

Модели и методы автоматической проверки решений задач

Докладчик: аспирант Кацман В. И.
Руководитель: д.т.н. Новиков Ф. А.

Автоматическая проверка решений

- ❑ Практическая деятельность - существенная часть обучения
- ❑ При решении типовых практических задач редко требуется изобретение “принципиально новой” модели
- ❑ Компетентность обучаемых во многом определяется степенью владения наиболее ходовым математическим аппаратом



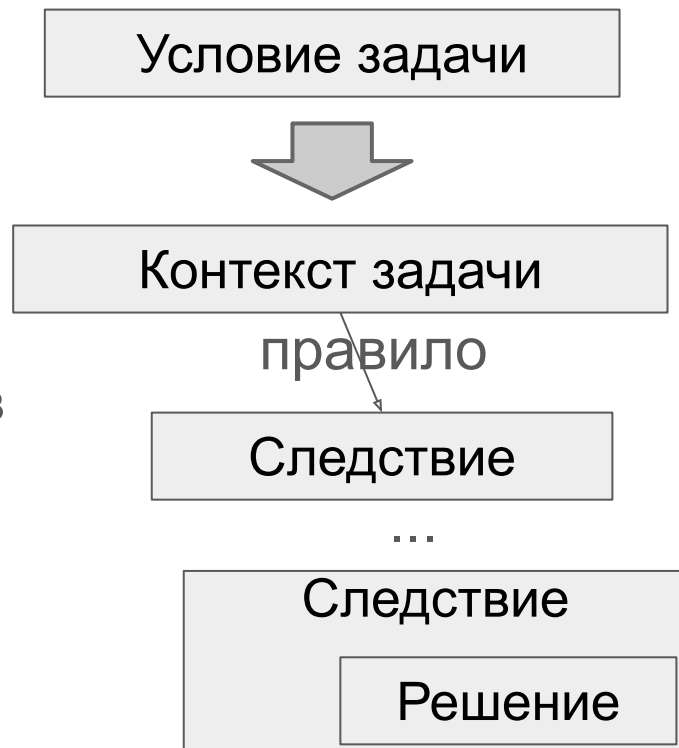
Стимулирование к решению типовых задач и автоматическая проверка их решений актуальны

Процесс решения задачи

1. Чтение и интерпретация условия

2. Пока решение не найдено:

Интерпретация известной информации в виде правил, по которым можно дополнять контекст задачи следствиями



Проверка решения задачи

- ❑ Корректное следствие - следствие, соответствующее возможностям существующей теории
- ❑ Тривиальное для обучаемого следствие - следствие, которое может быть выведено обучаемым за один шаг



Для проверки решения требуется проверить:

1. Корректность и тривиальность всех следствий
2. Наличие требуемого факта среди следствий

Задачи на преобразование формул

❑ Выражение

$$C(n,k) + n! - k^2$$

❑ Факт - логическое условие над выражениями

$$C(n,k) + n! - k^2 \leq k! + 2n!$$

$$\begin{cases} C(n,k) + n! - k^2 \leq k! + 2n! \\ k^2 = n^2 + x \\ x \geq k + n \end{cases}$$

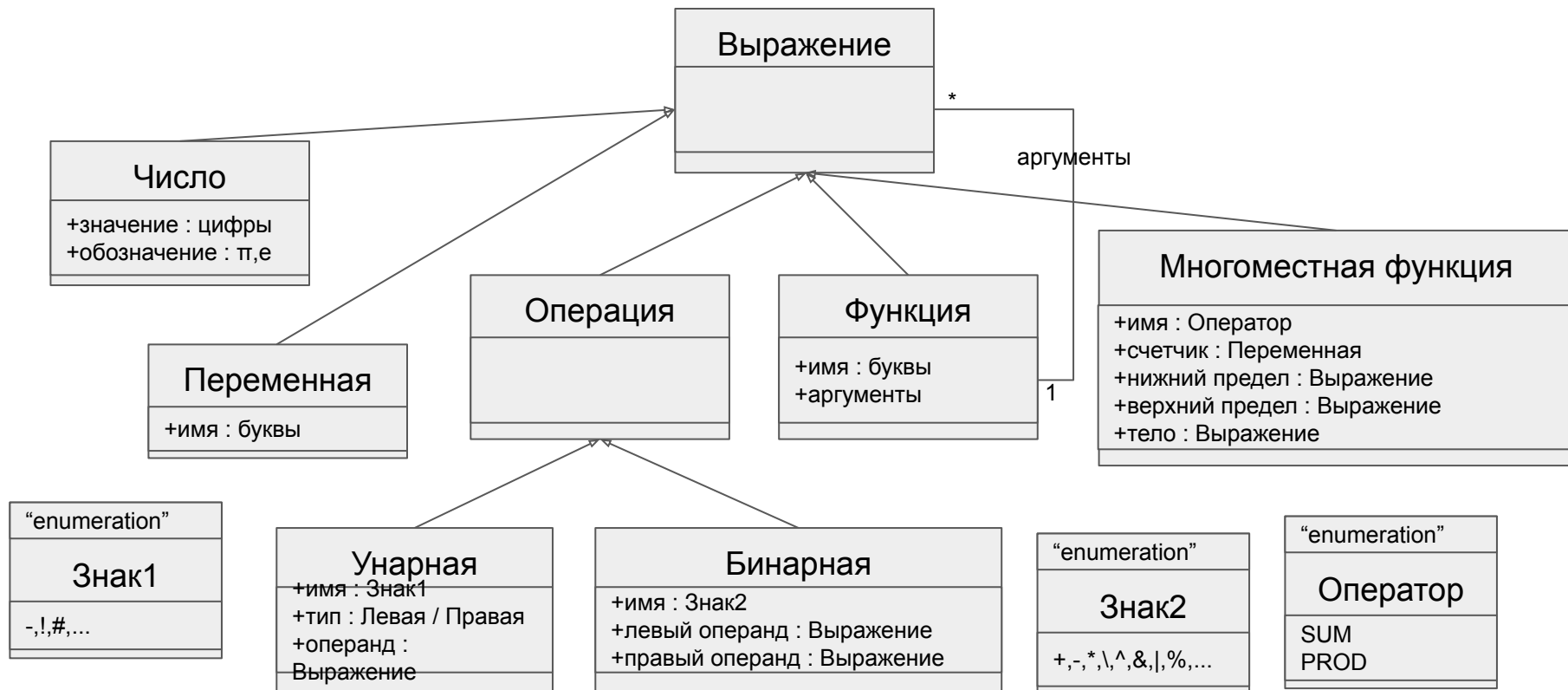
$$\begin{cases} C(n,k) + n! - k^2 \leq k! + 2n! \\ k^2 = n^2 + x \\ x \geq k + n \end{cases}$$

