

УДК 004

DOI: 10.46548/21vek-2020-0952-0003

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ЦЕПОЧЕК ПРЕОБРАЗОВАНИЙ НАД СИМВОЛЬНЫМИ ФАКТАМИ НА ОСНОВЕ МЕТОДА ПЕРЕБОРА ЛОГИЧЕСКИХ ПРАВИЛ

©2020

Кацман Виктор Игоревич, аспирант,

Новиков Федор Александрович, доктор технических наук, профессор,
Санкт-Петербургский Политехнический Университет Петра Великого
(195251 Санкт-Петербург, ул. Политехническая, д. 29,
e-mails: vikto9494@gmail.com, fedornovikov51@gmail.com)

Аннотация. Одним из наиболее важных элементов современного образовательного процесса в области технических наук является решение задач на преобразование фактов над символическими выражениями. В ходе решения таких задач обучаемым требуется вывести новое для них решение задачи опираясь на изученные ими правила. Закрепление навыков требует решения больших объемов задач, что влечет необходимость проверки полученных обучаемыми решений. Уже реализовано множество платформ, позволяющих автоматизировать проверку решений. Как правило, они проверяют исключительно ответ, но не ход решения задачи. Такой подход к проверке оставляет возможность получения правильного ответа и засчитанного задания при не до конца продуманном или неправильном решении. Мы предлагаем способ автоматической проверки решения задач, основанный на последовательной проверке совершенных преобразований путем эффективного перебора правил, по которым эти преобразования могли быть совершены. Система на каждом шаге решения перебирает допустимые преобразования, пытаясь найти использованное обучаемым. Проблемы неоднозначности формы записи решаются комбинацией методов тестирования, унификации и структурирования правил записи решения. Предложенный способ автоматической проверки был опробован на студентах младших курсов Политехнического и Электротехнического Университетов и показал неплохие результаты: производительность преподавательского труда при проверке задач повысилась более, чем в 4 раза.

Ключевые слова: образование, игрофикация, преобразование фактов над символическими выражениями, автоматическая проверка решений, правила преобразований

AUTOMATIC VERIFICATION OF TRANSFORMATION CHAINS OF SYMBOLIC FACTS BASED ON LOGICAL RULES SEARCH

©2020

Katsman Viktor Igorevich, postgraduate student,

Novikov Fedor Alexandrovich, PhD in Engineering, professor,
Peter the Great St. Petersburg Polytechnic University
(195251 St. Petersburg, Polytechnicheskaya st., 29,
e-mails: vikto9494@gmail.com, fedornovikov51@gmail.com)

Abstract. One of the most important elements of the modern educational process in the field of natural sciences is the solution of problems of transforming facts of symbolic expressions. When solving such problems, learners need to deduce a new solution of the problem based on the rules they have already learned. To consolidate these skills, learner need to solve a large volume of tasks, which entails the need of verification of the solutions. Many platforms have already been implemented that allow to automate the verification of solutions. Generally, these systems only check the answer, but not the progress of the problem. This approach of verification leaves the possibility of getting the correct answer and the task counted if the decision is not fully thought out or incorrect. We propose a method for automatic verification of problem solutions based on sequential verification of completed transformations by efficiently iterating through the rules that allowed these transformations to be performed. At each step of the solution, the system iterates through the allowed transformations, trying to find one, that learner used. Problems of ambiguity of the recording form are solved by a combination of testing method, unification, and structuring of the recording rules. The proposed method of automatic verification was tested on students of junior courses of the Polytechnic and Electrical engineering Universities and showed good results: the productivity of teaching staff when checking tasks increased more than 4 times.

Keywords: education, gamification, facts of symbolic expression transformations, automated solution verification, transformation rules

Введение. На текущий момент автоматизация труда – это один из самых эффективных, а поэтому и распространенных способов повышения производительности в самых разных отраслях, в том числе в образовании. Стремительно развиваются платформы автоматизации обучения, которые позволяют преподавателю переложить часть рутинной работы на них,

а самому сконцентрироваться на более интересном и важном. Для автоматизации рутинной работы достаточно разово, но тщательно продумать, что и как именно нужно сделать.

