

УДК 004.5 + 004.588

ББК 32.973.26-018.2

К АВТОМАТИЗИРОВАННОЙ ПРОВЕРКЕ РЕШЕНИЙ ОДНОГО КЛАССА ЗАДАЧ В СЛЕДЯЩИХ ИНТЕЛЛЕКТУАЛЬНЫХ ОБУЧАЮЩИХ СИСТЕМАХ

Смирнова Н. В.¹

*(Учреждение Российской академии наук
Институт проблем управления РАН, Москва)*

Рассматриваются обучающие программы, обеспечивающие интерактивную поддержку процесса решений задач предметов естественно-научного цикла и их автоматизированную оценку. Для решения проблемы измерения прогресса в решении задачи предлагается оригинальная совокупность алгоритмов и эвристик. С помощью ряда примеров демонстрируются ее преимущества по сравнению с известным алгоритмом.

Ключевые слова: следящая интеллектуальная обучающая система, автоматизированная обучающая система, автоматическая проверка решений

1. Введение

Следящими интеллектуальными обучающими системами (следящими ИОС, англ. model-tracing intelligent tutoring systems) называются программы, предназначенные для обучения предметам естественно-научного цикла, обеспечивающие интерактивную поддержку процесса решения задач обучаемым и автоматизированную оценку его решений. Название «следящие ИОС» обусловлено тем, что для проверки хода решения обучаемого на завершенность и правильность необходимо сверять

¹ *Наталья Викторовна Смирнова, научный сотрудник (smirvanatalia2008@gmail.com).*

шаги решения обучаемого с шагами имеющихся в ИОС решений.

К одному из наиболее распространенных типов следящих ИОС относятся программы, в которых в качестве решения требуется ввести одну формулу или заполнить текстовые поля, соответствующие недостающим частям некоторой формулы. Примером такой системы является STACK [5]: подсистема, разработанная для автоматического оценивания заданий по алгебре и началам анализа [6], которая может быть интегрирована в другие обучающие среды, например, в Moodle [10]. В STACK проверка ответа обучаемого осуществляется с использованием математической среды Maxima. Из выражения, введенного обучаемым, вычитается «эталонное» выражение. Если полученное выражение удастся упростить до 0, то этим устанавливается алгебраическая эквивалентность выражения, введенного обучаемым, и «эталонного» выражения, т.е. обучаемый ввел правильный ответ. В качестве примера подобных систем можно упомянуть ActiveMath [4] и ряд электронных учебников по школьной математике, созданных на основе работ М.А. Левинской [1].

По-видимому, наиболее развитой программой, в которой реализована проверка решений, вводимых в достаточно свободной форме, является Andes Physics Tutor [7,9]. В данной статье предлагается совокупность алгоритмов и эвристик для измерения прогресса в решении, отличные от алгоритма, реализованного в Andes Physics Tutor. К основным преимуществам предлагаемых средств относятся: способность обрабатывать формулы, содержащие векторы и матрицы; более низкая уязвимость к имитации обучаемыми правильно введенных шагов решения.

Вкратце опишем содержание данной статьи. В ч. 2 очерчиваются проблемы, возникающие при проверке решений обучаемых в следящих ИОС и обосновывается необходимость разработки алгоритмов измерения прогресса в решении. В ч. 3 описывается известный алгоритм измерения прогресса в решении, реализованный в Andes Physics Tutor; с помощью примеров демонстрируется его уязвимость к имитации обучаемыми правильно введенных шагов решения. В ч. 4 рассматривается ос-

новой алгоритм измерения прогресса в решении. В ч. 5 предлагается несколько эвристик для обработки различных крайних случаев форм шага решения обучаемого.