❑ Требуется вывести цепочку корректных и тривиальных преобразований

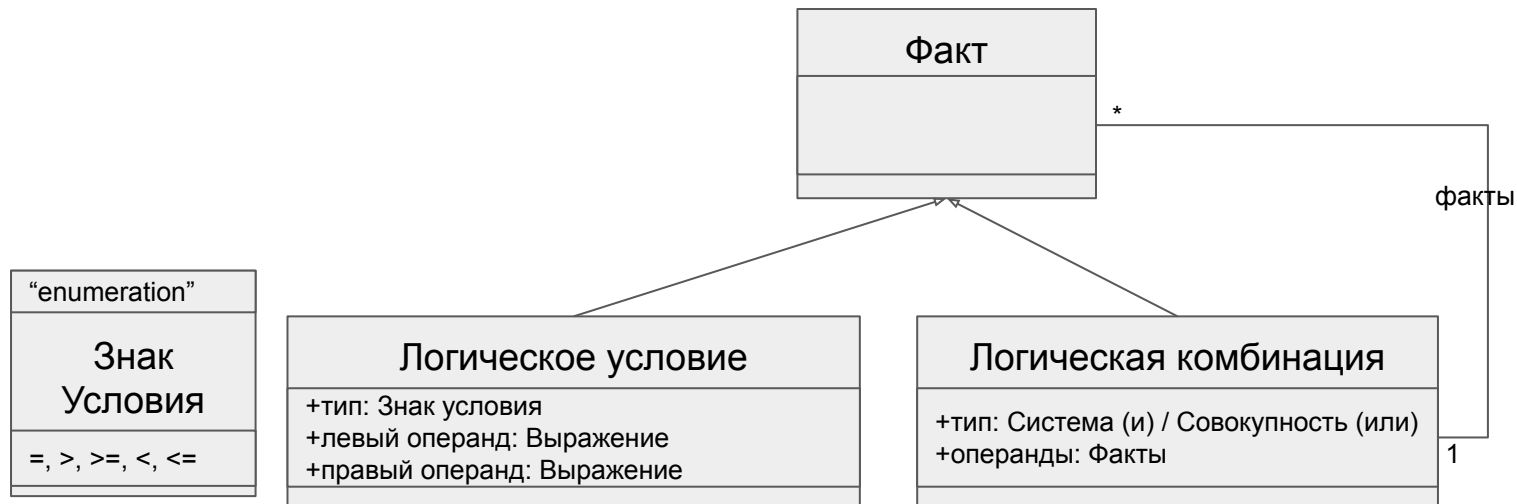
❑ Подтверждающую исходно данный факт

❑ Преобразующее исходно данное выражение или факт к определенному виду

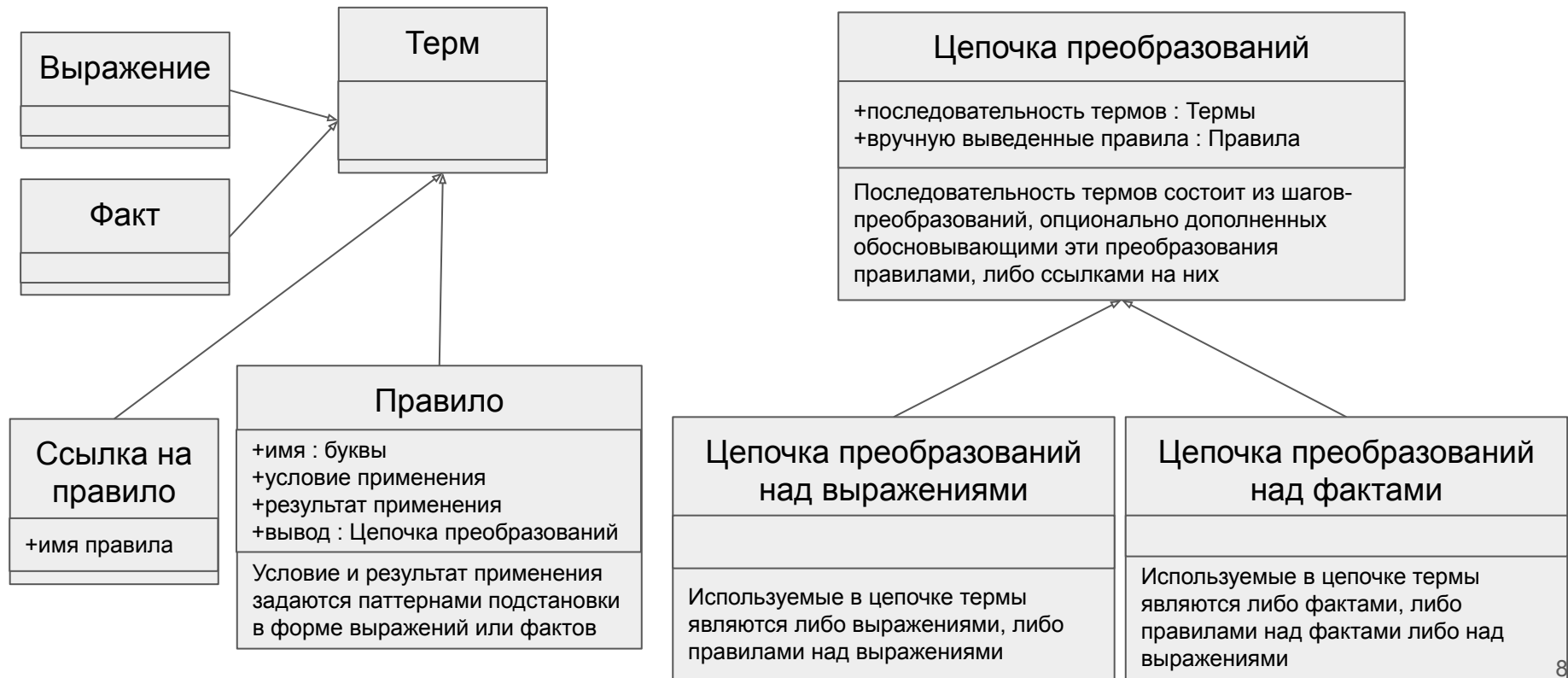
Метамодель выражения



Метамодель фактов над выражениями



Метамодель цепочек преобразования



Проверка решения по ответу

Решение правильно $\Leftrightarrow$ ответ удовлетворяет заданным критериям

- ☐ Подходит для широкого класса задач
- ☐ Несложен в понимании
- ☐ Распространен в системах дистанционного обучения

НО не проверяется путь, которым ответ получен. Не отлавливаются:

- ☐ Ошибки в процессе решения
- ☐ Решения, полученные непредусмотренным способом

Основная цель исследования

Разработать метод автоматической проверки решений задач на преобразование формул в целом, на основе проверки корректности и тривиальности всех переходов в решении.

$$C(x+1, x) = k$$

$$[C(x+1, x) = (x+1)! / x! / ((x+1) - x)! = (x+1)! / x! / 1! = (x+1)! / x! / 1 = x! * (x+1) / x! / 1 = x+1]$$

$$x+1 = k$$

$$(x+1) - 1 = k - 1 = >$$

$$x = k - 1$$



решение правильно / неправильно,
с указанием ошибочного перехода

Основные задачи исследования

1. Моделирование процесса решения учебных задач на развитие и закрепление навыков оперирования формулами.
2. Построение метода автоматической проверки решений задач.
3. Продумывание моделей представления решений в цифровом виде, в том числе создание языка для их записи.
4. Разработка программного комплекса, реализующего построенные алгоритмы и позволяющего автоматически проверять решения задач, поданных в разработанном формате.
5. Проведение экспериментальных исследования с участием студентов младших курсов и оценка эффективности разработанных моделей и алгоритмов.