Автоматизация актуальна для разных составляющих преподавательского труда, от объяснения нового материала до проверки решений учебных задач. Реше-

ния для автоматизации многих из этих составляющих уже существуют и активно используются. Например, в современных платформах онлайн обучения есть возможность подменить рассказ преподавателя видеозаписью, что позволяет масштабировать деятельность преподавателя: показывать разово записанных фрагмент занятия много раз.

Существенно сложнее качественно автоматизировать деятельность преподавателя в ситуациях, когда требуется индивидуальный подход к обучающимся, понимания уровня их познаний, настроения, психологии. Так, следом за изложением нового материала, необходимо добиться, чтобы обучаемые глубоко усвоили этот материал. А для этого им требуется решить большое число задач, что влечет за собой необходимость проверки решений большого числа задач. Желательно, чтобы по получаемым результатам проверки обучаемый мог понять, где конкретно он мог допустить ошибку, а, может, и получить наводку, как эту ошибку лучше исправлять.

Целью этой статьи является предложение метода, позволяющего автоматически проверять решения задач на преобразования фактов – логических комбинаций над символическими выражениями. А именно, такими фактами являются равенства, неравенства над алгебраическими и логическими выражениями, а также системы и совокупности других фактов. Задача на преобразование фактов подразумевает преобразование исходно данного факта так, чтобы он удовлетворял целевым критериям. Пример задачи на преобразование фактов над символическими выражениями: решить системы уравнений.

$$\begin{cases} x + 5 = y \\ x = 1 - y \end{cases}$$

Здесь приведенная выше система является исходно данным фактом. Целевые критерии: выведенный обучающимся в качестве решения задачи факт должен быть системой равенств, в левых частях которых содержатся переменные x и y , а в правых частях числа, которым они равны.

Таким образом, предлагаемый метод позволяет автоматически проверять учебные задачи на оперирование формулами и правилами преобразований в разных областях точных наук. Составляющие метода были неоднократно проверены нами на студентах младших курсов Политехнического и Электротехнического университетов.

Материалы и результаты исследования. Наиболее используемым на сегодняшний день способом автоматической проверки решений задач является проверка по ответу. Вручную преподаватель задает условие задачи и критерии правильного ответа, а затем по этим критериям автоматически проверяется решение обучаемого. Соответственно, решение засчитывается обучаемому если его ответ удовлетворяет критериям ответа, заданным преподавателем. Этот подход позволяет проверить ответы очень многих учебных задач на вывод, он активно используется в системах

дистанционного обучения, например в *Coursera* [4], *EdX* [5], *Stepic* [6], *WebWork* [7], *Moodle* [8] и других.

В задачах, где ответом является символическое выражение, проверка по ответу может работать двумя способами. Первый – алгоритмически точная проверка равенства выражений. А именно, алгоритм Тарского позволяет верифицировать истинность замкнутых арифметических формул первого порядка с переменными для вещественных чисел, то есть с конечным числом вещественных чисел [9].

Но для большинства символических выражений способов точной проверки по ответу не существует. Общая невозможность автоматической проверки равенства символических выражений, использующих рациональные числа и операции возведения в степень следует из теоремы Ричардсона [10]; невозможность автоматической проверки эквивалентности работы двух программ по их коду следует из теоремы Райса [11];

Существенно больше возможностей предоставляют вероятностные методы, а именно проверка путем совершения серии вычислительных экспериментов или тестированием. В каждом из вычислительных экспериментов результат, полученный обучаемым при определенных значениях символических переменных, сравнивается с правильным ответом. Если во всех вычислительных экспериментах ответ обучаемого совпал с правильным, решение считается правильным, в противном случае можно найти и предъявить обучаемому контрпример.