2. Основные проблемы, возникающие при проверке решений в следящих ИОС

Далее рассматриваются определенные формальные задачи на нахождение искомых объектов (см. классификацию учебных задач в [3]). При этом предполагается, что:

- 1) решение обучаемого состоит из ряда *независимых* формул и не содержит естественно-языковых вставок (в частности, пока не рассматриваются задачи, для решения которых обучаемому необходимо решать системы уравнений),
- 2) при вводе формул студент может использовать только те обозначения, которые заложены в программу преподавателем для данной задачи и не может вводить свои обозначения.

Подсистема проверки решений ИОС решает следующие задачи:

- 1) верификация шага решения,
- 2) определение того, насколько далеко обучаемый продвинулся в решении, введя данный шаг.

По результатам верификации шага решения система сообщает обучаемому, верно ли последнее введенное им выражение. Сведения о прогрессе в решении необходимы в моменты обработки запроса подсказки и формирования оценки за решение.

Верификация шага решения осуществима с помощью простого эвристического алгоритма. Например, в Andes Physics Tutor используется прием «color by numbers», заключающийся в следующем. В формулу, введенную обучаемым, подставляются значения содержащихся в ней переменных. Далее выполняется проверка на равенство обеих частей получившегося выражения. Известно, что использование такого способа проверки некорректность подвыражений в формуле, введенной обучаемым, может быть не выявлена, если они умножаются на выражение,

которое равно 0. В Andes Physics Tutor предусмотрены специальные эвристики для выявления подобных случаев.

Проблема измерения прогресса в решении представляет большой интерес. Наиболее простой способ ее решения заключается в представлении известных решений задачи в виде списков формул и организации сравнения формулы, введенной обучаемым, с формулами из этих списков. При использовании этого способа возникает проблема множественности эквивалентных комбинаций формул, генерируемых обучаемыми при решении задач (МЭКФ).

Поясним проблему МЭКФ на примере задачи вычисления расстояния между векторами a, b в евклидовом пространстве. Для решения этой задачи необходимо использовать следующие формулы:

$$(1) \quad p(a, b) = |c|,$$

$$(2) \quad \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} a_1 - b_1 \\ a_2 - b_2 \end{pmatrix}.$$

$$(3) \quad |c| = \sqrt{(c, c)}.$$

$$(4) \quad (c, c) = c_1^2 + c_2^2.$$

Комбинация формул сводится к замене величин в одной формуле алгебраическими выражениями, полученными из других формул, участвующих в комбинации. Например, комбинация формул (1) и (3) имеет вид:

$$p(a, b) = \sqrt{(c, c)},$$

а комбинация формул (1), (3), (4) такова:

$$p(a, b) = \sqrt{c_1^2 + c_2^2}.$$

Можно представить себе много возможных решений задачи: решение, состоящее из формулы (1) и комбинации формул (2) и (4); решение, состоящее из формул (1), (4) и комбинации формул (2) и (3) и т.д. Как видим, использование этого «наивного» способа представления и обработки информации об известных решениях задач приводит как к значительному повышению трудоемкости как внесения информации, необходимой для проверки решений, так и добавления дополнительной информа-

ции (например, кратких текстовых подсказок к тем или иным формулам). Далее обсуждаются алгоритмы решения проблемы измерения прогресса в решении, при использовании которых не возникает проблемы МЭКФ.

3. Алгоритм измерения прогресса в решении, реализованный в Andes Physics Tutor

В Andes Physics Tutor шаги решения, вводимые обучаемыми, рассматриваются с точки зрения уравнений и зависимостей между ними. Если уравнение, введенное обучаемым, является линейной комбинацией предыдущих введенных им уравнений, значит, оно является лишь повторением уже введенных шагов решения. Сведения об известных решениях задач хранятся в виде списков «базовых» уравнений, на основе которых можно сгенерировать эти решения. Знание о линейной зависимости уравнения, введенного обучаемым и некоторых уравнений из списка «базовых» для этой задачи уравнений, свидетельствует о том, что обучаемый при вводе шага решения использовал именно эти уравнения.

