

Анализ графов с использованием формальных языков в качестве ограничений на пути

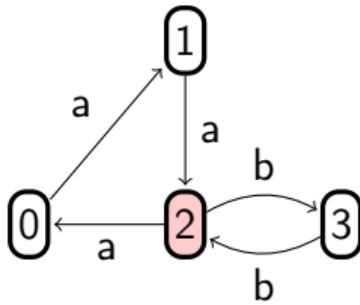
Заготовки для докторской диссертации

Семён Григорьев

Санкт-Петербургский Государственный Университет

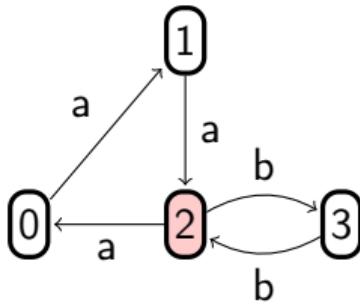
!!! !!! 2026

Формальные языки и ограничения на пути в графе



- $G = \langle V, E, L \rangle$ — (ориентированный) граф с метками на рёбрах
- Путь π задаёт слово: $\omega(2\pi_1) = \omega(2 \xrightarrow{a} 0 \xrightarrow{a} 1) = aa$
- Ищем пути, задающие слова определённого вида
 - ▶ Например, слова вида a^*b
- Множество слов — язык \mathcal{L} над алфавитом Σ
 - ▶ Для удобства будем считать, что $\Sigma \cap L \neq \emptyset$

Формальные языки и ограничения на пути в графе

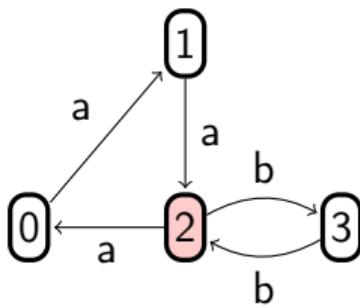


- $G = \langle V, E, L \rangle$ — (ориентированный) граф с метками на рёбрах
- Путь π задаёт слово: $\omega(2\pi_1) = \omega(2 \xrightarrow{a} 0 \xrightarrow{a} 1) = aa$
- Ищем пути, задающие слова определённого вида
 - ▶ Например, слова вида a^*b
- Множество слов — язык \mathcal{L} над алфавитом Σ
 - ▶ Для удобства будем считать, что $\Sigma \cap L \neq \emptyset$

Варианты задач Formal Language Constrained Path Quering (FLPQ)

Для данного графа G , стартовых вершин $V_s \in V$ и финальных вершин $V_f \in V$, языка \mathcal{L}

Формальные языки и ограничения на пути в графе



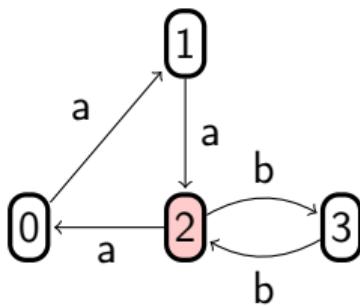
- $G = \langle V, E, L \rangle$ — (ориентированный) граф с метками на рёбрах
- Путь π задаёт слово: $\omega(2\pi_1) = \omega(2 \xrightarrow{a} 0 \xrightarrow{a} 1) = aa$
- Ищем пути, задающие слова определённого вида
 - ▶ Например, слова вида a^*b
- Множество слов — язык \mathcal{L} над алфавитом Σ
 - ▶ Для удобства будем считать, что $\Sigma \cap L \neq \emptyset$

Варианты задач Formal Language Constrained Path Quering (FLPQ)

Для данного графа G , стартовых вершин $V_s \in V$ и финальных вершин $V_f \in V$, языка \mathcal{L}

- Задача достижимости: $R = \{(u, v) \mid \omega(u\pi_v) \in \mathcal{L}, u \in V_s, v \in V_f\}$

Формальные языки и ограничения на пути в графе



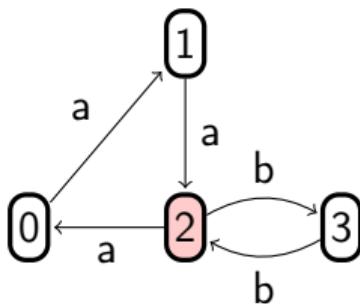
- $G = \langle V, E, L \rangle$ — (ориентированный) граф с метками на рёбрах
- Путь π задаёт слово: $\omega(\pi_1) = \omega(2 \xrightarrow{a} 0 \xrightarrow{a} 1) = aa$
- Ищем пути, задающие слова определённого вида
 - ▶ Например, слова вида a^*b
- Множество слов — язык \mathcal{L} над алфавитом Σ
 - ▶ Для удобства будем считать, что $\Sigma \cap L \neq \emptyset$

Варианты задач Formal Language Constrained Path Quering (FLPQ)

Для данного графа G , стартовых вершин $V_s \in V$ и финальных вершин $V_f \in V$, языка \mathcal{L}

- Задача достижимости: $R = \{(u, v) \mid \omega(u\pi_v) \in \mathcal{L}, u \in V_s, v \in V_f\}$
- Задача поиска всех путей: $P = \{\pi \mid \omega(u\pi_v) \in \mathcal{L}, u \in V_s, v \in V_f\}$

Формальные языки и ограничения на пути в графе



- $G = \langle V, E, L \rangle$ — (ориентированный) граф с метками на рёбрах
- Путь π задаёт слово: $\omega(\pi_1) = \omega(2 \xrightarrow{a} 0 \xrightarrow{a} 1) = aa$
- Ищем пути, задающие слова определённого вида
 - ▶ Например, слова вида a^*b
- Множество слов — язык \mathcal{L} над алфавитом Σ
 - ▶ Для удобства будем считать, что $\Sigma \cap L \neq \emptyset$

Варианты задач Formal Language Constrained Path Quering (FLPQ)

Для данного графа G , стартовых вершин $V_s \in V$ и финальных вершин $V_f \in V$, языка \mathcal{L}

- Задача достижимости: $R = \{(u, v) \mid \omega(u\pi_v) \in \mathcal{L}, u \in V_s, v \in V_f\}$
- Задача поиска всех путей: $P = \{\pi \mid \omega(u\pi_v) \in \mathcal{L}, u \in V_s, v \in V_f\}$
- Задача поиска одного пути
- Задача проверки наличия достижимых пар

Теория графов

- Классические варианты известных задач
 - ▶ Достигимость
 - ▶ Поиск путей
 - ▶ Между всем парами вершин
 - ▶ От заданных вершин
 - ▶ ...
- Способы представления графов
- Разновидности графов:
 - ▶ С циклами
 - ▶ Без циклов
 - ▶ ...