Здесь становится ключевым подбор значений параметров, для которых требуется провести вычислительные эксперименты, поскольку неправильный ответ может быть принят за правильный, если покрытие тестами не полное. Преподавателю требуется либо провести его вручную – продумать проверяющие тесты, либо алгоритмизировать этот подбор – продумать автоматическую генерацию тестов. Существенную проблему возникают при проверке методом тестирования выражений, имеющих небольшую область определения, поскольку в тестах требуется подбирать значения переменных, при которых значения сравниваемых выражений будут определены. Помогает переход к вычислению значений выражений в комплексных числах [12-13].

И вероятностный, и точный методы проверки решений по ответу довольно распространены и широко используются в системах дистанционного обучения. Например, при создании онлайн-курсов на платформе *Stepic* можно использовать все перечисленные методы проверки по ответу [14].

Но все существующие методы проверки решений по ответу позволяют проверить только конечный результат решения без проверки того, каким именно способом обучаемый к этому результату пришёл. А зачастую случается, что обучаемые приходят к правильному ответу случайно, в частности забывая про ряд важных особенностей или не практикуя необходимые навыки. Проверка не только ответа, но и самого ре-

шения может помочь обучаемому найти ту точку, начиная с которой его рассуждения стали ошибочными.

Предлагаемый в статье подход существенно расширяет пространство возможностей автоматической проверки решений задач в целом, а не только по ответу, что позволяет существенно повысить качество изучения почти всех областей точных наук.

Модель процесса решения задачи и его записи

Первым шагом при решении типовой учебной задачи обучаемый читает и интерпретирует условие задачи. Затем обучаемый начинает шаг за шагом выводить следствия из понятого им условия. Это происходит до тех пор, пока очередное выведенное следствие не приводит обучаемого к решению задачи. Пространство следствий, которые могут быть выведены из заданного условия весьма велико и для того, чтобы прийти к решению обучаемому нужно двигаться в правильном направлении.

В качестве решения задачи обучаемый предъявляет цепочку выведенных им следствий. Преподаватель проверяет, что каждое выведенное следствие действительно следует из предыдущих и, если все так, зачитывает решение задачи. А в случае ошибки указывает на недостаточно обоснованное следствие.

При решении задач на преобразование фактов над символьными выражениями, следствия могут выводиться из исходно данного факта, входящих в него выражений или допустимых правил преобразований. В зависимости от требований к целевому факту, к следствиям из исходно данного факта может применяться либо нет требование их эквивалентности – а именно, чтобы обратное преобразование также было верным. И в том, и в другом случае правильность решения определяется удовлетворением итогового результата целевым критериям и корректностью вывода следствий. Они считаются корректными, если выведены по допустимым правилам вывода и некорректными в противном случае.

Правила вывода различаются по следующим признакам:

1. Эквивалентность – возможность применения правила в обе стороны. Например, правило, которое меняет местами два уравнения в системе – эквивалентно, а правило, которое оставляет только одно из этих уравнений – нет.

2. Актуальность в контексте любых обозначений – правильность правила при подстановке вместо используемых в нем переменных и функций других выражений. Например, правило « $a + b = b + a$ » верно при любых значениях a и b и, соответственно, актуально в любом контексте. Правило « $I * R = U$ » верно только когда I соответствует силе тока, R – сопротивлению, а U – напряжению.

3. Для преобразования выражений / для преобразований фактов над выражениями. Например, « $a + b = b + a$ » — это правило преобразования выражений, а возможность прибавить слагаемое к обоим частям неравенства или вычесть одно уравнение из другого – это правила преобразования фактов.

4. Исходно известные / выводимые в процессе решения. Обучаемый может вывести недостающие ему правила в процессе решения задачи и затем их несколько раз переиспользовать – например, для сокращения объемов записи.

5. Требуемыми для применимости внешними ограничениям. Например, требованиями соблюдения условий на область принимаемых значений переменных.

6. Для правил преобразования выражений – порядковым контекстом применимости. Правило может быть верным при применении в цепочке равенств или в цепочке убываний, или в цепочке возрастаний.