Зависимость уравнений, в случае, когда все уравнения линейны, имеет место, если нормаль к гиперплоскости, задаваемой уравнением, введенным обучаемым, является линейной комбинацией нормалей гиперплоскостей, задаваемых остальными уравнениями. В нелинейном случае вместо коэффициентов каждого уравнения используются координаты его градиента, в котором все переменные заменены на числовые значения.

Рассмотрим этот алгоритм на разобранном ранее примере задачи вычисления расстояния между векторами. Отметим, что данным алгоритмом не предусматривается обработка формул, содержащих векторы и матрицы, поэтому в качестве списка «базовых» уравнений возьмем формулы (1), (3-6).

$$(5) \quad c_1 = a_1 - b_1,$$

$$(6) \quad c_2 = a_2 - b_2.$$

Пусть обучаемый ввел такую формулу:

$$(7) \quad p(a, b) = \sqrt{(c, c)}.$$

Для того, чтобы определить, какие формулы использовал обучаемый для вывода формулы (7), составим табл. 1. Строки табл. 1 соответствуют градиентам всех рассматриваемых формул в точке, координаты которой равны числовым значениям переменных из условий задачи.

Таблица 1. Градиенты «базовых» уравнений и формулы, введенной обучаемым, в точке, соответствующей решению задачи

Формула f	$grad f \Big _{c_1 = -2, c_2 = -2, (c, c) = 8}$
$p(a, b) - c $	$A = [1; 0; -1; 0; 0; 0; 0; 0; 0]$
$c_1 - a_1 + b_1$	$B = [0; 0; 0; 1; 0; -1; 0; 1; 0]$
$c_2 - a_2 + b_2$	$C = [0; 0; 0; 0; 1; 0; -1; 0; 1]$
$ c - \sqrt{(c, c)}$	$D = [0; -0,18; 1; 0; 0; 0; 0; 0; 0]$
$(c, c) - c_1^2 - c_2^2$	$E = [0; 1; 0; 4; 4; 0; 0; 0; 0]$
$p(a, b) - \sqrt{(c, c)}$	$F = [1; -0,18; 0; 0; 0; 0; 0; 0; 0]$

Далее решается система уравнений

$$(8) \quad \alpha A + \beta B + \chi C + \delta D + \varepsilon E = F.$$

Эта система переопределена (9 уравнений и 5 переменных). Путем комбинаторного перебора подмножеств уравнений этой системы генерируются квадратные системы уравнений. В случае успешного решения каждой полученной таким образом системы результаты запоминаются. В данном случае генерируется 18 квадратных систем, решение которых дает одинаковый набор значений $\alpha, \beta, \chi, \delta, \varepsilon$. Рассмотрим одну из систем (G - матрица системы, b - вектор свободных членов):

$$G = \begin{pmatrix} 0 & 0 & 0 & -0,18 & 1 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 4 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, b = \begin{pmatrix} -0,18 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Матрица G получена из совокупности 2, 3, 5, 8 и 9 уравнений системы (8). Решением данной системы уравнений является вектор

$$x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

Отличные от 0 координаты x имеют номера 1 и 4. Это означает, что для вывода формулы (7) обучаемый использовал формулы из 1 и 4 строки табл. 1, т.е. формулы (1) и (3).

Формула (7) соответствует шагу обучаемого в наиболее благоприятной для измерения прогресса в решении форме, поскольку в ней ни одна переменная из условия задачи не была заменена на соответствующее числовое значение. Очевидно, что чем больше переменных заменено на числовые значения и чем более вычислительно свернутой является формула, введенная обучаемым, тем большая неопределенность имеет место при измерении прогресса в решении задачи. Посмотрим, к каким результатам приводит использование вышеописанного алгоритма в случаях, когда в формуле, введенной обучаемым, часть переменных заменена на числовые значения.

ПРИМЕР 3.1.

Пусть обучаемый ввел формулу

$$(9) \quad (c, c) - (a_1 - b_1)^2 = 4.$$

Алгоритм выдаст правильный результат: совокупность формул (4-6), а также формулы $a_2 = 2$, $b_2 = 4$.