Оценка сложности преобразований

Преподаватель задает

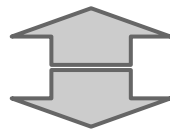
- ❑ Допустимые преобразования P с ненулевыми весами, соответствующими сложности преобразования для обучаемого
- ❑ Элементарные операции O , преобразования над которыми обучаемый может совершать неограниченно
 - ❑ Вместе с преобразованиями из P
 - ❑ Без преобразований из P

Преобразование может быть совершено за один переход, если существует сочетание комбинации преобразований из P с преобразованиями над элементарными операциями суммарным весом не более 1. (Элементарные операции имеют нулевой вес)

Орграф преобразований $G(V, E)$

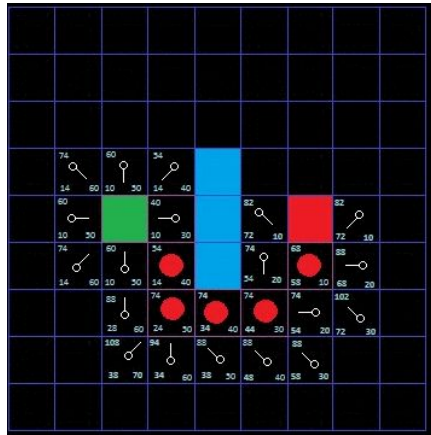
- ❑ Вершина $v \in V$:
 - ❑ соответствует выражению или факту
- ❑ Дуга $(u, v) \in E$:
 - ❑ соответствует наличию допустимого преобразования, с помощью которого выражение или факт, соответствующие u преобразуется в выражение или факт, соответствующие v .

Задача проверки
корректности и
тривиальности перехода
между L и R

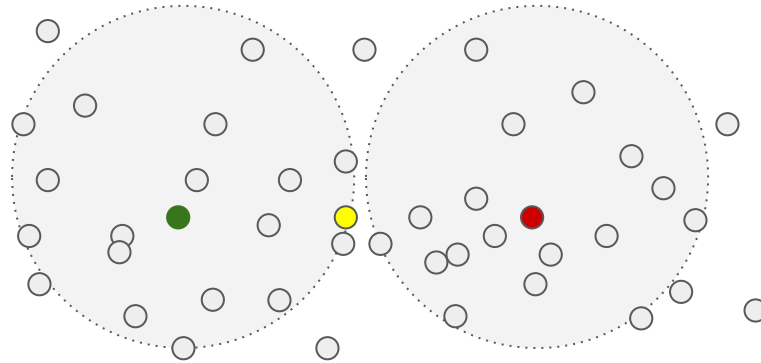


Задача проверки
существования / поиска
пути длины не больше N в
графе G между
вершинами,
соответствующими L и R

Идея перебора возможных преобразований



алгоритм A^*



Двусторонняя система продукций

Можно одновременно выполнять два экземпляра алгоритма A^* - использовать двустороннюю систему продукций с эвристикой для предварительной оценки расстояния

Оценка расстояния между выражениями (1)

Применима для $O = \{ '+', '-', '*', '/' \}$

1. Преобразование разности выражений ($L - R$):
 - 1.1. Избавление от операций деления: домножение ($L - R$) на все встретившиеся в разности делители
 - 1.2. Введение новых переменных, соответствующих операндам операций сложения, умножения, вычитания, не содержащим эти операциям (F - множество введенных переменных)
 - 1.3. В полученном выражении раскрываются все скобки и приводятся подобные слагаемые
 - 1.4. Получается многочлен K , зависящий от исходных переменных и переменных множества F
2. Расстояние между L и R ($\text{dist}(L, R)$) считается равным мощности множества F в многочлене K

$$L = P(n) + u!/m!$$
$$R = n! + (u!+k)/m!$$

$$L-R = (P(n) + u!/m!) - (n! + (u!+k)/m!)$$

$$1.1 (P(n)*m! + u!) - (n!*m! + (u!+k))$$

$$1.2 (a*b + d) - (c*b + (d+k))$$

$$1.3 a*b - c*b + k$$

$$1.4 K = a*b - c*b + k$$

$$\text{dist}(L,R)=|\{a,b,c\}|=\underline{3}$$

Оценка расстояния между выражениями (2)

Общий случай.

1. Все функции и операции, не входящие в O , заменяются переменными
2. Полученные выражения сравниваются методом вычислительных экспериментов
3. Если выражения равны - расстояние равно 0
4. Иначе расстояние равно числу вхождений введенных переменных

Алгоритм проверки преобразования ARCOMB

АЛГОРИТМ:

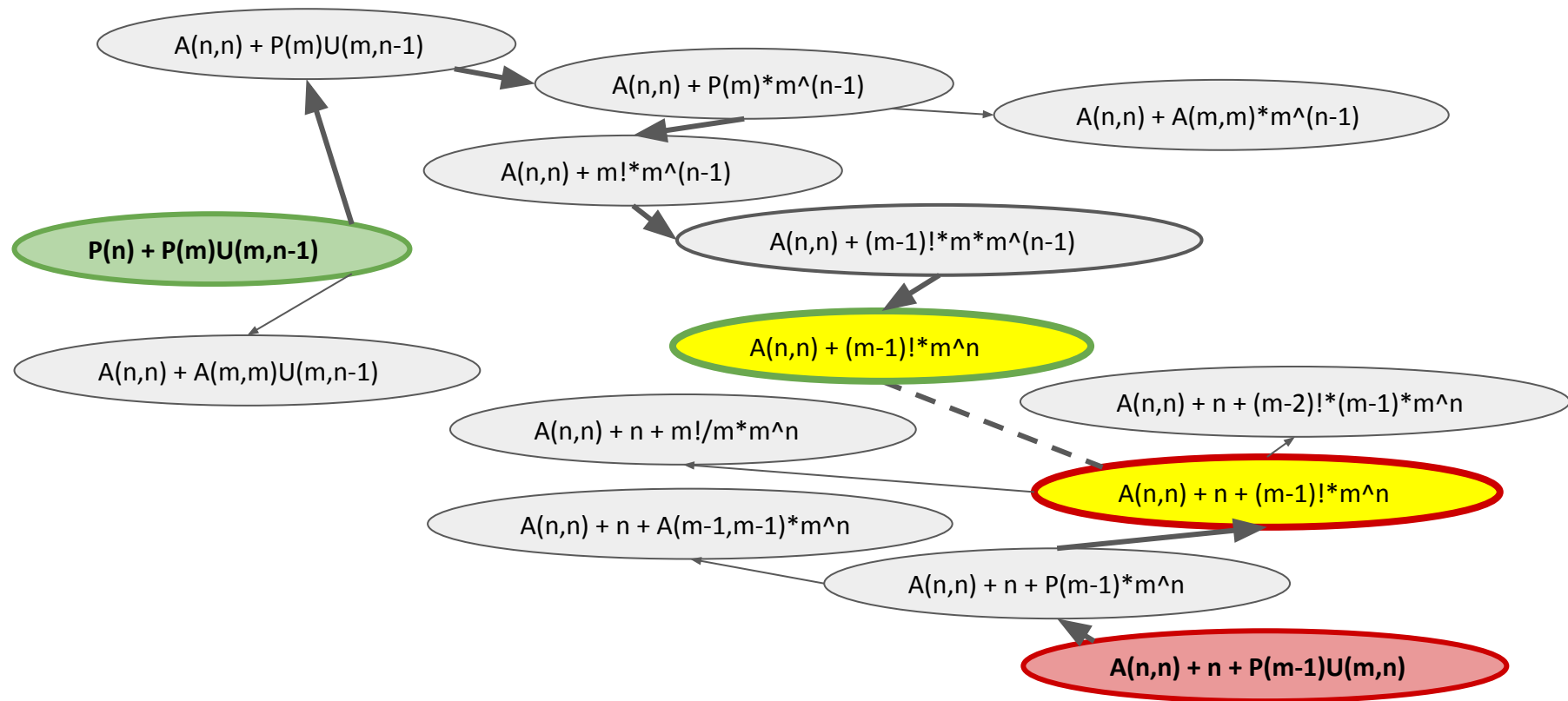
1. Для всех пар вершин (u, v) , соответствующих L и смежным для L вершинам, R и смежным для R вершинам соответственно:
 - 1.1. Если $\text{dist}(u, v) = 0$ - алгоритм завершает работу, переход между L и R корректен
2. Инициализация:
 - 2.1. В CL (CR) помещаем вершины, соответствующие L (R)
 - 2.2. В OL (OR) помещаем вершины, смежные соответствующим L (R)
3. Пока OL или OR не пустые:
 - 3.1. Если OL не пустая - выполняется процедура извлечения вершины из OL
 - 3.2. Если OR не пустая - выполняется процедура извлечения вершины из OR
4. Алгоритм завершает работу: переход между L и R либо некорректен, либо нетривиален

OL и OR - очереди открытых вершин для L и R
CL и CR - списки закрытых вершин для L и R

Процедура извлечения вершины u из очереди OL (OR):

1. Для каждой смежной u вершине v , такой что:
 - а). Расстояние от v до вершины, соответствующей $L(R)$ не больше N
 - б). $v \notin OL \cup OR \cup CL \cup CR$
 - 1.1. $\text{prior} = \min(v' \in CR (CL), \text{dist}(v, v'))$
 - 1.1.1. Если при вычислении prior оказалась, что $\text{dist}(v, v') = 0$ - алгоритм завершает работу
 - 1.2. Добавление v в $OL(OR)$ с приоритетом, обратным prior
2. Удаление u из $OL(OR)$; Помещение u в $CL(CR)$

Пример работы алгоритма



Эффективность алгоритма

- ❑ В худшем случае: $O((2 * |P|)^{N*2})$
 - ❑ оценка следует из необходимости рассмотреть каждую вершину, расстояние от которой до L (R) не больше N, а таких вершин $O((2 * |P|)^N)$, и для каждой из таких вершин требуется посчитать оценку расстояния еще до $O(|P|^N)$ вершин
- ❑ В проверяемых решениях задач большинство соседних выражений в цепочке преобразований имеют схожую структуру