7. Применяемые к основной цепочке преобразований фактов / к второстепенным, используемым для выведения вспомогательных правил

8. Сложностью для применения обучаемым. Применение слишком большого числа правил за один шаг вывода делает такой шаг нетривиальным умозаключением, которое при написании решения требуется разбить на несколько шагов. Отсутствие ограничений на число применяемых правил за один переход делает, например, корректными решения задач на доказательство тождеств, состоящие исключительно из сопоставления фактов, данных в условии.

Набор допустимых правил зависит как от предметной области, так и от конкретной задачи. Например, если целевой факт должен быть эквивалентен исходному, то все правила преобразований, применяемые обучаемым в основной цепочке, должны сохранять эквивалентность. Под основной цепочкой преобразований здесь подразумевается цепочка, начинающаяся с исходно данного факта и заканчивающаяся целевым.

При записи решения основной акцент делается на записи последовательности фактов, выведенных в основной цепочке преобразований. Примененные в ходе вывода правила преобразования обычно опускаются, за исключением случаев, когда определение примененного правила явно неочевидно. Вывод вспомогательных правил зачастую бессистемно выносятся на поля или в другие места, не привязанные напрямую к месту их применения.

Метод, подходящий для автоматической проверки, должен учитывать описанные выше нюансы, а нотация решений, записываемых в цифровом виде, должна быть близка к общепринятой нотации при записи решений на бумаге, в противном случае переход на нее будет затруднителен для обучающихся.

Предлагаемый метод автоматической проверки. В основе предлагаемого метода лежит метод эффективного перебора правил, который описывался нами в [12,13,15] на примере задачи автоматической проверки корректности цепочек преобразований символьных выражений. Суть метода в отдельной проверке каждого совершенного обучаемым перехода путем попытки найти примененную им комбинацию правил преобразования, которая покажет обоснованность данного перехода. Поиск осуществляется путем эффективного перебора, на первом этапе которого

отбираются правила, применимые к левой и правым частям совершенного преобразования. На втором этапе правила применяются к левой и правым частям, а затем результаты применения правил сравниваются между собой методом тестирования, то есть путем совершения серии вычислительных экспериментов. Глубина рассматриваемого перебора правил подбирается в зависимости от сложности правил. Если комбинации правил, трансформирующих правые и левые части совершенного обучаемым преобразования в равные выражения, находятся, то такой переход считается корректным. В противном случае – не корректным или слишком сложным для обучаемого.

Если все совершенные обучаемым переходы оказываются корректными, а итоговый результат-ответ удовлетворяет целевым требованиям, то решение объявляется корректным, а задача засчитанной. В противном случае обучаемому может быть предъявлен неправильный или слишком сложный переход. В этом случае обучаемый может попробовать разбить его на несколько более простых переходов.

Для применимости метода к автоматической проверке преобразований фактов над символьными выражениями, эффективный перебор правил должен удовлетворять требованиям описанной в предыдущей секции модели.

Так, при требовании сохранения эквивалентности перебор может осуществляться исключительно среди эквивалентных правил преобразования. При отсутствии такого требования односторонние правила могут быть применены только одним способом. Актуальность в контексте любых обозначений учитывается при поиске применимых правил. Правила, выведенные обучаемым самостоятельно, актуальны в любом контексте, если и только если они выводились исключительно на основе правил, актуальных в любом контексте.

Все доступные при решении задачи правила вывода делятся на 3 группы: системные правила; правила, настраиваемые преподавателем и правила, выводимые обучающимся. К первой группе относятся правила, связанные с особенностями оперирования фактов внутри систем и совокупностей. В эту часть входит множество ситуаций с трудоемким перебором, например, факты могут быть переставлены в произвольном порядке $n!$ способами. При этом, такие перестановки фактов являются для обучаемых тривиальными переходами, которые могут быть совмещены с другими преобразованиями, например, с группировкой некоторых из них и сложением (рис. 1). При проверке применимости системных правил, совершается предварительная оценка сложности перебора, далее в нетрудоемких случаях происходит полный перебор. В более трудоемких ситуациях перебор сокращается вплоть до рассмотрения исключительно одного варианта, в котором все внутренние факты сортируются лексикографически. Например, в случае перестановки фактов, это означает что в случае, когда факты были только переставлены, но никакие другие преобразова-

