Лицей «Физико-техническая школа» Санкт-Петербургского Академического университета

Курсовая работа (отчет по практике)

Создание программы генериующей кроссворды из регулярных выражений

Работу выполнили:
Данилевич Леонид (2022А)
Лельчук Александр (2022А)
Научный руководитель:
Дворкин Михаил Эдуардович
Место прохождения практики:
Лицей «ФТШ»

Санкт-Петербург, 2021

Аннотация

Мы смогли разработать алгоритм генерации строк, которые подходят под различне человекочитаемые паттерны (например: палиндромы, прогресии, словарные ключи, повторы и так далее), которые могли бы легко восприниматься человеком. Также мы смогли создать уникальный алгоритм генерации наиболее короткого регулярного выражения для заданной строки.

Кроссворды из регулярных выражений

Глава 1

Введение

Регулярные выражения — формальный язык поиска подстрок в тексте и манипуляций с ними. Например, регулярному выражению «.*amp(le)?» соответствуют строки «Sample», «example», «lamp» и некоторые другие. С помощью регулярных выражений можно достаточно легко искать в тексте подстроки определённого формата и заменять их на соответствующие им другие подстроки. В 2013 году, как задание конкурса «MIT Mystery Hunt», был создан кроссворд из регулярных выражений (рис. 1.1). Мы решили написать приложение, автоматически генерирующее подобные кроссворды, а также позволяющее их решать. Мы считаем, что такое приложение будет полезно многим людям, изучающим регулярные выражения, а для уже знакомых с ними оно будет просто интересно.

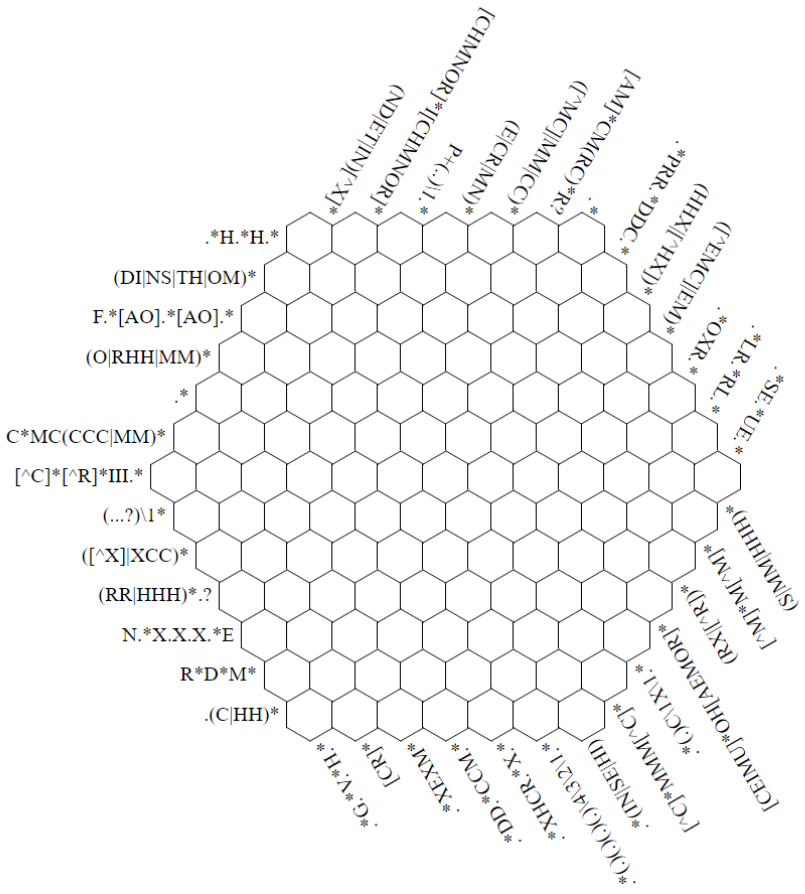

Рис. 1.1: Кроссворд с «MIT Mystery Hunt» 2013 года

Глава 2

Постановка задачи

Создать приложение на базе Android, функционалом которого является генерация, отображение и поддержка решения пользователем красивых кроссвордов из регулярных выражений.

Кроссворд из регулярных выражений – клеточная сетка правильной формы (прямоугольник, шестиугольник и т. д). Решением кроссворда называется такое сопоставление символа каждой клетке, что линии, образованные этими символами (например, в прямоугольнике это столбцы и строчки) формируют буквенные строчки, кодирующиеся соответственными регулярными выражениями.

