



Санкт-Петербургский государственный университет
Математическое обеспечение и администрирование
информационных систем

Алгоритм в терминах линейной алгебры для поиска путей от нескольких стартовых вершин с регулярными ограничениями

Порсев Денис Витальевич, 19.Б10-мм

9 июня 2023

Научный руководитель: доцент кафедры информатики, к.ф.-м.н., С.В. Григорьев

Рецензент: эксперт ООО “Техкомпания Хуавэй” С.В. Моисеев

Санкт-Петербург
2023

- Графовая модель данных
 - ▶ Уникальный Id вершины
 - ▶ Вершины имеют свойства в формате ключ-значение
 - ▶ Метка из \mathcal{L} на каждом ребре
- Применение:
 - ▶ Анализ социальных сетей
 - ▶ Биоинформатика
 - ▶ Графовые базы данных (Redis Graph, Neo4j)

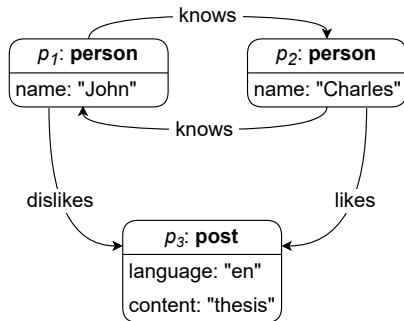
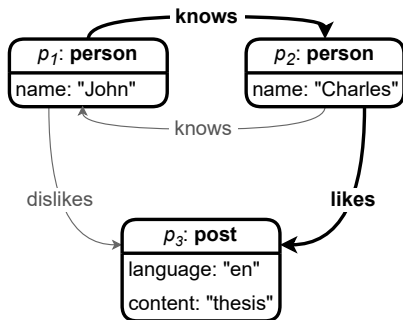


Рис.: *

$\mathcal{L}: \{knows, likes, dislikes\}$

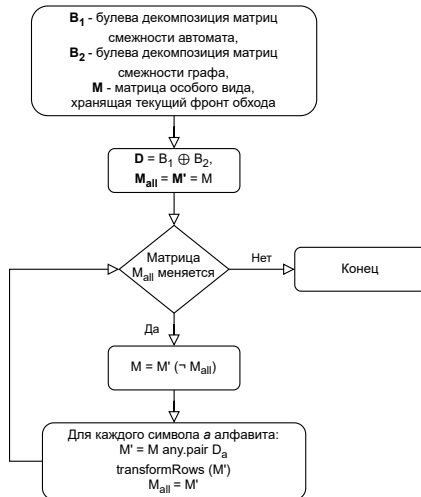
Поиск путей с регулярными ограничениями (RPQ)

- Запрос к графовой БД
- Ограничения на пути в графе в виде регулярного языка
- *Property paths* в SPARQL v1.1
 - $?person :knows^* ?person .$
 - $?person (:knows/:likes)^+ ?post .$
- Частичная поддержка в Cypher (Neo4j)
 - $(person)-[:knows^*]->(person)$



Разработанный ранее алгоритм (MSBFS)

- Решает задачу RPQ
- Основан на матричном умножении
- Разреженные матрицы смежности
- BFS выражен в терминах линейной алгебры
- Реализован с помощью PyGraphBLAS¹



¹Обёртка над библиотекой примитивных операций с разреженной лин. алгеброй SuiteSparse:GraphBLAS

Постановка задачи

Цель: проведение экспериментального исследования алгоритма для решения задачи RPQ, основанного на поиске в ширину и выраженного в терминах линейной алгебры

Задачи:

- Выбрать множество аналогов для проведения сравнения с ними
- Подготовить датасет, состоящий из графов и регулярных запросов
- Спроектировать инструмент автоматизации экспериментов
- Провести экспериментальное исследование алгоритмов и проанализировать результаты

Существующие решения задачи RPQ

- Решения, основанные на использовании конечных автоматов и различных поисков (например, BFS)
- Datalog
 - + Выразителен
 - + Является стандартным бенчмарком для сравнения
 - Необходимо самостоятельно реализовывать запросы с ограничениями в виде регулярных языков
- Решения, строящие индекс по путям в графе
 - + Получение результата за один просмотр индекса
 - Большой расход памяти

Существующие решения задачи RPQ

MSBFS

- Решения, основанные на использовании конечных автоматов и различных поисков (например, BFS)
 - ▶ Выразимы с помощью матричных операций