ния при этом не совершались – переход будет признан верным. При дополнительном совершении других правил преобразования – не обязательно.

$$\begin{aligned}
 [V(x,2) = [V(n,k) = C(k+n-1,k) = \frac{(k+n-1)!}{(n-1)! \times k!}] &= \frac{(x+2-1)!}{(x-1)! \times 2!} = \frac{(x+1)!}{(x-1)! \times 2!}] \\
 V(x,2) &= A(x,1) \\
 \frac{(x+1)!}{(x-1)! \times 2!} &= A(x,1) \\
 \frac{(x+1)!}{(x-1)! \times 2!} &= \frac{x!}{(x-1)!} \\
 \begin{cases} \frac{2(x-1)!}{x!} \neq 0 \\ \frac{(x+1)!}{(x-1)! \times 2!} \times \frac{2(x-1)!}{x!} = \frac{x!}{(x-1)!} \times \frac{2(x-1)!}{x!} \Rightarrow \\ \frac{(x+1)!}{x!} = 2 \Rightarrow x+1 = 2 \Rightarrow x+1-1 = 2-1 \Rightarrow x = 1 \end{cases} \\
 x &= 1
 \end{aligned}$$

Рисунок 1 - Пример записи корректного решения задачи на комбинаторные формулы

Правила, настраиваемые преподавателем и выводимые обучающимся, имеют один и тот же вид. Разница в том, что правила, настроенные преподавателем, считаются верными по-умолчанию, а правила, добавляемые обучающимся, должны быть из них выведены. Также в преподавательских правилах сложность настраивается на усмотрение преподавателя. Правила, выведенные обучаемым за один шаг, могут быть применены только с другими правилами, помеченными преподавателем как тривиальные, обычно это базовые арифметические правила, например, приведение подобных.

Эксперименты и результаты. Эксперименты были проведены с целью удостовериться, что разработанный метод применим не только для преподавателей, но и для обучаемых. А именно, проверить, что обучаемые могут быстро усвоить используемую для записи решений нотацию, что запись решения в цифровом виде требует примерно столько же времени, сколько требует запись решения на бумаге, а также, что результаты, полученные при автоматической проверке решений будут близки к результатам ручной проверки.

На текущий момент мы уже успели провести эксперименты на 2х потоках студентов 1 и 2го курса Политехнического и Электротехнического Университетов в рамках курсов по дискретной математике в 2017-2020 годах. За прошедшие годы общая концепция экспериментов не поменялась, но разработанная система автопроверки была существенно улучшена: повысилось количество поддерживаемых типов задач, поправлены баги и усовершенствована логика проверки. В ходе экспериментов студентам кратко объяснялась логика работы автоматической проверки и правила записи решений, после чего студентам предлагался ряд задач по комбинаторике, теории множеств и логическим исчислениям.

Результаты экспериментов показали, что студенты в состоянии быстро освоить новые правила записи решений: из приступавших к решению задач нашлось только двое, которые не смогли решить ни одной за-

дачи. Скорость решения задач в системе получилась в среднем примерно в полтора раза медленнее скорости записи решений на бумаге.

Отдельной целью экспериментов было понимание о том, сколько времени экономит преподаватель благодаря автоматической проверке. Довольно трудоемкой работой оказалось составление допустимых правил преобразований (на это уходят часы), но это разовое действие. Сведение результатов автоматической проверки воедино для одного эксперимента занимает в среднем полчаса-час, в зависимости от количества студентов. Ручная проверка решений тех же задач для другого потока студентов занимала в среднем более 6-ти часов.

Следовательно, проведенные эксперименты подтверждают актуальность и возможность использования разработанного метода для автоматической проверки решений учебных задач на преобразования фактов над символьными выражениями.