Мы называем кроссворд из регулярных выражений красивым, если он имеет единственное решение. В нём регулярные выражения, использующиеся для кодирования линий символов, разнообразны (используется большое количество конструкций: символьные классы, группы, перечисления, квантификаторы, обратные ссылки и т. д.). В полученных линиях символов должны наблюдаться паттерны, которые легко воспринимаются человеком.

Глава 3

Методика решения задачи

Так как задача состоит в написании программы, то решается программно. Задача дробится на части:

3.1 Программное решение кроссворда

Собственно, решение уже имеющегося кроссворда. Понадобится для оценки сложности кроссворда и и удостоверения единственности решения.

3.1.1 Разбор регулярных выражений

Для решения произвольного кроссворда необходимо разобрать («распарсить») регулярные выражения, кодирующие его строчки. Для этого написана рекурсивная функция с запоминанием ответа (техника «мемоизации» динамического программирования), которая для заданного регулярного выражения и требуемой длины строчки находит в компактном виде всевозможные строчки, подходящие под данное выражение. Компактный вид обусловлен тем, что строчки состоят не из символов, а из битовых масок, где из любой маски можно выбрать любой символ и получить корректную строчку. Обратные ссылки в регулярных выражениях обрабатываются следующей техникой: два символа, запрашиваемые быть одинаковыми, соотвествуют одному и тому же битовому объекту-маске.

3.1.2 Отгадывание букв

После разбора регулярных выражений кроссворд решается методом инкрементальных улучшений. Изначально каждой букве сопоставлена битовая маска, соответствующая всем символам алфавита. Алгоритм рассматривает все ещё не разгаданные буквы по порядку, сужая для каждой множество возможных значений. Существует вероятность встретить кроссворд, имеющий решение, но нерешаемый данной техникой. Однако для его разгадывания человеку потребуется долгий и утомительный перебор. Создание таких кроссвордов не входит в нашу задачу. В случае полного разгадывания кроссворда можно утверждать, что его решение существует и единственно.

3.2 Генерация буквенного заполнения кроссворда

Для красоты кроссворда необходимо образование строк, подходящих под различные регулярные выражения. Максимизируется красота заполнения — субъективное свойство, включающее в себя регулярность кроссворда. Именно, максимизируется количество и длина вхо-

дящих в кроссворд подстрок, являющихся шаблонами следующих типов: палиндром (строка, читающаяся одинаково слева направо и справа налево); повтор (строка, состоящая из одинаковой части, повторённой несколько раз); прогрессия (строка, в которой на нечётных (в 1-индексации) местах стоят одинаковые буквы, а на чётных - разные; словарный ключ - существующее английское слово.

3.2.1 Имитация отжига

Для генерации случайного поля, максимизирующего потенциальную оценку кроссворда, а именно, разнообразие типов регулярных выражений, разгадка необходимой степени сложности, и субъективная оценка красоты, применён метод имитации отжига («simulated annealing»). Большое количество итераций заменяют одну букву на изначально равномерно случайно сгенерированном поле, после чего в некотором случае изменение принимается, поле обновляется.

3.3 Генерация регулярных выражений

Генерация регулярных выражений, лучшим образом задающих получившиеся строчки. Пока не реализована.

3.4 Оценка сложности, проверка единственности решения

Оценку сложности и проверку единственности решения планируется осуществлять путём программного решения. Сложность решения предполагается оценивать, как усреднённое значение количества итераций, требуемых алгоритму для решения сгенерированного кроссворда при рассмотрении клеток, содержащих неразгаданные буквы, в случайном порядке.

Глава 4

Результаты

4.1 Программное решение кроссворда

Получившаяся программа успешно разгадывает кроссворды. При вводе регулярных выражений из кроссворда, упоминавшегося во введении (рис. 1.1), программа выдаёт следующее решение (рис. 4.1)

```
S E C U E M C
M L R C R L M C
M M X O X R X M H
H E M H E M H E M H
H H X M I R H H X D C
H P R R M I O H H D D C
S T X M C M I E C R X R G
A M A M M C M R C R C R
H O X M M C C O X R N
E M N M N C R E C R
P O X O X C X R V
H I O M C M R O
N D F M M C H
```

Рис. 4.1: Решение кроссворда (рис. 1.1)

4.2 Генерация буквенного заполнения кроссворда

. Алгоритм получилось написать. Получились довольно хорошо удовлетворяющие условиям кроссворды (рис. 4.2): Рассмотрим подробнее результат работы отжига. Выделю некоторые существенные паттерны в одном из кроссвордов (рис. 4.3)

4.3 Генерация регулярных выражений

Генерация регулярных выражений, лучшим образом задающих получившиеся строчки. Пока не реализована.