- Datalog

- + Выразителен
- + Является стандартным бенчмарком для сравнения
- Необходимо самостоятельно реализовывать запросы с ограничениями в виде регулярных языков

- Решения, строящие индекс по путям в графе

- + Получение результата за один просмотр индекса
- Большой расход памяти

Выбор аналогов

- Реализация тензорного алгоритма в репозитории CFPQ_Pyalgo²
 - ▶ Представление запросов:

$$I^* \Rightarrow S \rightarrow I \mid S \mid \epsilon$$

- Souffle — реализация Datalog, часто используемая в задачах языкового анализа и набирающая большую популярность
 - ▶ Генерация программ Datalog:

$$\begin{array}{c} I^* \\ \Downarrow \\ \textit{path}(x, y) :- \textit{edge}(x, l, z), \textit{path}(z, y). \\ \textit{path}(x, y) :- \textit{edge}(x, l, y). \end{array}$$

- Алгоритмы, строящие индексы, не вошли в сравнительный анализ
 - ▶ Большинство встроены в системы баз данных

²Репозиторий для исследования алгоритмов поиска путей с ограничениями в виде формальных языков, реализованных на основе стандарта GraphBLAS

Сбор данных: графы

Данные:

- RDF-графы
- Социальные сети

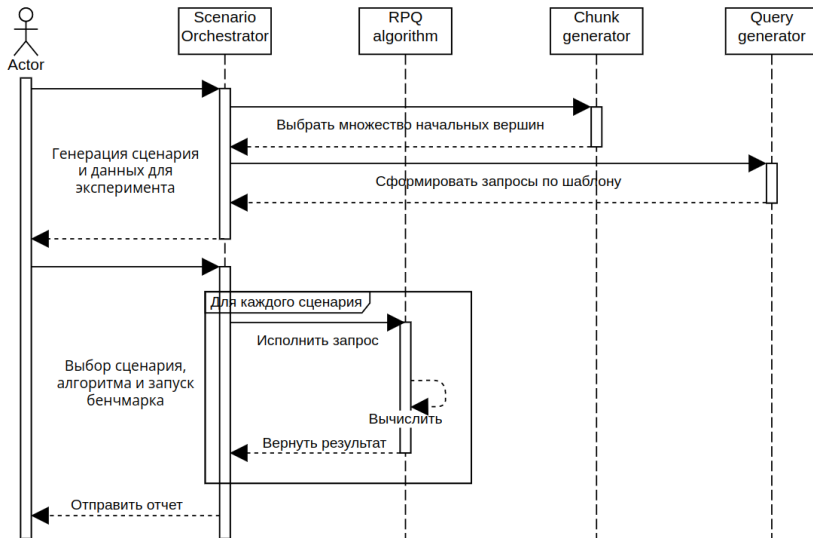
Graph	#V	#E	#L
enzyme	48 815	86 543	14
eclass	239 111	360 248	10
go	582 929	1 437 437	47
geospecies	450 609	2 201 532	158
taxonomy	5 728 398	14 922 125	21
advogato	6 541	51 127	3
youtube	15 088	27 257 790	5

Сбор данных: запросы

- Набор из 16 популярных шаблонов для запросов
- Генерация запросов с самыми популярными метками

Name	Query	Name	Query
q_0	a^*	q_8	$a \cdot b$
q_1	$a \cdot b^*$	q_9	$a \cdot b \cdot c$
q_2	$a \cdot b^* \cdot c^*$	q_{10}	$a \cdot b \cdot c \cdot d$
q_3	$a \cdot b^* \cdot c$	q_{11}	$(a \cdot b)^+ \mid (c \cdot d)^+$
q_4	$a^* \cdot b^*$	q_{12}	$(a \cdot (b \cdot c)^*)^+ \mid (d \cdot e)^+$
q_5	$a \cdot b \cdot c^*$	q_{13}	$(a \cdot b \cdot (c \cdot d)^*)^+ \mid (e \mid f)^*$
q_6	$(a \mid b \mid c \mid d \mid e)^+$	q_{14}	$(a \mid b)^+(c \mid d)^+$
q_7	$(a \mid b \mid c \mid d \mid e) \cdot f^*$	q_{15}	$a \cdot b \cdot (c \mid d \mid e)$