Задача на стыке областей

Теория графов

- Классические варианты известных задач
 - ▶ Достигимость
 - ▶ Поиск путей
 - ▶ Между всем парами вершин
 - ▶ От заданных вершин
 - ▶ ...
- Способы представления графов
- Разновидности графов:
 - ▶ С циклами
 - ▶ Без циклов
 - ▶ ...

Теория формальных языков

- Способы задания ограничений
 - ▶ Автоматы
 - ▶ Грамматики
 - ▶ ...
- Разрешимость
 - ▶ Проверка пустоты пересечения языков
 - ▶ Замкнутость относительно пересечения с регулярными языками
 - ▶ ...
- Представление результата
 - ▶ Пересечение языков — язык
 - ▶ Способы задания языков
 - ▶ ...

Области применения

- Регулярные языки
 - ▶ Графовые базы данных (Regular Path Queries, RPQ)
 - ★ Поддерживается (частично) в ряде существующих языков запросов (типа Cypher)

Области применения

- Регулярные языки
 - ▶ Графовые базы данных (Regular Path Queries, RPQ)
 - ★ Поддерживается (частично) в ряде существующих языков запросов (типа Cypher)
- Контекстно-свободные языки
 - ▶ Статический анализ кода (Context-Free Language Reachability, CFL-r)
 - ★ Анализ потока данных (межпроцедурный анализ указателей), разрешение имён, унификация, ...
 - ▶ Графовые базы данных (Context-Free Path Querying, CFPQ)
 - ★ Биоинформатика, анализ происхождения данных, анализ иерархий (в RDF), ...

Области применения

- Регулярные языки
 - ▶ Графовые базы данных (Regular Path Queries, RPQ)
 - ★ Поддерживается (частично) в ряде существующих языков запросов (типа Cypher)
- Контекстно-свободные языки
 - ▶ Статический анализ кода (Context-Free Language Reachability, CFL-r)
 - ★ Анализ потока данных (межпроцедурный анализ указателей), разрешение имён, унификация, ...
 - ▶ Графовые базы данных (Context-Free Path Querying, CFPQ)
 - ★ Биоинформатика, анализ происхождения данных, анализ иерархий (в RDF), ...
- За пределами контекстно-свободных языков
 - ▶ Многокомпонентные контекстно-свободные (MCFL): статический анализ кода
 - ▶ Линейные конъюнктивные (Linear Conjunctive): статический анализ кода

Запросы с ограничениями в виде регулярных языков

- Хорошо изучены
 - ▶ Впервые введены в работе A. O. Mendelzon and P. T. Wood. 1989. Finding regular simple paths in graph databases.
 - ▶ Большое количество прикладных и теоретических работ

Запросы с ограничениями в виде регулярных языков

- Хорошо изучены
 - ▶ Впервые введены в работе A. O. Mendelzon and P. T. Wood. 1989. Finding regular simple paths in graph databases.
 - ▶ Большое количество прикладных и теоретических работ
- Часть стандартов ISO языков запросов
 - ▶ ISO/IEC 9075-16:2023 Information technology — Database languages — SQL Part 16: **Property Graph Queries (SQL/PGQ)**
 - ▶ ISO/IEC 39075:2024 Information technology — Database languages — **GQL**

Запросы с ограничениями в виде регулярных языков

- Хорошо изучены
 - ▶ Впервые введены в работе A. O. Mendelzon and P. T. Wood. 1989. Finding regular simple paths in graph databases.
 - ▶ Большое количество прикладных и теоретических работ
- Часть стандартов ISO языков запросов
 - ▶ ISO/IEC 9075-16:2023 Information technology — Database languages — SQL Part 16: **Property Graph Queries (SQL/PGQ)**
 - ▶ ISO/IEC 39075:2024 Information technology — Database languages — **GQL**
- ! Проблемы с производительностью
 - ▶ Angela Bonifati, George Fletcher, Hannes Voigt, and Nikolay Yakovets. 2018. Querying Graphs
 - ▶ Diego Arroyuelo, et al. 2023. Optimizing RPQs over a compact graph representation

Запросы с ограничениями в виде контексто-свободных языков

- Активно изучаются
 - ▶ Родоначальники: Томас Репс и Михалис Яннакакис
 - ★ Thomas Reps. 1997. Program analysis via graph reachability
 - ★ Mihalis Yannakakis. 1990. Graph-theoretic methods in database theory
 - ▶ Большое количество прикладных и теоретических работ
 - ▶ Известные открытые вопросы
 - ★ Существование истинно субкубического алгоритма для задачи достижимости

Запросы с ограничениями в виде контексто-свободных языков

- Активно изучаются
 - ▶ Родоначальники: Томас Репс и Михалис Яннакакис
 - ★ Thomas Reps. 1997. Program analysis via graph reachability
 - ★ Mihalis Yannakakis. 1990. Graph-theoretic methods in database theory
 - ▶ Большое количество прикладных и теоретических работ
 - ▶ Известные открытые вопросы
 - ★ Существование истинно субкубического алгоритма для задачи достижимости
- Существуют инструменты (академические прототипы) анализа кода на основе CFL-р
 - ▶ POCR, PEARL, GASPAK, GIGASCALE, ...