ПРИМЕР 3.2.

Пусть обучаемый узнал, что $(c, c) = 8$ (подобная ситуация может возникнуть в компьютерном классе после разговора между двумя сидящими рядом студентами) и ввел формулу
(10) $(c, c) = 16/2$.

Формула (10) корректна, но не может быть засчитана как шаг решения задачи. Она лишь является его имитацией. Между тем, алгоритм покажет, что при выводе формулы (10) обучаемый использовал совокупность формул (4-6), а также формулы $a_1 = 1, a_2 = 2, b_1 = 3, b_2 = 4$.

ПРИМЕР 3.3.

Пусть обучаемый узнал, что $(c, c) = 8$ и (c, c) некоторым образом зависит от c_1, c_2 и ввел формулу
(11) $(c, c) = c_1 + c_2 + 12$.

Поскольку $c_1 = -2, c_2 = -2$, то формула корректна, хотя и не может быть засчитана как шаг решения задачи. Между тем, алгоритм снова выдаст неправильный результат, аналогичный результату примера 2.

Как показывают примеры 3.2-3.3, в случаях, когда в формуле, введенной обучаемым, часть переменных заменена на числовые значения, алгоритм измерения прогресса в решении, реализованный в подсистеме проверки решений Andes Physics Tutor, становится *уязвимым* для имитации правильно введенных шагов решения.

4. Предлагаемый алгоритм измерения прогресса в решении: основной алгоритм

Известно, что осознанное решение любых учебных задач включает в себя анализ ситуации и выполнение некоторой последовательности вычислительных действий, в ходе выполнения которых обучаемый находит значения ряда величин, устанавливающих связь между данными и искомыми объектами. Будем называть эти величины и величины, соответствующие искомым объектам, целевыми. Естественно предположить,

что в большинстве шагов решения обучаемого будет присутствовать хотя бы одно обозначение целевой величины. Это предположение лежит в основе предлагаемого алгоритма.

Представим сведения о возможных решениях учебной задачи в виде набора отношений вычислимости одних величин через другие. Для приведенного ранее примера отношения вычислимости имеют следующий вид (см. табл. 2). В табл. 2 обозначение вида $x \rightarrow y$ расшифровывается следующим образом: «зная x , можно вычислить y по некоторой формуле». На отношения вычислимости накладываются следующие ограничения: правая часть отношения должна включать только одну переменную, соответствующую одной из «целевых» величин задачи и ничего более.

Таблица 2. Отношения вычислимости для задачи вычисления расстояния между векторами a и b .

Формула	Отношения вычислимости
$p(a,b) = c $	$C_1 : c \rightarrow p(a,b)$
$ c = \sqrt{(c,c)}$	$C_2 : (c,c) \rightarrow c $
$(c,c) = c_1^2 + c_2^2$	$C_3 : c_1, c_2 \rightarrow (c,c)$
$c = a - b$	$C_4 : a, b \rightarrow c$
$c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$	$C_5 : c_1, c_2 \rightarrow c$
$a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$	$C_6 : a_1, a_2 \rightarrow a$
$b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$	$C_7 : b_1, b_2 \rightarrow b$
$c_1 = a_1 - b_1$	$C_8 : a_1, b_1 \rightarrow c_1$
$c_2 = a_2 - b_2$	$C_9 : a_2, b_2 \rightarrow c_2$
«по c можно вычислить (c,c) »	$C_{10} : c \rightarrow (c,c)$