Инструмент для автоматизации экспериментов



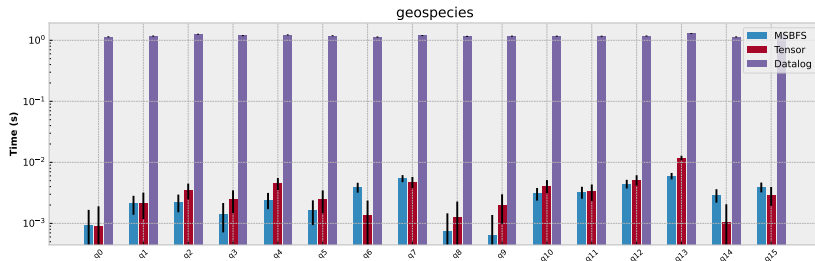
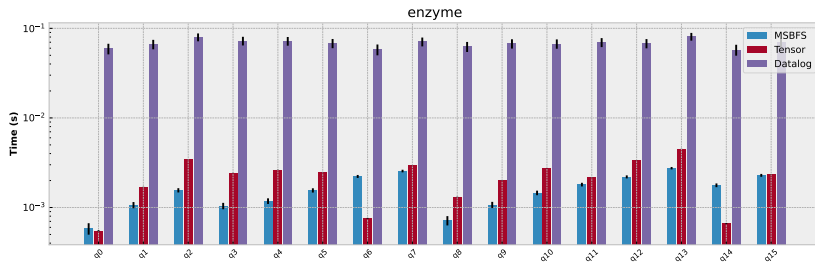
Конфигурация:

- Ubuntu 20.04, процессор Intel i7-4790 CPU @ 3.60GHz CPU, оперативная память DDR4 64 Gb

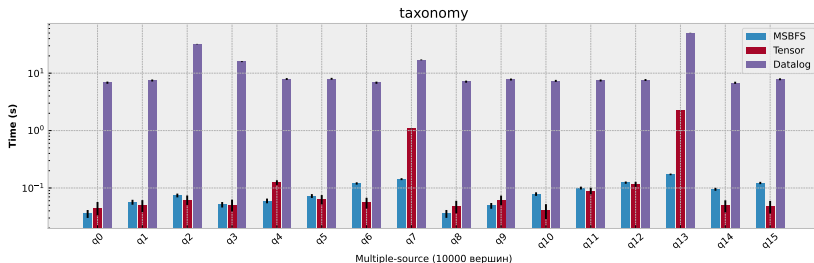
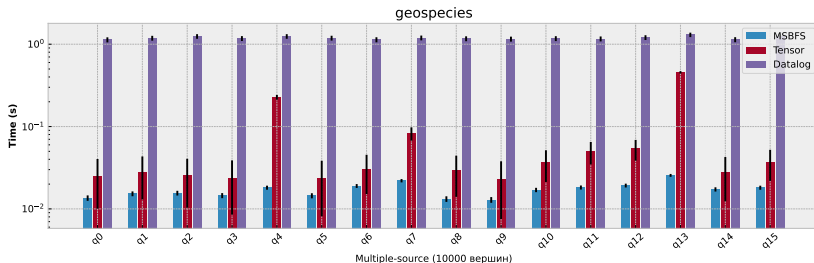
Исследовательские вопросы:

- B1:** Какова производительность разработанного алгоритма по сравнению с существующими аналогами?
- B2:** Как влияет размер множества стартовых вершин на производительность реализации разработанного алгоритма?

B1: Single-source запросы



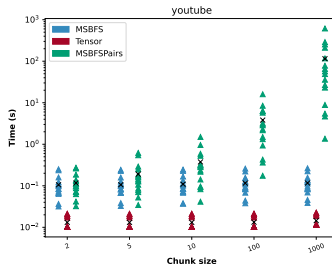
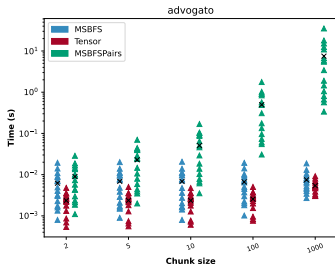
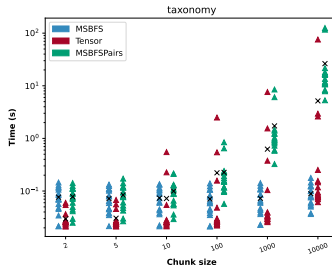
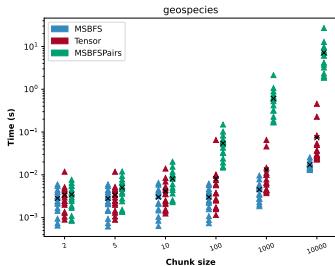
V1: Multiple-source запросы (10 000 стартовых вершин)