Запросы с ограничениями в виде контекстно-свободных языков

- Активно изучаются
 - ▶ Родоначальники: Томас Репс и Михалис Яннакакис
 - ★ Thomas Reps. 1997. Program analysis via graph reachability
 - ★ Mihalis Yannakakis. 1990. Graph-theoretic methods in database theory
 - ▶ Большое количество прикладных и теоретических работ
 - ▶ Известные открытые вопросы
 - ★ Существование истинно субкубического алгоритма для задачи достижимости
- Существуют инструменты (академические прототипы) анализа кода на основе CFL-р
 - ▶ POCR, PEARL, GASPAR, GIGASCALE, ...
- Попытки интегрировать в графовые базы данных
 - ▶ Предложение о расширении Cypher
 - ▶ Arseniy Terekhov et al. 2021. Multiple-Source Context-Free Path Querying in Terms of Linear Algebra

Запросы с ограничениями в виде контекстно-свободных языков

- Активно изучаются
 - ▶ Родоначальники: Томас Репс и Михалис Яннакакис
 - ★ Thomas Reps. 1997. Program analysis via graph reachability
 - ★ Mihalis Yannakakis. 1990. Graph-theoretic methods in database theory
 - ▶ Большое количество прикладных и теоретических работ
 - ▶ Известные открытые вопросы
 - ★ Существование истинно субкубического алгоритма для задачи достижимости
 - Существуют инструменты (академические прототипы) анализа кода на основе CFL-р
 - ▶ POCR, PEARL, GASPAR, GIGASCALE, ...
 - Попытки интегрировать в графовые базы данных
 - ▶ Предложение о расширении Cypher
 - ▶ Arseniy Terekhov et al. 2021. Multiple-Source Context-Free Path Querying in Terms of Linear Algebra
- ! Проблемы с производительностью
- ▶ Jochem Kuijpers, George Fletcher, Nikolay Yakovets, and Tobias Lindaaker. 2019. An Experimental Study of Context-Free Path Query Evaluation Methods
 - ▶ Chenghang Shi, et al. 2024. Pearl: A Multi-Derivation Approach to Efficient CFL-Reachability Solving

Более широкие классы языков в качестве ограничений на пути

- Перспективны
 - ▶ Позволяют улучшить «точность» некоторых видов анализа кода

Более широкие классы языков в качестве ограничений на пути

- Перспективны
 - ▶ Позволяют улучшить «точность» некоторых видов анализа кода

! Изучены слабо

- ▶ Qirun Zhang and Zhendong Su. 2017. Context-sensitive data-dependence analysis via linear conjunctive language reachability
- ▶ Giovanna Kobus Conrado, Adam Husted Kjelstrøm, Jaco van de Pol, and Andreas Pavlogiannis. 2025. Program Analysis via Multiple Context Free Language Reachability
- ▶ Epelbaum I.V., Azimov R.Sh., Grigorev S.V. 2023. Multiple context-free path querying by matrix multiplication

Высокопроизводительный анализ графов

- Обобщённая разреженная линейная алгебра — один из перспективных путей к высокопроизводительному анализу графов
 - ▶ «Естественная» параллельность — возможность использовать современное аппаратное обеспечение, в том числе GPGPU
 - ▶ Выразительность — алгоритмы для достаточно широкий спектра задач выражимы в терминах линейной алгебры
 - ★ Транзитивное замыкание, поиск кратчайших путей
 - ★ Обход в ширину, построение минимального остовного дерева
 - ★ Подсчёт треугольников, ...

Высокопроизводительный анализ графов

- Обобщённая разреженная линейная алгебра — один из перспективных путей к высокопроизводительному анализу графов
 - ▶ «Естественная» параллельность — возможность использовать современное аппаратное обеспечение, в том числе GPGPU
 - ▶ Выразительность — алгоритмы для достаточно широкий спектра задач выражимы в терминах линейной алгебры
 - ★ Транзитивное замыкание, поиск кратчайших путей
 - ★ Обход в ширину, построение минимального остовного дерева
 - ★ Подсчёт треугольников, ...
- GraphBLAS API — стандарт, декларирующий базовые примитивы линейной алгебры и операции над ними
 - ▶ Предложена качественная референсная реализация: SuiteSparse:GraphBLAS
 - ▶ Разрабатывается библиотеки прикладных алгоритмов LAGraph

Высокопроизводительный анализ графов

- Обобщённая разреженная линейная алгебра — один из перспективных путей к высокопроизводительному анализу графов
 - ▶ «Естественная» параллельность — возможность использовать современное аппаратное обеспечение, в том числе GPGPU
 - ▶ Выразительность — алгоритмы для достаточно широкий спектра задач выражимы в терминах линейной алгебры
 - ★ Транзитивное замыкание, поиск кратчайших путей
 - ★ Обход в ширину, построение минимального остовного дерева
 - ★ Подсчёт треугольников, ...
 - GraphBLAS API — стандарт, декларирующий базовые примитивы линейной алгебры и операции над ними
 - ▶ Предложена качественная референсная реализация: SuiteSparse:GraphBLAS
 - ▶ Разрабатывается библиотеки прикладных алгоритмов LAGraph
- ! Не все алгоритмы естественным образом выражимы в терминах линейной алгебры
- ! Эффективное использование GPGPU — нетривиальная задача

Обучение

- Существуют курсы, затрагивающие те или иные аспекты решения задач анализа графов с использованием формальных языков
 - ▶ Анализ данных, базы данных
 - ▶ Статический анализ кода

Обучение

- Существуют курсы, затрагивающие те или иные аспекты решения задач анализа графов с использованием формальных языков
 - ▶ Анализ данных, базы данных
 - ▶ Статический анализ кода
- Курсы затрагивают отдельные аспекты
 - ▶ Фундаментальные: алгебра, теория графов, ...
 - ▶ Прикладные: анализ данных, базы данных, построение компиляторов, ...