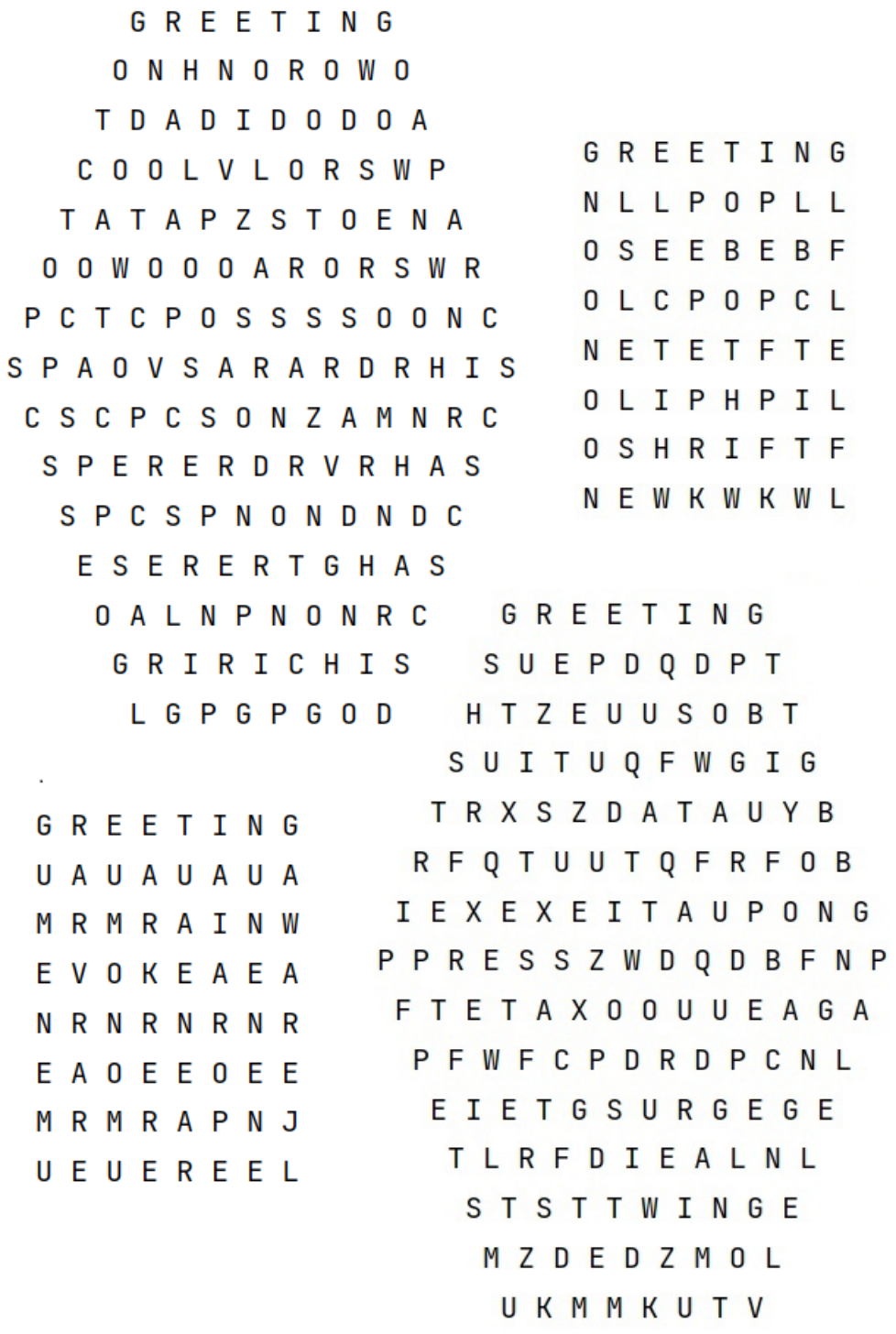

Рис. 4.2: Сгенерированные кроссворды

4.4 Оценка сложности, проверка единственности решения

Решение кроссворда заодно проверяет и единственность данного решения. Оценка не реализована, так как оценивать нечего, а именно, по причине отсутствия генерации регулярных выражений. По появлении генерации регулярных выражений сделать оценку будет достаточно просто: лишь слегка модифицировать само решение кроссворда, чтобы помимо решения, оно возвращало усреднённое количество понадобившихся итераций при случайном порядке рассмотрения клеток.

