла *d*.

2. Дана последовательность узлов, перечислите те, которые относятся к одному уровню иерархии.

3. Дана последовательность узлов, перечислите те, которые принадлежат поддереву узла *d*.

4. Дана последовательность узлов, перечислите те, которые являются листьями.

5. Дана некоторая последовательность *a, b, {x, y, z}, d* (выберите один из узлов *x, y, z*, который превратит эту последовательность в след дерева).

На рис. 4 показана некоторая иерархическая классификация земноводных [9]. Используя эту иерархию, можно генерировать следующие типы вопросов.

Тип вопроса 1

Выбираются случайно элементы иерархии 1.1, 1.2, 1.3. Перечислите все относящиеся к заданной группе (классу).

Перечислены следующие земноводные:

1) рыбозмеи,

2) углозубы,

3) протеи,

4) свистуны.

Укажите среди них безногих.

Рис. 4. Пример иерархической классификации

Тип вопроса 2

Выбирается случайно один из листьев этой иерархии и перечисляются классы. Необходимо указать класс для выбранного элемента.

Укажите класс земноводных, к которому относятся рыбозмеи:

1) безногие;

2) хвостатые;

3) бесхвостые.

Подсчет числа вариантов вопросов

Приведем подсчет вариантов вопросов для первого типа.

Вариант 1. Среди предложенных узлов указать сына узла d.

Для подсчета необходимо использовать узлы дерева, не являющиеся листьями. Пусть их количество будет n. Общее число узлов m. Общее число сыновей будет $m - 1$ (корень дерева не является сыном). Подсчет числа дуг в дереве даст количество вариантов отношения *«x сын y»*. Список узлов, не являющихся сыновьями данного узла размером k, можно сформировать на основе сочетания общего числа узлов, не являющихся сыновьями данного узла по k:

$$v = \sum_{i=1}^{n} L_i C_{m-L_i-1}^{k},$$

где L_i – количество сыновей i-го узла; $C_{m-L_i-1}^{k}$ – число сочетаний из множества узлов не являющихся сыновьями i-го узла по k.

Вариант 2. Среди предложенных узлов указать сыновей узла d.

В этом случае правильных вариантов может быть несколько.

Тогда общее число вариантов можно подсчитать по формуле

$$v = \sum_{i=1}^{n} (2^{L_i} - 1) C_{m-L_i-1}^{k}.$$

При формулировке вопроса для i-го узла может быть сформировано $2^{L_i} - 1$ правильных вариантов ответов (формула подсчета множества подмножеств).

Вариант 3. Среди предложенных узлов указать всех сыновей узла d.

В списке вариантов должны быть перечислены все сыновья данного узла, тогда формула будет следующей:

$$v = \sum_{i=1}^{n} L_i C_{m-L_i-1}^{k}.$$

При условии, что альтернатив в меню должно быть фиксированное количество, формула усложняется.

Алгоритм генерации

Дано: дерево, где m узлов и n узлов, не являющихся листьями, некоторое число j, удовлетворяющее условию

$$0 \le j \le v.$$

Необходимо получить вопрос по первому типу второго варианта (общий случай).

1. Находим такое l, чтобы $j < \sum_{i=1}^{l} (2^{L_i} - 1) C_{m-L_i-1}^{k}$, $l=1,n$; l-будет показывать номер узла для формирования меню вопроса.

2. Записываем множество сыновей l-узла и подсчитываем новое значение для j по формуле

$$j = j - \sum_{i=1}^{l-1} (2^{L_i} - 1) C_{m-L_i-1}^{k}.$$

3. Получаем номер подмножества

$$j_1 = j \bmod (2^{L_l} - 1).$$

4. Получаем номер сочетания

$$C_l = j / (2^{L_l} - 1).$$

5. Генерируем подмножество и сочетание, используя алгоритмы генерации, описанные в [3]. Формируем вектор и перемешиваем его.

6. Записываем меню вопрос: «*из приведенного списка узлов, перечислите сыновей l-узла*».

Узел $k1$, узел $k2$, узел $k3$ … узел Kk.

7. Записываем правильные варианты ответа.

ЗАКЛЮЧЕНИЕ

Реализация описанных выше моделей генерации вопросов в реальных компьютерных учебных программах и внедрение этих программ в практику дистанционного обучения позволила существенно повысить эффективность тестового контроля знаний. С одной стороны, генераторы обеспечивают выдачу индивидуальных заданий и вопросов каждому студенту, с другой – преподаватели, освоившие данные модели, сравнительно легко получают банки вопросов для создания тестирующих программ по своим курсам. В настоящее время в ТМЦДО уже создано и внедрено свыше 30 генераторов вопросов и тестовых заданий.

Дальнейшее развитие этого направления приведет к появлению нового класса языков и компьютерных систем, облегчающих формирование различных банков вопросов.

СПИСОК ЛИТЕРАТУРЫ

[1] **Кручинин В.В.** Разработка компьютерных учебных программ. – Томск: Изд-во Томск. ун-та, 1998.

[2] **Башмаков А.И., Башмаков И.А.** Разработка компьютерных учебников и обучающих систем. – М.: Информационно-издательский дом «Филинъ», 2003.

[3] **Кручинин В.В**. Генераторы в компьютерных учебных программах. – Томск: Изд-во Томск. ун-та, 2003.

[4] **Егоркина Ю.В., Кручинин В.В., Шарапов А.В**. Пакет генераторов тестовых заданий по циклу «Цифровые микропроцессорные устройства»// Современное образование: Интеграция учебы, науки производства: Мат-лы рег. науч.-метод. конф. – Томск: Изд-во ТУСУР, 2003. С. 86–87.

[5] **Кручинин В.В., Морозова Ю. В.** Модели и алгоритмы генерации задач в компьютерном тестировании // Изв. ТПУ. – 2004. – № 5. – С. 127–131.

[6] **Кручинин В.В., Морозова Ю.В.** Модели генераторов вопросов для компьютерного контроля знаний//Открытое и дистанционное образование. – 2004. – Вып. 2(14). – С. 36–42.

[7] **Евстигнеев В.А**. Применение теории графов в программировании. – М.: Наука, 1985.

[8] http://kuking.net/2_86.htm.

[9] Жизнь животных. – Т. 5: Земноводные. Пресмыкающиеся. – М.: Просвещение, 1985.

Кручинин Владимир Викторович, кандидат технических наук, заведующий лабораторией инструментальных систем моделирования и обучения Томского университета систем управления и радиоэлектроники, заместитель директора по научной работе Томского межвузовского центра дистанционного образования.

Имеет свыше 50 публикаций, в том числе две монографии. Основное направление научных исследований – системное программирование, трансляторы, применение методов искусственного интеллекта в компьютерных учебных программах, инструментальные системы и среды для автоматизированного обучения.