Идея предлагаемого алгоритма заключается в следующем. Сначала из формулы, введенной обучаемым, извлекаются $N_l^{(0)}$ – «целевая» переменная, которую вычисляет обучаемый в данном шаге решения и N_r – остальные переменные. Если в левой части формулы, введенной обучаемым, находится только одна переменная, то $N_l^{(0)}$ полагается равной этой переменной. Например, если шаг решения имеет вид $a = b + c$, то $N_l^{(0)} = \{a\}$, $N_r = \{b, c\}$. В остальных случаях $N_l^{(0)}$ полагается равной переменной, имеющей наибольший приоритет и не вычислявшейся в предыдущих шагах решения обучаемого. Сведения о приоритете «целевых» величин зависят от их порядкового номера по пути от данных объектов к искомым и вносятся в систему на этапе настройки сведений о возможных решениях учебной задачи. Например, в рассматриваемой учебной задаче наибольший приоритет имеет переменная $p(a, b)$, а наименьший – переменные

$$c_1, c_2, \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}.$$

Путем поиска по отношениям вычислимости $N_l^{(0)}$ преобразовывается до тех пор, пока не включит в себя N_r или не останется отношений вычислимости, способных преобразовать $N_l^{(i)}$ в $N_l^{(i+1)}$. В процессе поиска запоминаются соответствующие выбираемым отношениям вычислимости формулы – они и являются результатом работы алгоритма. Очевидно, что при условии конечного множества отношений вычислимости и ограничений на форму их записи (см. выше) предлагаемый алгоритм всегда завершает свою работу за конечное число шагов. Поясним работу алгоритма на примере.

ПРИМЕР 4.1.

Пусть обучаемый ввел формулу (7).

Шаг 1.

$$N_l^{(0)} = \{p(a, b)\}, N_r = \{(c, c)\}, I = \{p(a, b)\}, C = \emptyset.$$

Шаг 2.

$$N_l^{(1)} = \{|c|\}, I = \{p(a, b), |c|\}, C = \{C_1\}$$

Шаг 3.

$$N_l^{(2)} = \{(c, c)\}, I = \{p(a, b), |c|, (c, c)\}, C = \{C_1, C_2\}$$

Т.к. $N_l^{(2)} \equiv N_r$, алгоритм останавливается. Формулами, которые использовал обучаемый при вводе формулы (7), являются формулы, соответствующие отношениям вычислимости C_1 и C_2 , т.е. формулы (1) и (3).

В примере C – множество отношений вычислимости, использованных для преобразования $N_l^{(0)}$ в $N_l^{(i+1)}$, а I – множество «целевых» величин из правых частей этих отношений. Множество I позволяет алгоритму не заикливаться на одних и тех же отношениях вычислимости, поскольку на каждом шаге поиска из числа рассматриваемых отношений вычислимости исключаются отношения C_j : $x_1, \dots, x_n \rightarrow u$, в которых $u \in I$.

ПРИМЕР 4.2.

Пусть обучаемый ввел формулу (9).

Шаг 1.

$$N_l^{(0)} = \{(c, c)\}, N_r = \{a_1, b_1\}, I = \{(c, c)\}, C = \emptyset$$

$N_l^{(0)}$ можно преобразовать в $N_l^{(1)}$ тремя способами. Для краткости изложения рассмотрим последние шаги всех трех способов.

СПОСОБ 1.

...

$$\begin{aligned} \text{Шаг 3. } N_l^{(2)} &= \{a_1, b_1, a_2, b_2\}, I = \{(c, c), c_1, c_2, a_1, b_1, a_2, b_2\}, \\ C &= \{C_3, C_8, C_9\} \end{aligned}$$

СПОСОБ 2.

...

Шаг 4.

$$N_l^{(3)} = \{a_1, a_2, b_1, b_2\}, I = \{(c, c), c, a, b, a_1, a_2, b_1, b_2\},$$

$$C = \{C_{10}, C_4, C_6, C_7\}$$

СПОСОБ 3.

...

Шаг 4.

$$N_l^{(3)} = \{a_1, b_1, a_2, b_2\}, I = \{(c, c), c, c_1, c_2, a_1, b_1, a_2, b_2\},$$

$$C = \{C_{10}, C_5, C_8, C_9\}$$

Согласно результатам работы алгоритма, обучаемый при вводе формулы (9) мог использовать следующие формулы:

- формулы (4-6);
- формулы

$$a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}, c = a - b$$

и утверждение «по c можно вычислить скалярное произведение (c, c) »;

- формулу $c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$,

формулы (5-6) и утверждение «по c можно вычислить скалярное произведение (c, c) ».