Обучение

- Существуют курсы, затрагивающие те или иные аспекты решения задач анализа графов с использованием формальных языков
 - ▶ Анализ данных, базы данных
 - ▶ Статический анализ кода
- Курсы затрагивают отдельные аспекты
 - ▶ Фундаментальные: алгебра, теория графов, ...
 - ▶ Прикладные: анализ данных, базы данных, построение компиляторов, ...

! Целостность картины

- ▶ Разные прикладные области используют общий фундамент

Обучение

- Существуют курсы, затрагивающие те или иные аспекты решения задач анализа графов с использованием формальных языков
 - ▶ Анализ данных, базы данных
 - ▶ Статический анализ кода
- Курсы затрагивают отдельные аспекты
 - ▶ Фундаментальные: алгебра, теория графов, ...
 - ▶ Прикладные: анализ данных, базы данных, построение компиляторов, ...

! Целостность картины

- ▶ Разные прикладные области используют общий фундамент

! Преподавание фундаментальных дисциплин для инженеров-программистов

- ▶ Связь теории и практики
- ▶ Применимость в конкретных областях
 - ★ Теория формальных языков — синтаксический анализ (языков программирования)
 - ★ А что, если я не собираюсь заниматься компиляторами?

Выводы

- Анализ графов с использованием формальных языков в качестве ограничений на пути — активно развивающаяся область, имеющая прикладное значение

Выводы

- Анализ графов с использованием формальных языков в качестве ограничений на пути — активно развивающаяся область, имеющая прикладное значение
- Междисциплинарность: теория графов, теория формальных языков,
линейная алгебра, статический анализ кода, графовые базы данных,
параллельные вычисления

Выводы

- Анализ графов с использованием формальных языков в качестве ограничений на пути — активно развивающаяся область, имеющая прикладное значение
- Междисциплинарность: теория графов, теория формальных языков, линейная алгебра, статический анализ кода, графовые базы данных, параллельные вычисления
- ! Отсутствие системности в исследованиях
 - ▶ Выстраивание междисциплинарных связей
 - ▶ Переиспользование методов, инструментов, терминологии, . . .
 - ▶ Сопоставление результатов

Выводы

- Анализ графов с использованием формальных языков в качестве ограничений на пути — активно развивающаяся область, имеющая прикладное значение
- Междисциплинарность: теория графов, теория формальных языков, линейная алгебра, статический анализ кода, графовые базы данных, параллельные вычисления
- ! Отсутствие системности в исследованиях
 - ▶ Выстраивание междисциплинарных связей
 - ▶ Переиспользование методов, инструментов, терминологии, . . .
 - ▶ Сопоставление результатов
- ! Проблемы с прикладными алгоритмами
 - ▶ Точечные решения вместо методов решения классов задач
 - ▶ Плохая производительность
 - ▶ Слабая база для экспериментальных исследований

Выводы

- Анализ графов с использованием формальных языков в качестве ограничений на пути — активно развивающаяся область, имеющая прикладное значение
- Междисциплинарность: теория графов, теория формальных языков, линейная алгебра, статический анализ кода, графовые базы данных, параллельные вычисления
- ! Отсутствие системности в исследованиях
 - ▶ Выстраивание междисциплинарных связей
 - ▶ Переиспользование методов, инструментов, терминологии, . . .
 - ▶ Сопоставление результатов
- ! Проблемы с прикладными алгоритмами
 - ▶ Точечные решения вместо методов решения классов задач
 - ▶ Плохая производительность
 - ▶ Слабая база для экспериментальных исследований
- ! Отсутствие системности в подготовке специалистов
 - ▶ Выстраивание междисциплинарных связей
 - ▶ Формирование целостной картины

Цель работы

Целью данной работы является создание методологии анализа графов с использованием формальных языков в качестве ограничений на пути

- Нацеленную на получение практически значимых результатов
- Предоставляющую гибкий подход к решению задач
- Позволяющую формально рассуждать о свойствах объектов (модели, алгоритмы)

Положения, выносимые на защиту

- 1 Создана методология анализа графов с использованием формальных языков в качестве ограничений на пути, нацеленная на конструирование алгоритмов FLPQ

Положения, выносимые на защиту

- ① Создана методология анализа графов с использованием формальных языков в качестве ограничений на пути, нацеленная на конструирование алгоритмов FLPQ
- ② Разработан методы конструирования алгоритмов решения FLPQ

Положения, выносимые на защиту

- ① Создана методология анализа графов с использованием формальных языков в качестве ограничений на пути, нацеленная на конструирование алгоритмов FLPQ
- ② Разработан методы конструирования алгоритмов решения FLPQ
 - ▶ Использующих операции линейной алгебры

Положения, выносимые на защиту

- ① Создана методология анализа графов с использованием формальных языков в качестве ограничений на пути, нацеленная на конструирование алгоритмов FLPQ
- ② Разработан методы конструирования алгоритмов решения FLPQ
 - ▶ Использующих операции линейной алгебры
 - ▶ Основанных на классических алгоритмах синтаксического анализа

Положения, выносимые на защиту

- ① Создана методология анализа графов с использованием формальных языков в качестве ограничений на пути, нацеленная на конструирование алгоритмов FLPQ
- ② Разработан методы конструирования алгоритмов решения FLPQ
 - ▶ Использующих операции линейной алгебры
 - ▶ Основанных на классических алгоритмах синтаксического анализа
- ③ Разработаны методические рекомендации по проведению экспериментальных исследований решений FLPQ

Положения, выносимые на защиту

- ① Создана методология анализа графов с использованием формальных языков в качестве ограничений на пути, нацеленная на конструирование алгоритмов FLPQ
- ② Разработан методы конструирования алгоритмов решения FLPQ
 - ▶ Использующих операции линейной алгебры
 - ▶ Основанных на классических алгоритмах синтаксического анализа
- ③ Разработаны методические рекомендации по проведению экспериментальных исследований решений FLPQ
- ④ Спроектированы и реализованы библиотеки разреженной линейной алгебры, использующие графические ускорители для решения FLPQ

