

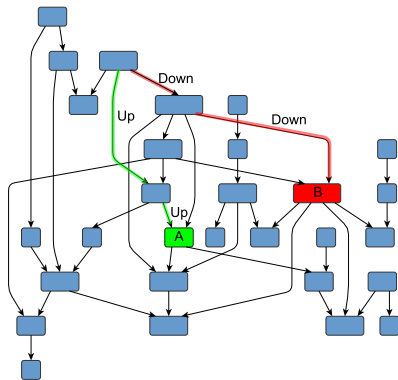
# Поиск путей в графе с КС ограничениями через произведение Кронекера

**Егор Орачев, Илья Эпельбаум**  
Ментор: Семен Григорьев

JetBrains Research, Лаборатория языковых инструментов  
Санкт-Петербургский Государственный университет

31 Августа 2020

# Context-Free Path Querying



## Навигация в графе

- Находятся ли вершины A и B на одном уровне иерархии?
- Существует ли путь вида  $Up^n Down^n$ ?
- Найти все такие пути  $Up^n Down^n$ , которые начинаются в вершине A

- $\mathbb{G} = (\Sigma, N, P)$  — контекстно-свободная грамматика
  - ▶  $\Sigma$  конечное множество терминалов
  - ▶  $N$  конечное множество нетерминалов
  - ▶  $P$  конечное множество правил вывода
  - ▶  $L(\mathbb{G}, A) = \{\omega \mid A \Rightarrow^* \omega\}, A \in N$
- $G = (V, E, L)$  — ориентированный граф с метками
  - ▶  $v \xrightarrow{l} u \in E$
  - ▶  $L \subseteq \Sigma$
- $\omega(\pi) = \omega(v_0 \xrightarrow{l_0} v_1 \xrightarrow{l_1} \dots \xrightarrow{l_{n-2}} v_{n-1} \xrightarrow{l_{n-1}} v_n) = l_0 l_1 \dots l_{n-1}$
- $R_A = \{(n, m) \mid \exists n \pi m, \text{ такой что } \omega(\pi) \in L(\mathbb{G}, A)\}, A \in N$

# Существующие решения

- Решения, основанные на различных техниках парсинга (CYK, LL, LR, etc.)
- Решения, основанные на матричных операциях
- Существующие решения слишком *тяжеловесны* для запросов, выраженных регулярными ограничениями<sup>1</sup>
- Все существующие решения работают только с КС грамматикой в нормальной форме (Нормальная форма Хомского)
- **Трансформация требует дополнительного времени на обработку и приводит к разрастанию грамматики**

---

<sup>1</sup>Regular path quering (RPQ) — поиск путей с регулярными ограничениями

# Мотивация разработки нового алгоритма

- Работает с достаточно *хорошим* представлением входной грамматики, которое не требует дополнительной трансформации в нормальную форму и не приводит к увеличению числа продукций
- Реализуется в терминах операций линейной алгебры
- Приемлем для запросов, выраженных регулярными ограничениями
- Позволяет получать информацию не только о достижимости вершин, но и о путях, соединяющих эти вершины

- Формализация алгоритма, описание *наивной* реализации в категории операций линейной алгебры с использованием булевой матричной декомпозиции
- Оформление текста статьи для конференции **2020 ACM SIGMOD**
- Подготовка **LUBM** датасета для тестирования регулярных запросов

- Реализация *наивной* версии алгоритма на основе библиотеки **PyGraphBLAS**<sup>2</sup>
- Реализация на основе **PyGraphBLAS** извлечения путей из *индекса*, построенного в ходе выполнения основного алгоритма
- Проведение CFPQ и RPQ замеров работы алгоритма построения индекса и алгоритма извлечения путей

---

<sup>2</sup>PyGraphBLAS — python-обертка для SuiteSparse C — реализации GraphBLAS API для работы с графами в терминах операций линейной алгебры

- Выполнены задачи по разработке и реализации алгоритма
- Получены *оптимистичные* и воодушевляющие результаты работы алгоритма относительно *классического* матричного аналога
- Подготовлена статья для конференции **2020 ACM SIGMOD**



## Дальнейшие задачи

- Исследование проблемы извлечения всех путей
- Детальное сравнение с классическим матричным алгоритмом
- Реализация алгоритма на GPGPU с использованием разреженных булевых матриц
- Модификация алгоритма для распределенного вычисления
- Интеграция алгоритма с графовой базой данных (RedisGraph)

- Семен Григорьев:
  - ▶ [s.v.grigoriev@spbu.ru](mailto:s.v.grigoriev@spbu.ru)
  - ▶ [Semen.Grigorev@jetbrains.com](mailto:Semen.Grigorev@jetbrains.com)
- Рустам Азимов:
  - ▶ [rustam.azimov19021995@gmail.com](mailto:rustam.azimov19021995@gmail.com)
  - ▶ [Rustam.Azimov@jetbrains.com](mailto:Rustam.Azimov@jetbrains.com)
- Екатерина Шеметова: [katyacyfra@gmail.com](mailto:katyacyfra@gmail.com)
- Егор Орачев: [egor.orachev@gmail.com](mailto:egor.orachev@gmail.com)
- Илья Эпельбаум: [iliyepelbaun@gmail.com](mailto:iliyepelbaun@gmail.com)
  
- Датасет: [https://github.com/JetBrains-Research/CFPQ\\_Data](https://github.com/JetBrains-Research/CFPQ_Data)
- Реализация: <https://github.com/YaccConstructor/RedisGraph>