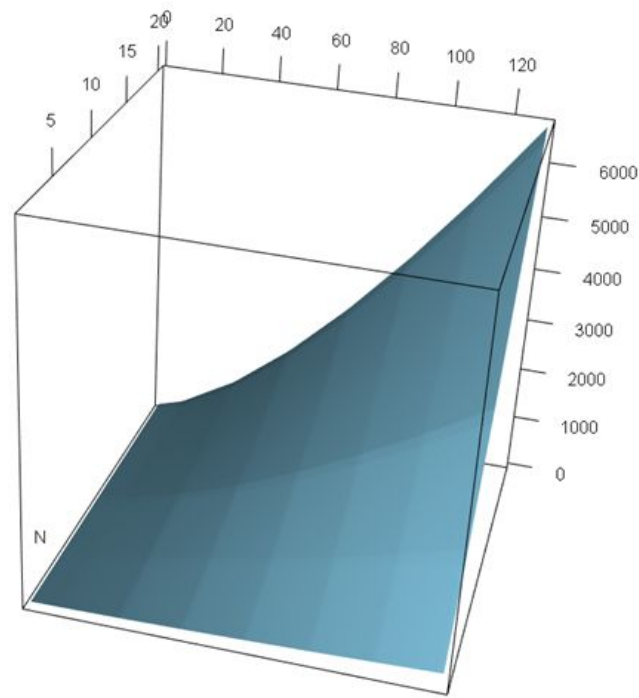


На практике ожидается, что алгоритм в большинстве случаев будет завершать работу по результатам вычисления расстояния и перебирать небольшой объем вершин

Экспериментальная оценка эффективности (от $|P|$ и N)

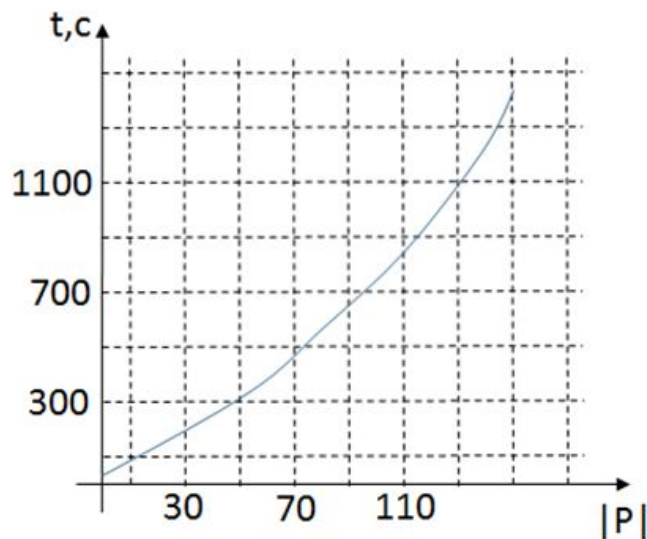
- ❑ Данные: 100 000 пар символьных выражений, полученных путем случайных замен терминальных символов из пар равных выражений, используемых при описании решений задач
- ❑ Результат: Усредненные значения t - времени работы алгоритма (в миллисекундах) для множества допустимых преобразований мощности $|P|$

N - среднее ожидаемое число преобразований из P , подтверждающих корректность преобразования



Экспериментальная оценка эффективности (от $|P|$ при $N = 1$)

- ❑ Данные: 100 000 пар символьных выражений, полученных путем случайных замен терминальных символов из пар равных выражений, используемых при описании решений задач
- ❑ Результат: Усредненные значения t - времени работы алгоритма (в миллисекундах) для множества допустимых преобразований мощности $|P|$



N - среднее ожидаемое число преобразований из P , подтверждающих корректность преобразования

Метод вычислительных экспериментов

- ❑ Для N случайных наборов значений переменных сравниваем значения выражений. Если хотя бы одно сравнение не проходит - считаем преобразование некорректным. Иначе - корректным
- ❑ Не проверяет тривиальность
 - ❑ Используем только для проверки преобразований над элементарными операциями
- ❑ Может зачесть ошибочный переход как правильный. Это наиболее вероятно при проверке
 - ❑ Переходов между логическими выражениями $\Rightarrow$ используем сведение к НФ
 - ❑ Переходов между выражениями с маленькой областью определения
 - ❑ Неравенств

Метод вычислительных экспериментов

- ❑ Для N случайных наборов значений переменных сравниваем значения выражений. Если хотя бы одно сравнение не проходит - считаем преобразование некорректным. Иначе - корректным
- ❑ Не проверяет тривиальность
 - ❑ Используем только для проверки преобразований над элементарными операциями
- ❑ Может зачесть ошибочный переход как правильный. Это наиболее вероятно при проверке
 - ❑ Переходов между логическими выражениями $\Rightarrow$ используем сведение к НФ
 - ❑ Переходов между выражениями с маленькой областью определения
 - ❑ Неравенств

Игрофикация задач на преобразование формул

Цели:

- ❑ Стимулировать обучаемых к решению большего числа задач
- ❑ Упростить взаимодействие обучаемых с интерфейсом, концентрировать их внимание на практикуемых навыках
- ❑ Усложнить заимствование результатов чужого труда

Модель игрофикации “Правило - место применения”

- ❑ Дано: выражение и цель, к которой его нужно свести
- ❑ Задача: найти сводящую цепочку преобразований