Положения, выносимые на защиту

- ① Создана методология анализа графов с использованием формальных языков в качестве ограничений на пути, нацеленная на конструирование алгоритмов FLPQ
- ② Разработан методы конструирования алгоритмов решения FLPQ
 - ▶ Использующих операции линейной алгебры
 - ▶ Основанных на классических алгоритмах синтаксического анализа
- ③ Разработаны методические рекомендации по проведению экспериментальных исследований решений FLPQ
- ④ Спроектированы и реализованы библиотеки разреженной линейной алгебры, использующие графические ускорители для решения FLPQ
- ⑤ Создан набор данных для экспериментального исследования решений FLPQ

Положения, выносимые на защиту

- ① Создана методология анализа графов с использованием формальных языков в качестве ограничений на пути, нацеленная на конструирование алгоритмов FLPQ
- ② Разработан методы конструирования алгоритмов решения FLPQ
 - ▶ Использующих операции линейной алгебры
 - ▶ Основанных на классических алгоритмах синтаксического анализа
- ③ Разработаны методические рекомендации по проведению экспериментальных исследований решений FLPQ
- ④ Спроектированы и реализованы библиотеки разреженной линейной алгебры, использующие графические ускорители для решения FLPQ
- ⑤ Создан набор данных для экспериментального исследования решений FLPQ
- ⑥ Разработаны, реализованы, интегрированы в пользовательские библиотеки и инструменты алгоритмы решения различных вариантов FLPQ

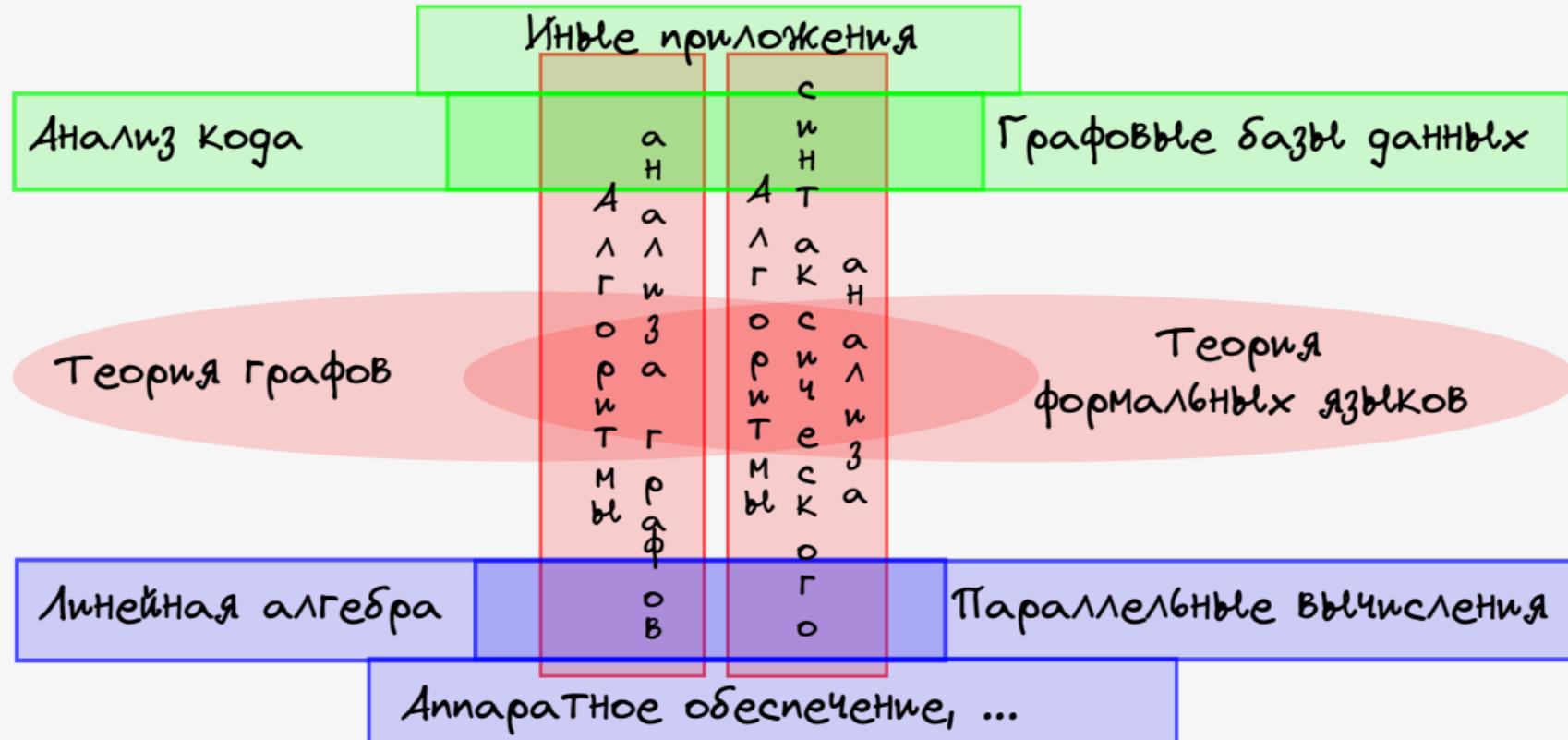
Положения, выносимые на защиту

- ① Создана методология анализа графов с использованием формальных языков в качестве ограничений на пути, нацеленная на конструирование алгоритмов FLPQ
- ② Разработан методы конструирования алгоритмов решения FLPQ
 - ▶ Использующих операции линейной алгебры
 - ▶ Основанных на классических алгоритмах синтаксического анализа
- ③ Разработаны методические рекомендации по проведению экспериментальных исследований решений FLPQ
- ④ Спроектированы и реализованы библиотеки разреженной линейной алгебры, использующие графические ускорители для решения FLPQ
- ⑤ Создан набор данных для экспериментального исследования решений FLPQ
- ⑥ Разработаны, реализованы, интегрированы в пользовательские библиотеки и инструменты алгоритмы решения различных вариантов FLPQ
- ⑦ Разработаны методические рекомендации по выстраиванию междисциплинарных связей, улучшающих понимание областей применения и алгоритмов решения FLPQ