В2: постановка эксперимента

- Число стартовых вершин: 1, 2, 5, 10, 100, 1000, 10000
- Две реализации MSBFS:
 - ▶ MSBFS находит множество достижимых вершин
 - ▶ MSBFSPairs находит множество достижимых вершин для каждой стартовой вершины
- Сравнение с тензорным алгоритмом
- Взяты самые большие RDF графы + графы социальных сетей

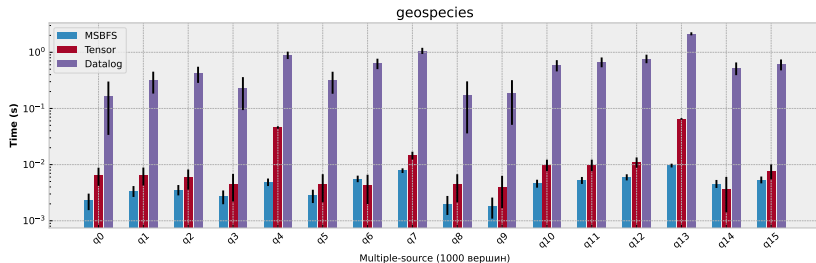
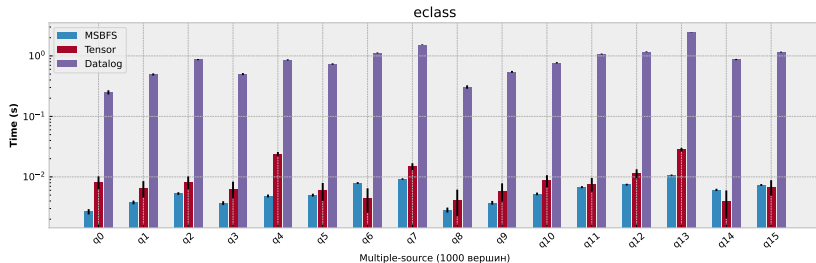
В2: Размер множества стартовых вершин



- Проведен обзор и выбрано два аналога для сравнения: тензорный алгоритм, Datalog (Souffle)
- Собран датасет из графов и регулярных запросов
- Разработан инструмент³ для автоматизации экспериментов
- Проведено экспериментальное исследование нового алгоритма:
 - ▶ Алгоритм MSBFS показывает приемлемое время работы, опережает Datalog
 - ▶ На графах RDF алгоритм MSBFS в среднем более производителен аналогов, на графах социальных сетей тензорный алгоритм оказался самым быстрым
 - ▶ При увеличении числа стартовых вершин незначительное падение производительности реализации MSBFS множество-множество; для MSBFSPairs наблюдается сильный рост времени исполнения

³<https://github.com/bahbyega/paths-benchmark>

Дополнительно B1: Multiple-source запросы (1 000)



Дополнительно: Алгоритм в псевдокоде

Algorithm 1 Алгоритм в терминах линейной алгебры для поиска путей от нескольких стартовых вершин с регулярными ограничениями

```
1: procedure MSBFS ( $\mathcal{R} = \langle Q, \Sigma, P, F, Q_s \rangle, \mathcal{G} = \langle V, E, L \rangle, V_s$ )
2:    $k \leftarrow |Q|, n \leftarrow |V|$ 
3:    $\mathcal{M}_A \leftarrow$  булева декомпозиция матрицы смежности для  $\mathcal{R}$ 
4:    $\mathcal{M}_G \leftarrow$  булева декомпозиция матрицы смежности для  $\mathcal{G}$ 
5:   for all  $q \in Q_s$  do
6:     for all  $v \in V_s$  do
7:        $M[q, q + v + 1] \leftarrow 1$  ▷ Где  $M^{k \times (k+n)}$  с 1 на главной
      диагонали
8:     for all  $a \in (\Sigma \cap L)$  do
9:        $\mathcal{D}_a \leftarrow \mathcal{M}_A \oplus \mathcal{M}_G$ 
10:     $M' \leftarrow M, M_{all} \leftarrow M$ 
11:    while Матрица  $M_{all}$  меняется do
12:       $M \leftarrow M' \langle \neg M_{all} \rangle$ 
13:      for all  $a \in (\Sigma \cap L)$  do
14:         $M' \leftarrow M \text{ any.pair } \mathcal{D}_a$  ▷ Матр. умножение в полукольце
15:         $M' \leftarrow \text{TransformRows}(M')$  ▷ Приведение  $M'$  к виду  $M$ 
16:         $M_{all} \leftarrow M'$ 
17:    return  $M_{all}$ 
```