- ❑ Совершение преобразования:
 - ❑ Игрок выбирает правило из списка доступных
 - ❑ Если мест применения правила несколько, игрок выбирает одно из них
 - ❑ Выражение автоматически преобразуется по выбранному правилу

- ❑ Уровень считается пройденным, когда игрок сводит данное ему выражение к цели

Модель оказалась неудачной из-за слишком большого пространства выбора среди правил

Модель игрофикации “Место применения - правило”

- ❑ Дано: выражение и цель, к которой его нужно свести
- ❑ Задача: найти сводящую цепочку преобразований

- ❑ Совершение преобразования:
 - ❑ Игрок выбирает места применения правил преобразования
 - ❑ Игрок выбирает одно из применимых в этих местах правил преобразования
 - ❑ Выражение автоматически преобразуется по выбранному правилу

- ❑ Уровень считается пройденным, когда игрок сводит данное ему выражение к цели

Модель довольно точно эмулирует процесс решения задачи, концентрирует внимание на планировании дальнейших шагов исходя из сведений о правилах преобразования.

Модель игрофикации “Проверка решения”

- ❑ Дано: цепочка преобразований, генерируемых системой по заданным правилам, и корректным, и некорректным
- ❑ Задача: найти неправильные преобразования в цепочке
- ❑ Закрепляет знания правил преобразования в изучаемой предметной области
- ❑ Оперативно адаптируется под плохо усвоенные правила

Эксперименты еще не проводились

Текущие возможности разработанной системы

1. Автоматическая проверка решений учебных задач на преобразование выражений по формулам в областях теории множеств, комбинаторики, тригонометрии
2. Стандартный интерфейс для ввода решений
 - а. Поддерживаются MathML, TeX и PlainText из блокнотов Wiris и MathQuil
3. “Игровой” интерфейс для решения учебных задач
 - а. Решение задачи вводится в виде последовательности выбора мест преобразований выражения и правил их преобразования

Демо <https://www.mathhelper.online>

Android-приложение <https://play.google.com/store/apps/details?id=mathhelper.games.matify>

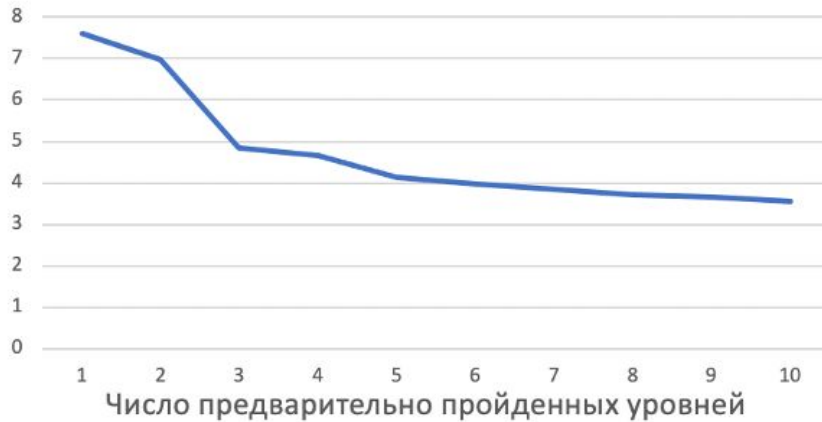
Сайт с примерами автопроверяемых задач <https://www.mathhelper.space/twf/prototype/demo.html>

Апробация на студентах технических ВУЗов

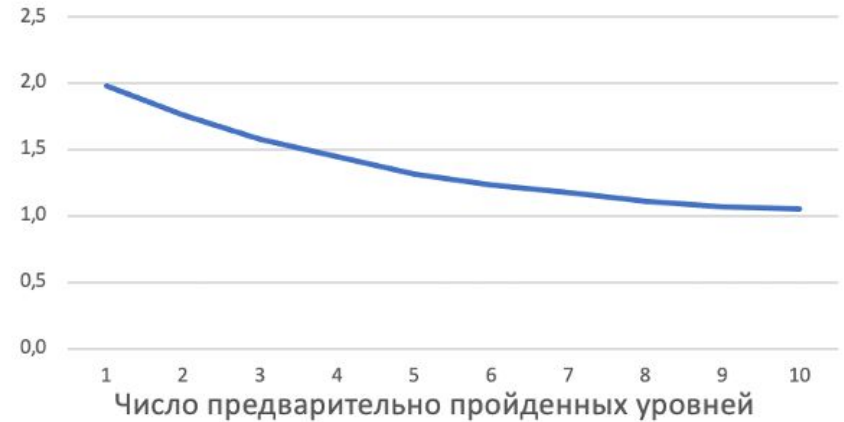
1. Более 95% студентов в состоянии в течение 10-20 минут освоить основные принципы разработанного языка и записывать на нем решения задач
2. Запись решений в предлагаемых программных комплексах занимает в среднем на 20% меньше времени, чем запись аналогичных решений на бумаге (при развитии навыка)
3. Эффективность преподавательского труда при проверке решений возросла более чем в 4 раза
4. Выявлены различия в оценке критериев тривиальности решений: в системе они жестче, чем в понимании студентов