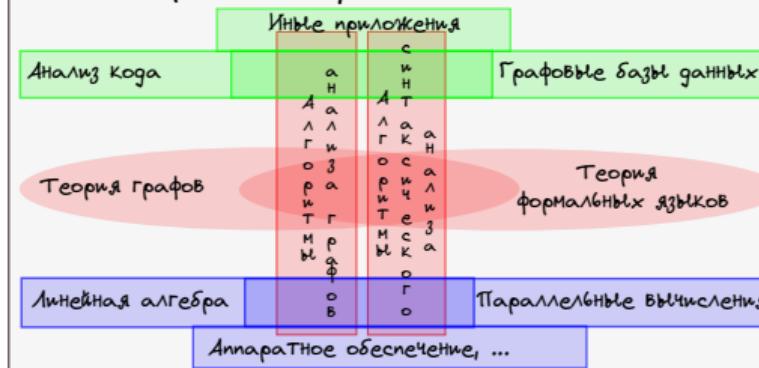
Положения, выносимые на защиту

- ① Создана методология анализа графов с использованием формальных языков в качестве ограничений на пути, нацеленная на конструирование алгоритмов FLPQ
- ② Разработан методы конструирования алгоритмов решения FLPQ
 - ▶ Использующих операции линейной алгебры
 - ▶ Основанных на классических алгоритмах синтаксического анализа
- ③ Разработаны методические рекомендации по проведению экспериментальных исследований решений FLPQ
- ④ Спроектированы и реализованы библиотеки разреженной линейной алгебры, использующие графические ускорители для решения FLPQ
- ⑤ Создан набор данных для экспериментального исследования решений FLPQ
- ⑥ Разработаны, реализованы, интегрированы в пользовательские библиотеки и инструменты алгоритмы решения различных вариантов FLPQ
- ⑦ Разработаны методические рекомендации по выстраиванию междисциплинарных связей, улучшающих понимание областей применения и алгоритмов решения FLPQ
- ⑧ Разработан курс для программных инженеров, выстраивающий изучение основ теории формальных языков вокруг различных вариантов FLPQ

Методология анализа графов с использованием формальных языков в качестве ограничений на пути

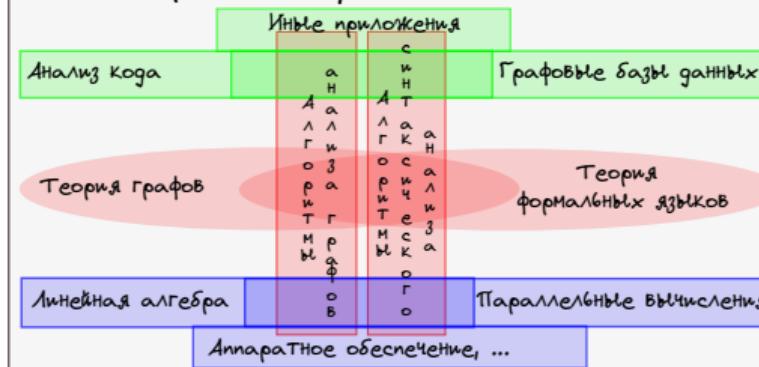


Методология анализа графов с использованием формальных языков в качестве огра ничений на пути



- Объект — задачи анализа графов с использованием формальных языков в качестве ограничений на пути
 - Предмет — алгоритмы решения задач анализа графов с использованием формальных языков в качестве ограничений на пути

Методология анализа графов с использованием формальных языков в качестве ограничений на пути



- Объект — задачи анализа графов с использованием формальных языков в качестве ограничений на пути
- Предмет — алгоритмы решения задач анализа графов с использованием формальных языков в качестве ограничений на пути

- Теоретическая обоснованность: границы применимости, оценки сложности, доказательство корректности
- Универсальность: возможность построить решатели в стиле SAT или SMT
- Настраиваемость: класс языков \longleftrightarrow структура графа \longleftrightarrow библиотеки
- Эволюционность: использование новых результатов из конкретных областей
- Основные фазы (цикл)
 - ▶ Формулировка задачи, моделирование предметной области
 - ▶ Дизайн: построение алгоритма, теоретический анализ
 - ▶ Разработка: реализация алгоритма, оптимизации
 - ▶ Экспериментальное исследование: оценка показателей, сравнительный анализ

Методология анализа графов
с использованием формальных языков
в качестве ограничений на пути

Методология анализа графов с использованием формальных языков в качестве ограничений на пути

Методы создания алгоритмов

- Для различных классов языков
- С различными свойствами
- ...

Алгоритмы

- Формальные свойства: корректность, сложность
- Базовые примитивы: матрицы, конечные автоматы, ...
- ...

Библиотеки

- Производительность
- Перенесование компонент
- ...

Методология анализа графов с использованием формальных языков в качестве ограничений на пути

Методы создания алгоритмов

- Для различных классов языков
- С различными свойствами
- ...

Алгоритмы

- Формальные свойства: корректность, сложность
- Базовые примитивы: матрицы, конечные автоматы, ...
- ...

Библиотеки

- Производительность
- Перенесение компонент
- ...

Методические рекомендации по проведению
экспериментального исследования разработанных решений

- Для различных типов задач
- Для различных прикладных областей
- ...