Заключение. Предложенный метод автоматической проверки решений учебных задач на преобразование фактов над символьными выражениями сильно расширяет возможности автоматической проверки учебных задач в областях точных наук. Это позволяет повысить навыки учащихся по владению формулами, и, таким образом, способствует повышению качества образования в целом. В текущий период вынужденной самоизоляции предлагаемые способы становятся особенно актуальны.

Мы продолжаем работать в этом направлении, наши дальнейшие планы связаны с увеличением числа поддерживаемых предметных областей, а также с улучшением записи решений задач.

СПИСОК ЛИТЕРАТУРЫ:

1. Auvinen T. Harmful Study Habits in Online Learning Environments with Automatic Assessment. Proceedings of the 2015 International Conference on Learning and Teaching in Computing and Engineering. – 2015 – С. 50-57.
2. Willman S., Linden R., Kaila E., Rajala T. and Laakso M. On study habits on an introductory course on programming. Computer Science Education.- 2015. – С. 276-291.
3. Tirronen M. and Tirronen V. A framework for evaluating student interaction with automatically assessed exercises. Koli Calling '16 Proceedings of the 16th Koli Calling International Conference on Computing Education Research. - 2016. – С. 180-181.
4. Coursera. [Электронный ресурс] – Режим доступа: <https://ru.coursera.org/>
5. EDX. [Электронный ресурс] – Режим доступа: <https://www.edx.org/>. Последний доступ 05.08.2020.
6. Stepik. [Электронный ресурс] – Режим доступа: <https://welcome.stepik.org/en>.
7. WeBWorK. [Электронный ресурс] – Режим доступа: <https://webwork.elearning.ubc.ca/webwork2/>.
8. Moodle. [Электронный ресурс] – Режим доступа: <https://moodle.org/>.
9. Rogers H. The Theory of Recursive Functions and Effective Computability. MIT Press, 1987, ISBN: 0-262-68052-1;

ISBN: 0-07-053522-1.

10. Richardson D 1968, Some Unsolvable Problems Involving Elementary Functions of a Real Variable (Journal of Symbolic Logic 33 (4)) pp 514-520 10.2307/2271358.
11. Rice H 1953 Classes of Recursively Enumerable Sets and Their Decision Problems (Transactions of the American Mathematical Society 74 (2)) p 358 10.2307/1990888.
12. Novikov, F., Katsman, V., Mosin, V., 2020. Automated verification of expression transformation chains based on computational experiments. IOP Conference Series: Materials Science and Engineering. Scopus, 012132.
13. Новиков Ф. А., Кацман В. И. Автоматическая проверка решений учебных задач на основе комбинации методов перебора логических правил и тестирования. Цифровые технологии в инженерном образовании: новые тренды и опыт внедрения, Москва: Издательство МГТУ им. Н. Э. Баумана, - 2020 – С. 266-273.
14. Stepik Math Lesson. [Электронный ресурс] – Режим доступа: <https://stepik.org/lesson/9176/step/1?unit=1721>.
15. Novikov F. and Katsman V. Gamification of Problem Solving Process based on Logical Rules. Springer LNSC. – 2018 - С. 369-380.
16. А. Ахо, Дж. Хопкрофт, Дж. Ульман «Построение и Анализ Вычислительных Алгоритмов», М., Мир, 1979.
17. М. Гэри, Д. Джонсон, «Вычислительные Машины и Труднорешаемые Задачи», М., Мир, 1982.
18. Н. Нильсон, «Принципы искусственного интеллекта», М., Радио и связь, 1985.
19. О. П. Кузнецов, Г.М. Адельсон-Вельский, «Дискретная математика для инженера», М., Энергоатомиздат, 1988.
20. Ф. А. Новиков, «Дискретная математика для программистов», Питер, 2003, ISBN 5-947233-55-X.

Статья поступила в редакцию 11.08.2020

Статья принята к публикации 14.09.2020