Отметим, что идея и реализация предлагаемого алгоритма измерения прогресса в решении похожа на некоторые идеи и техники из теории систем переписывания термов [8]. Так, например, отношения вычислимости похожи на *редукции* – бинарные отношения, с помощью которых кодируются правила преобразования термов; в ходе переписывания термов так же, как и в предлагаемом алгоритме, осуществляется преобразование (переписывание) совокупности одних величин (термов) в другие. К сожалению, напрямую аналогию провести нельзя, поскольку отличаются цели и критерии останковки преобразования.

5. Предлагаемый алгоритм измерения прогресса в решении: дополнительные эвристики

Без использования дополнительных эвристик предлагаемый алгоритм, также как и алгоритм, реализованный в Andes Physics Tutor, является уязвимым для имитации правильно введенных шагов решения (данное утверждение легко проверяется с помощью примеров 3.2-3.3). Ниже рассматриваются различные крайние случаи форм шагов решения обучаемых и предлагаются эвристики для их обработки.

СЛУЧАЙ 1.

Пусть в шаге решения обучаемого часть переменных заменена на числовые значения, но

$$N_l^{(0)} \neq \emptyset, N_r \neq \emptyset.$$

Примерами таких шагов являются формулы (9) и (11). Модификация алгоритма заключается в проведении дополнительной проверки:

1. с помощью формул, соответствующих найденной последовательности отношений вычислимости Ci_1, \dots, Ci_k , в формуле, введенной обучаемым, заменить переменную $N_l^{(0)}$ на выражение с переменными из $N_l^{(k-1)}$;
2. в полученном выражении заменить все переменные, кроме переменных из N_r , на соответствующие им числовые значения;
3. с помощью инструмента символьных вычислений вычесть из левой части выражения его правую часть и упростить полученное выражение;
4. если полученное выражение не упростилось до 0, то это означает, что имела место имитация правильно введенного шага.

Поясним предлагаемую эвристику выявления имитации правильно введенного шага с помощью примеров.

ПРИМЕР 5.1.

Пусть обучаемый ввел формулу (9).

Шаг 1. После применения формул, соответствующих отношению вычислимости $C=\{C_3, C_8, C_9\}$, формула, введенная обучаемым, принимает вид:

$$(12) (a_1 - b_1)^2 + (a_2 - b_2)^2 - (a_1 - b_1)^2 = 4.$$

Шаг 2. Т.к. $N_r=\{a_1, b_1\}$, то в формуле (12) заменяются на числовые значения переменные a_2, b_2 . Получается выражение

$$(a_1 - b_1)^2 + (2 - 4)^2 - (a_1 - b_1)^2 = 4,$$

которое упрощается до 0, т.е. обучаемый не пытался имитировать правильно введенный шаг.

ПРИМЕР 5.2.

Пусть обучаемый ввел формулу (11).

Шаг 1. Посредством применения формул, соответствующих отношениям вычислимости $C=\{C_3, C_8, C_9\}$, формула, введенная обучаемым, принимает вид:

$$(13) c_1^2 + c_2^2 = c_1 + c_2 + 12.$$

Шаг 2. Т.к. $N_r=\{c_1, c_2\}$, то в формуле (13) отсутствуют переменные, которые могут быть заменены на числовые значения. Получившееся выражение $c_1^2 + c_2^2 - c_1 - c_2 - 12$ не упрощается до 0, т.е. имела место попытка имитации правильно введенного шага.

