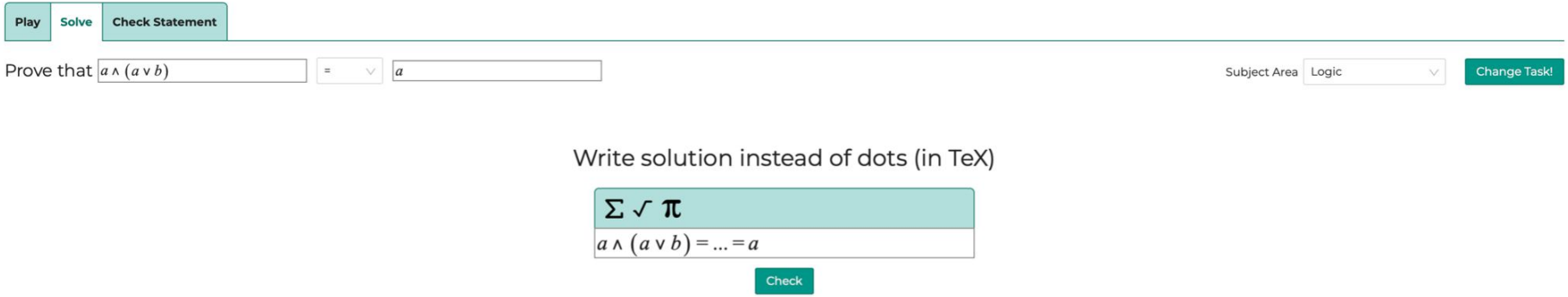
Автогенерация задач на преобразование формул

Багрянова Екатерина Андреевна
научный руководитель Виктор Кацман
НИУ ВШЭ – Санкт-Петербург

19 июня 2021 г.

Введение

mathhelper.online



Цели и задачи

- Разработать алгоритм, генерирующий задачу по настройкам, которые задает преподаватель
- Придумать, какими должны быть эти настройки

Аналоги

- Статья о генерации текстовых задач для школьников (на примере задачи о нахождении НОД двух чисел)
  - Общее:
    - Попытки придумать задачи примерно одинаковой сложности
    - Желание видеть достаточно красивые данные в задаче
  - Различия:
    - В моей работе формулы, здесь текстовые задачи
    - В моей работе важен ход решения, здесь только ответ
    - В целом генерация такой задачи кажется гораздо более простой
- Статья о генерации задач по линейной алгебре
  - Общее:
    - Генерация от красивого ответа
  - Различия:
    - Применялись специфические подходы для линейной алгебры

Исходный генератор

- Берет красивое начальное выражение
- Рандомно его преобразовывает
- Настройки
  - Разрешенные подстановки
  - Количество шагов генерации

Проблемы:

- Конечное выражение сложнее, чем начальное
- Выражения очень некрасивые
- Обратные преобразования

Начало работы

- Выбрала область работы – тригонометрию
- Изучала задачи из задачников и егэ, пыталась их классифицировать и придумывать похожие сама
- Разбиралась с существующим генератором и языком Kotlin

Технологии

- Новый для меня язык Kotlin
- Библиотека TWF, которая была написана участниками проекта ранее
  - Структуры для хранения и работы с выражениями и преобразованиями
  - Способы задания выражений
  - Алгоритмы для работы с выражениями и преобразованиями

Процесс работы

- Генерация от красивого выражения
- Теперь генерация от конечного выражения к начальному
  - Задания стали больше похожи на реальные
- Отсутствие переходов в предыдущие состояния генерации
  - Концепция количества шагов стала более логичной
  - Получающиеся выражения стали сложнее на вид
  - Получающиеся выражения стали более однородные по сложности

Настройки. Новые идеи

Настройки:

- Разрешенные подстановки
- Сложность выражения вместо количества шагов в генерации

Новые идеи:

- На сложность выражения влияет путь по преобразованиям от начального выражения к конечному
- При применении преобразований мы обычно хотим получать достаточно красивые выражения

Паттерн чекеры и новый генератор

Паттерн чекеры

- Если в выражении присутствует какой-то конкретный паттерн, оно штрафуется на соответствующее этому паттерну количество баллов
- Штраф зависит от того, насколько паттерн “некрасивый”
- Со штрафами можно играться и получать разные результаты генерации.

Новый генератор

- На каждом шаге генерации есть не одно выражение, а набор
- К каждому выражению из этого набора применяются несколько случайных преобразований
- Из получившихся результатов выбираются выражения с минимальным штрафом для следующего шага генерации

Текущие результаты

- Немного видоизменилась концепция настроек
- Генерируются более хорошие задачи

Изначальный генератор

$$\sin^2 x + \cos^2 x \longrightarrow 1$$

$$\sin^2 x + \cos^2 x \longrightarrow \left(1 - \left(\left(1 - \left(-\sin^2(x + \pi)\right)\right)^{0.5}\right)^2\right) + \left(-\sin^2 x + 1\right)$$

Генератор без паттерн чекеров

$$\left(1 - \left(\left(1 - \left(-\sin^2(x + \pi)\right)\right)^{0.5}\right)^2\right) + \left(-\sin^2 x + 1\right) \longrightarrow \sin^2 x + \cos^2 x$$

$$\left(-\left(\left(-\left(-\left(1 - \cos^2(x + \pi)\right)^{0.5}\right)\right) \cdot \left(-\cos(\pi)\right) + \sin(\pi) \cdot \cos(x)\right)\right)^2 + \cos^2 x \longrightarrow \sin^2 x + \cos^2 x$$

$$\cos^2 x + \frac{1}{\frac{1}{\frac{1}{\frac{1}{tg^2 x} + 1}}} \longrightarrow \sin^2 x + \cos^2 x$$

Генератор с имеющимися паттерн чекерами

$$\cos x \cdot \cos x + \sin(\pi - x) \cdot \sin x \longrightarrow \sin^2 x + \cos^2 x$$

$$\frac{1}{1 + ctg\, x \cdot ctg\, x} + \left(-(\cos x)^2\right) \longrightarrow \sin^2 x + \cos^2 x$$

Планы на будущее

- Добавить паттерн чекеров
- Начать работу со сложностью выражения (пока в планах через концепцию путей между выражениями по преобразованиям)

Ссылка на репозиторий

bitbucket.org/vkatsman/twf/src/master/