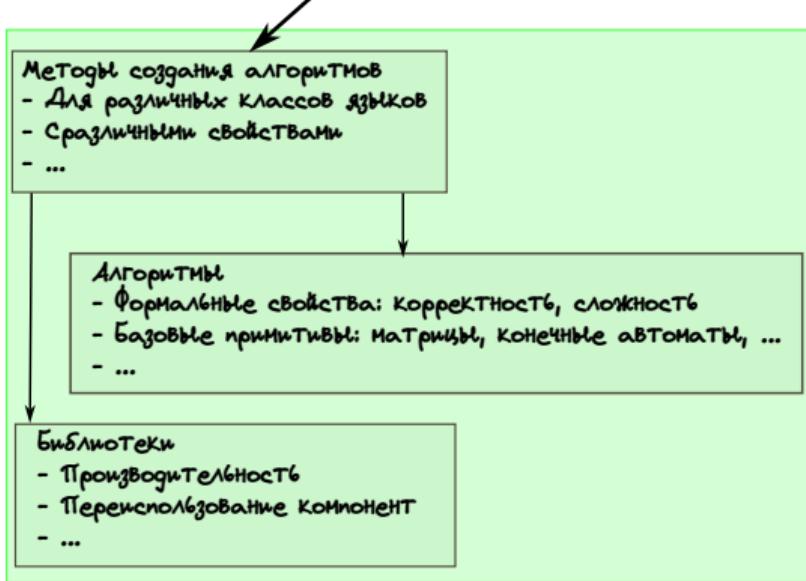
Наборы данных

- Графы
- Ограничения
- Метаданные
- ...

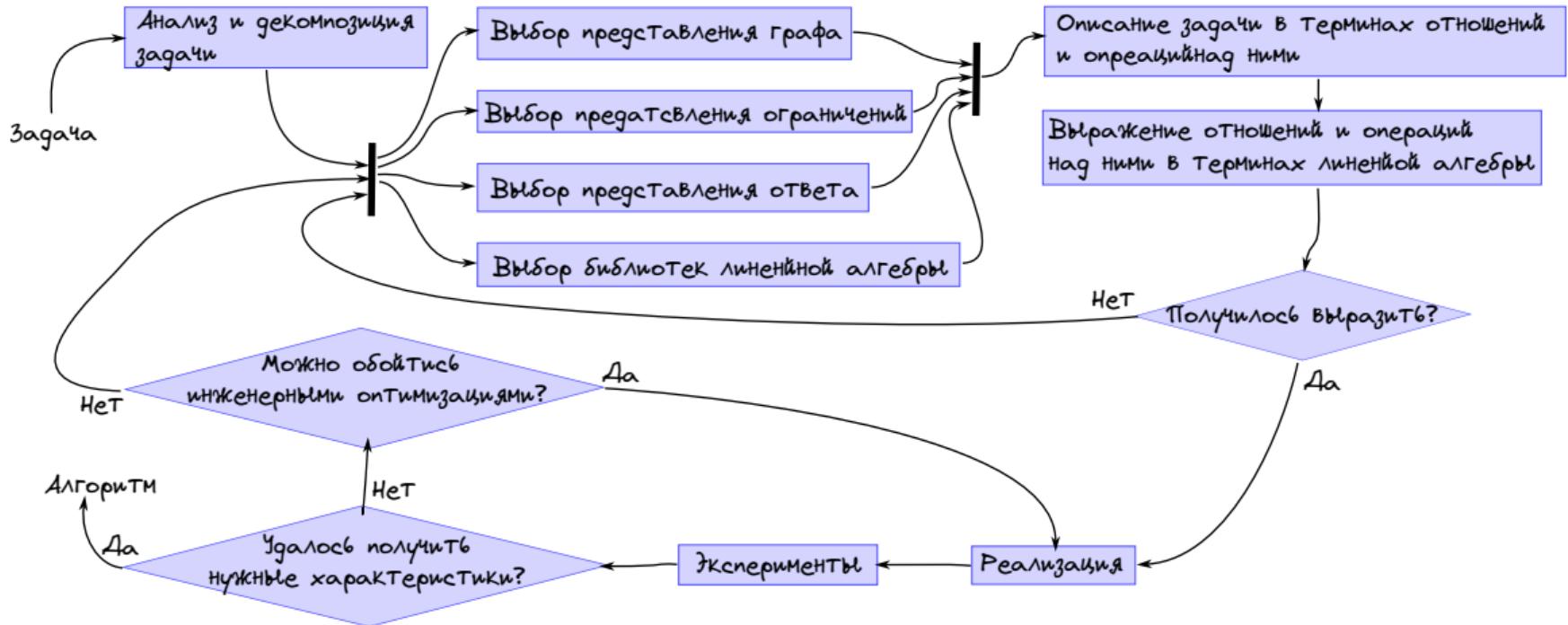
Вспомогательные инструменты

- Матричные с операциями
- Проверка корректности результатов
- ...

Методология анализа графов с использованием формальных языков в качестве ограничений на пути



Метод конструирования алгоритмов FLPQ, основанных на линейной алгебре



Алгоритмы FLPQ, основанные на линейной алгебре

- Новые алгоритмы анализа графов с использованием следующих классов языков
 - ▶ Регулярные языки (RPQ)
 - ▶ Контекстно-свободные языки (CFPQ)
 - ▶ Булевы и конъюнктивные языки
 - ▶ Многокомпонентные контекстно-свободные языки

Алгоритмы FLPQ, основанные на линейной алгебре

- Новые алгоритмы анализа графов с использованием следующих классов языков
 - ▶ Регулярные языки (RPQ)
 - ▶ Контекстно-свободные языки (CFPQ)
 - ▶ Булевы и конъюнктивные языки
 - ▶ Многокомпонентные контекстно-свободные языки
- Решаются различные варианты задач FLPQ
 - ▶ Достигимость
 - ▶ Поиск путей
 - ▶ Между всеми парами вершин
 - ▶ От заданного множества вершин