Что касается известных проблем алгоритмической неразрешимости массовой задачи эквивалентности термов (равенства слов), то еще в 1970-х гг. (В.М. Глушков и др.) отмечалась полезность понятия практической разрешимости, учитывающей эвристики и вычислительную сложность задач, в сравнении с зачастую практической неразрешимостью ряда теоретически разрешимых задач. Кроме того, практический опыт разработчиков следящих ИОС, упомянутых во введении данной статьи, а также опыт автора, свидетельствует о том, что в подавляющем

большинстве случаев проверок решений не слишком сложных учебных задач упрощение разности выражений с использованием достаточно развитых инструментов символьных вычислений будет успешно осуществлено.

СЛУЧАЙ 2.

Пусть

$$N_l^{(0)} \neq \emptyset, N_r = \emptyset,$$

т.е. шаг решения обучаемого можно привести к виду <переменная>=<числовое выражение>. Модификация алгоритма измерения прогресса в решении заключается в проведении следующей дополнительной проверки:

1. в формуле, введенной обучаемым, $N_l^{(0)}$ заменить на пустую строку;
2. составить выражения e_i , являющиеся результатом алгебраических комбинаций формул, соответствующих преобразованию переменной из $N_l^{(0)}$ в числовое выражение;
3. для каждого выражения e_i с помощью некоторой метрики (например, расстояния Левенштейна) установить степень сходства с формулой из предыдущего пункта;
4. если найдется единственное выражение e_j , обладающее наибольшей степенью сходства

$s(e_j) \geq \alpha$, где α – некоторый заранее заданный порог,

то можно предположить, что числовое выражение e_j соответствует формуле, введенной обучаемым, а формула, введенная обучаемым, является результатом алгебраических комбинаций формул, с помощью которых было получено выражение e_j .

Пусть обучаемый ввел формулу

$$(c, c) - (1 - 2)^2 = (3 - 4)^2.$$

Выражение e_1 имеет вид $(c, c) = (1 - 3)^2 + (2 - 4)^2$. Пусть порог $\alpha = 5$. Тогда значение расстояния Левенштейна $s(e_1)=4$ говорит о высоком сходстве выражения e_1 и формулы, введенной обучаемым, т.е. обучаемый не пытался имитировать правильно введенный шаг.

СЛУЧАЙ 3.

В шаге решения обучаемого нет переменных. Примерами соответствующих данному случаю формул являются формулы

(14) $5 = 5$,

(15) $(-2)^2 + (-2)^2 = 8$.

Очевидно, что нецелесообразно обрабатывать формулы вида (14). Формула (15) в контексте рассматриваемого примера может соответствовать вычислительному действию, осуществляемому с целью нахождения (c, c) . Как и в предыдущем случае, путем генерации и последующего нечеткого сравнения строковых выражений можно сделать предположение о том, какую именно величину обучаемый пытался найти. Предположение можно проверить, например, показав обучаемому диалоговое окно с уточняющим вопросом типа «какую из этих величин вы пытались найти?». Для того, чтобы при этом косвенным образом не предоставить обучаемому подсказку по плану решения задачи, список переменных в вопросе можно дополнить двумя случайно выбранными величинами из списка «целевых» величин задачи.

СЛУЧАЙ 4.

В шаге решения обучаемого есть переменные, но не получается выделить единственную «целевую» величину. Примером такого шага является формула $c_1^2 + c_2^2 = 8$. В этой формуле содержатся две «целевые» величины с одинаковым и небольшим приоритетом. По сути, эта формула соответствует вычислительным действиям, осуществляемым с целью нахождения (c, c) . В подобных случаях можно среди отношений вычислимости поискать такое отношение $C_j: X \rightarrow y$, у которого X совпадает с переменными из шага решения обучаемого. Если такое отношение найдется, то можно предположить, что обучаемый хотел найти «целевую» величину y и уточнить это предположение посредством диалогового окна.

6. Заключение

Разумеется, необходимо проанализировать на возможность автоматизированной проверки не только другие разделы линейной алгебры, но и другие естественно-научные дисциплины. Но интересен и промежуточный итог. В данной статье на примере задачи из курса линейной алгебры рассмотрен ряд случаев, возникающих при проверке решений одного класса задач. Предложена совокупность алгоритмов и эвристик для важной для управления процессом обучения задачи – измерения прогресса в решении.