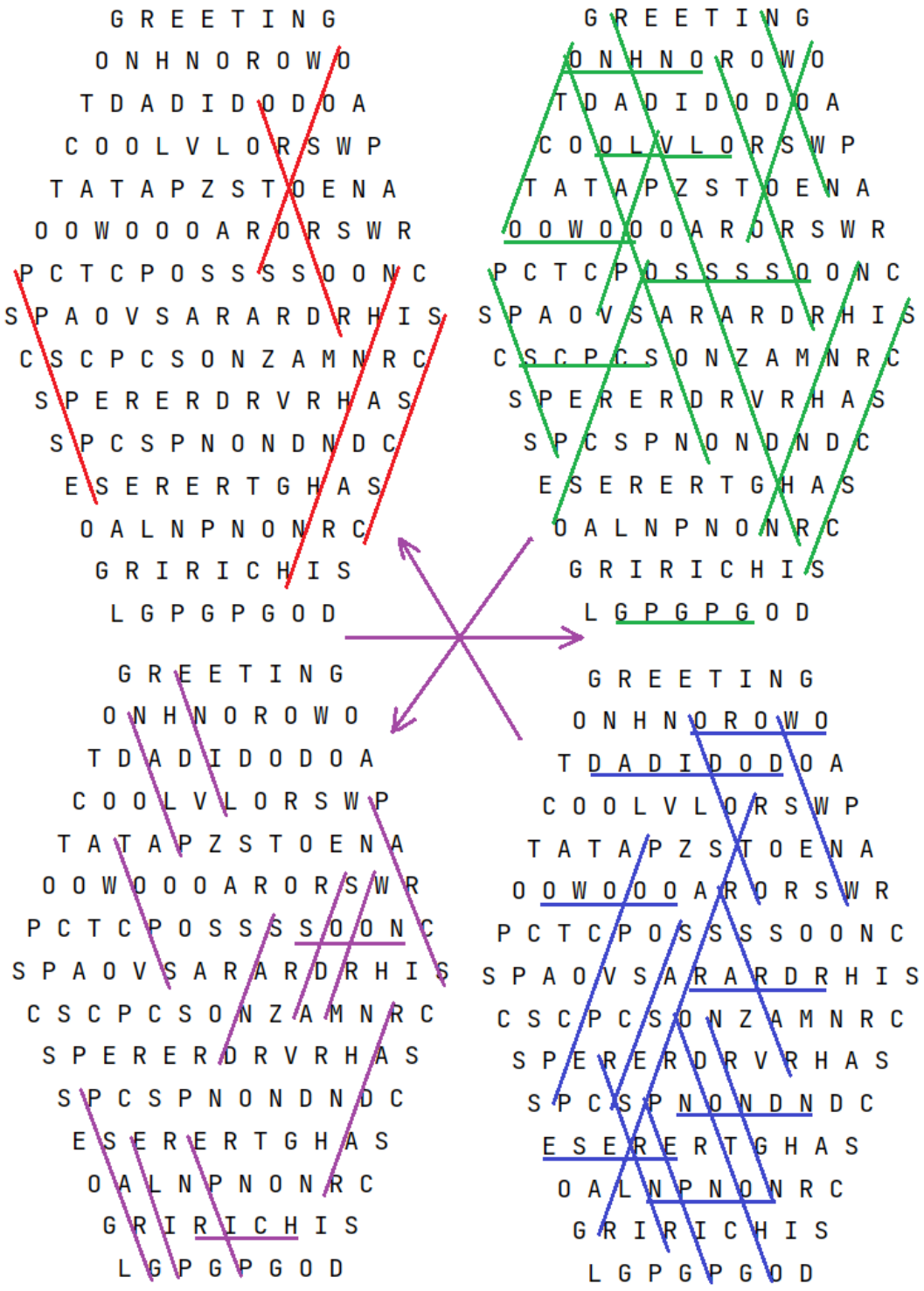

Рис. 4.3: Слева направо, сверху вниз: повторы, палиндромы, слова, прогрессии

Глава 5

Анализ результатов

Пока результаты не очень большие, но мы сделали многое из того, что хотели.
Мы получили новый алгоритм, которого раньше не существовало, и который в перспективе может использоваться не только в нашем приложении, но также для генерации шаблонов для описания различных строк на основании только лишь одной входной строки.

Глава 6

Благодарности

Мы благодарим нашего научного руководителя **Дворкина Михаила Эдуардовича** за научное руководжение и направление теоретической части нашей работы.