Апробация игрофикации

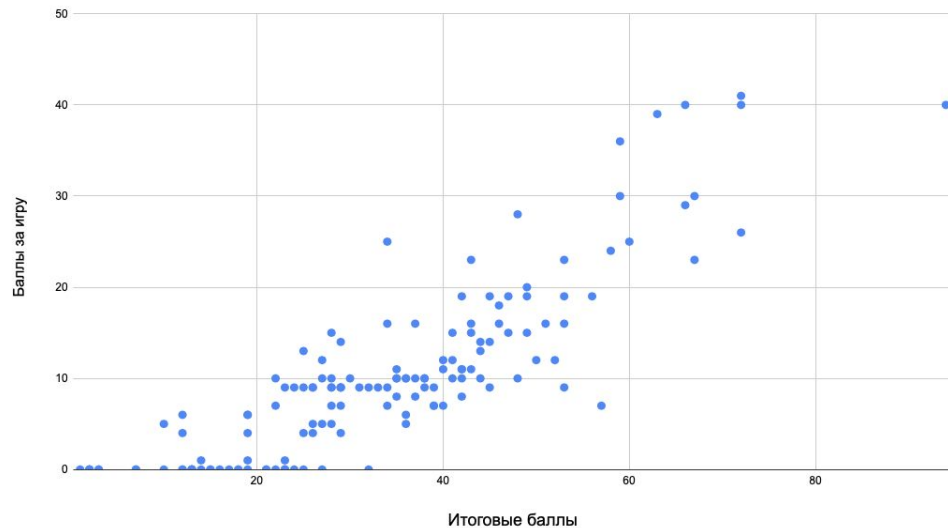
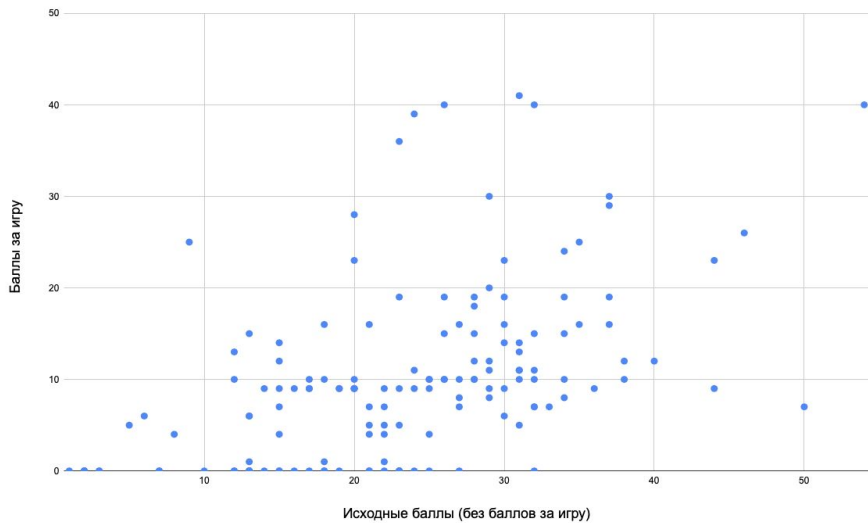
Отношение затраченного на решение времени
к сложности задачи



Отношение числа шагов в решении к
сложности задачи



Корреляция между баллами за игру и итоговыми



Направления дальнейших работ

1. Автоматическая генерация учебных задач
2. Адаптивный подбор задач в соответствии с индивидуальными потребностями обучающихся
3. Подбор точности вычислений в зависимости от структуры сравниваемых выражений
4. Уточнение функции оценки расстояния между выражениями и фактами
5. Поддержка новых типов преобразований
6. Внедрение разработанной системы, глубокое исследование применимости

Заключение

1. Смоделирован процесс решения учебных задач на преобразование формул, сформулированы требования к качественной проверки решений таких задач.
2. Предложены и применены модификации метода вычислительных экспериментов для проверки равенств и неравенств символьных выражений, позволяющие производить проверку с высокой точностью даже в случаях выражений, задающих немонотонные функции с маленькими областями определения.
3. Предложена и применена комбинация методов эффективного перебора правил и вычислительных экспериментов, которая позволяет с высокой точностью проверять корректности и тривиальность переходов, выполняемых студентами в ходе решения задач на преобразование формул.
4. Предложен язык записи символьных преобразований, дающий обучаемым возможность составлять решения задач на преобразование формул в цифровом виде.
5. Разработан и внедрен программный комплекс, позволяющий обучаемым решать задачи и записывать их решения в цифровом виде, а затем проверять их решения автоматически.
6. Предложена модель игрофикации процесса решения задач на развитие навыков оперирования формулами, который позволяет стимулировать обучаемых к практикованию навыков и повышению числа решаемых ими задач.
7. Экспериментально подтверждена эффективность использования разработанных комплексов в образовательном процессе для повышения качества обучения техническим наукам школьников старших классов и студентов младших курсов.