Как показывает ряд приведенных выше примеров, автоматизированная проверка решений, введенных в достаточно свободной форме, возможна, но имеет свои ограничения. Чем меньше переменных и больше числовых значений в формуле, введенной обучаемым, тем большая неопределенность возникает при ее анализе. Предложенные эвристики помогают в ряде случаев, но не во всех, поэтому при проектировании следящей ИОС желательно предусматривать возможность для обучаемого временно засчитать нераспознанный системой шаг решения и вывести его в специальном отчете для преподавателя.

Дальнейшее развитие исследований предполагает уточнение области применимости предложенной совокупности алгоритмов и эвристик, а также выделение из всего множества задач по предметам естественно-научного цикла круга задач, для которых возможна автоматизированная проверка решений. К наиболее близким исследованиям по этому вопросу можно отнести исследования по разработке автоматических решателей [2].

Литература

1. ЛЕВИНСКАЯ М. А. *Продукционная модель интерактивной компоненты обучающей системы* // Сб. науч. тр. «Математика. Компьютер. Образование». – Ижевск, Научно-издательский центр «Регулярная и хаотическая динамика», 2003. – Т. 1. – С. 81–92.

2. ПОДКОЛЗИН А. С. *Компьютерное моделирование процессов решения математических задач*: – М.: Изд-во ЦПИ при мех.-мат. ф-те МГУ, 2001. – Т.1-2. – 235 с.
3. ФРИДМАН Л. М. *Логико-педагогический анализ школьных учебных задач*: – М.: Педагогика, 1977. – 208 с.
4. MELIS E. *ActiveMath: an Intelligent Tutoring System for Mathematics* / E. Melis, J. Siekmann // *Artificial Intelligence and Soft Computing – ISAISC 2004*. Springer Berlin Heidelberg, 2004. – Vol. 3070. – P. 91–101.
5. SANGWIN C. J. *Assessing Elementary Algebra with STACK* // *International Journal of Mathematical Education in Science and Technology*. – 2008. – Vol. 38, №. 8. – P. 987-1002.
6. SANGWIN C. J. Automating the marking of core calculus and algebra: eight years on. URL: <http://web.mat.bham.ac.uk/C.J.Sangwin/Publications/2009-12-mmng.pdf> (Дата обращения: 12.03.2013)
7. SHAPIRO J. A. *An Algebra Subsystem for Diagnosing Students' Input in a Physics Tutoring System* // *International Journal of Artificial Intelligence in Education*. – IOS Press, 2005. – № 15. – P. 205-228.
8. TERESE *Term Rewriting Systems* // *Cambridge Tracts in Theoretical Computer Science*. – Cambridge University Press, 2003. – Vol 55. – 908 P.
9. VANLEHN K. *The Andes Physics Tutoring System: Lessons Learned* / // Восточный фронт Армии Генерала А. В. Колчака: сайт. – URL: <http://east-front.narod.ru/memo/latchford.htm> (дата обращения 23.08.2007).
10. <http://docs.moodle.org/24/en/Mathematics> (Дата обращения: 16.03.2013).

ARTICLE TITLE, ПЕРЕВОД НАЗВАНИЯ НА АНГЛИЙСКИЙ

Natalia Smirnova, Institute of Control Sciences of RAS, Moscow, researcher (smirnovanatalia2008@gmail.com).

Abstract: The paper is dedicated to computer-aided assessment of multi-step problems provided by interactive model-tracing intelligent tutoring systems. A new approach is suggested for determining which formulas could student's formula be derived from. The suggested approach is compared to the known algorithm.

Keywords: model-tracing intelligent tutoring systems, tutoring systems, computer-aided assessment

Статья представлена к публикации
членом редакционной коллегии ...заполняется редактором...

Поступила в редакцию ...заполняется редактором...
Опубликована ...заполняется редактором...