Алгоритмы FLPQ, основанные на линейной алгебре

- Новые алгоритмы анализа графов с использованием следующих классов языков
 - ▶ Регулярные языки (RPQ)
 - ▶ Контекстно-свободные языки (CFPQ)
 - ▶ Булевы и конъюнктивные языки
 - ▶ Многокомпонентные контекстно-свободные языки
- Решаются различные варианты задач FLPQ
 - ▶ Достигимость
 - ▶ Поиск путей
 - ▶ Между всеми парами вершин
 - ▶ От заданного множества вершин
- Используются разнообразные операции линейной алгебры
 - ▶ Поэлементные операции над матрицами и векторами
 - ▶ Умножение матриц и матрицы на вектор в различных полукольцах
 - ▶ Произведение Кронекера

Алгоритмы FLPQ, основанные на линейной алгебре

- Новые алгоритмы анализа графов с использованием следующих классов языков
 - ▶ Регулярные языки (RPQ)
 - ▶ Контекстно-свободные языки (CFPQ)
 - ▶ Булевы и конъюнктивные языки
 - ▶ Многокомпонентные контекстно-свободные языки
- Решаются различные варианты задач FLPQ
 - ▶ Достигимость
 - ▶ Поиск путей
 - ▶ Между всеми парами вершин
 - ▶ От заданного множества вершин
- Используются разнообразные операции линейной алгебры
 - ▶ Поэлементные операции над матрицами и векторами
 - ▶ Умножение матриц и матрицы на вектор в различных полукольцах
 - ▶ Произведение Кронекера
- Сформулированы и доказаны теоремы о корректности полученных алгоритмов
- Получены оценки сложности для предложенных алгоритмов

Метод конструирования алгоритмов FLPQ, основанных на классических алгоритмах синтаксического анализа

- ① Обобщённый синтаксический анализ
- ② Понятие shift-shift конфликта

Алгоритмы CFPQ, основанные на классических алгоритмах синтаксического анализа

- Новые алгоритмы анализа графов с использованием контекстно-свободных языков

Алгоритмы CFPQ, основанные на классических алгоритмах синтаксического анализа

- Новые алгоритмы анализа графов с использованием контекстно-свободных языков
- Используются различные методы и алгоритмы синтаксического анализа
 - ▶ Генеративный на основе обобщённого LR (Generalized LR, GLR)
 - ▶ Генеративный на основе обобщённого LL (Generalized LL, GLL)
 - ▶ Комбинаторы парсеров (Parser combinators)
 - ▶ Комбинаторы грамматик (Grammar combinators)

Алгоритмы CFPQ, основанные на классических алгоритмах синтаксического анализа

- Новые алгоритмы анализа графов с использованием контекстно-свободных языков
- Используются различные методы и алгоритмы синтаксического анализа
 - ▶ Генеративный на основе обобщённого LR (Generalized LR, GLR)
 - ▶ Генеративный на основе обобщённого LL (Generalized LL, GLL)
 - ▶ Комбинаторы парсеров (Parser combinator)
 - ▶ Комбинаторы грамматик (Grammar combinator)
- Решаются различные варианты задач FLPQ
 - ▶ Достигимость
 - ▶ Поиск путей
 - ▶ Между всеми парами вершин
 - ▶ От заданного множества вершин

Алгоритмы CFPQ, основанные на классических алгоритмах синтаксического анализа

- Новые алгоритмы анализа графов с использованием контекстно-свободных языков
- Используются различные методы и алгоритмы синтаксического анализа
 - ▶ Генеративный на основе обобщённого LR (Generalized LR, GLR)
 - ▶ Генеративный на основе обобщённого LL (Generalized LL, GLL)
 - ▶ Комбинаторы парсеров (Parser combinator)
 - ▶ Комбинаторы грамматик (Grammar combinator)
- Решаются различные варианты задач FLPQ
 - ▶ Достигимость
 - ▶ Поиск путей
 - ▶ Между всеми парами вершин
 - ▶ От заданного множества вершин
- Сформулированы и доказаны теоремы о корректности полученных алгоритмов
- Получены оценки сложности для предложенных алгоритмов

Методические рекомендации по проведению экспериментальных исследований решений FLPQ

- Выбор и подготовка входных данных
 - ▶ Данные из различных прикладных областей (особенно для универсальных решений)
 - ▶ Форма представления запроса существенно влияет на производительность решения
 - ▶ Правила выбора стартовых вершин зависят от предметной области
 - ★ Крайне редко стоит выбирать случайные вершины

Методические рекомендации по проведению экспериментальных исследований решений FLPQ

- Выбор и подготовка входных данных
 - ▶ Данные из различных прикладных областей (особенно для универсальных решений)
 - ▶ Форма представления запроса существенно влияет на производительность решения
 - ▶ Правила выбора стартовых вершин зависят от предметной области
 - ★ Крайне редко стоит выбирать случайные вершины
- Какие шаги включать в замеры
 - ▶ Предподготовка данных: существуют «неотчуждаемые» преобразования
 - ▶ Постобработка результата
 - ★ Достижимость: матрица смежности → множество пар вершин
 - ★ Поиск путей: «индекс» путей → отдельные пути (один, несколько, все)
 - ▶ Построение индексов
 - ★ Сохраняем результат для всех вершин, **быстро** решаем задачу для конкретных вершин

Методические рекомендации по проведению экспериментальных исследований решений FLPQ

- Выбор и подготовка входных данных
 - ▶ Данные из различных прикладных областей (особенно для универсальных решений)
 - ▶ Форма представления запроса существенно влияет на производительность решения
 - ▶ Правила выбора стартовых вершин зависят от предметной области
 - ★ Крайне редко стоит выбирать случайные вершины
- Какие шаги включать в замеры
 - ▶ Предподготовка данных: существуют «неотчуждаемые» преобразования
 - ▶ Постобработка результата
 - ★ Достижимость: матрица смежности → множество пар вершин
 - ★ Поиск путей: «индекс» путей → отдельные пути (один, несколько, все)
 - ▶ Построение индексов
 - ★ Сохраняем результат для всех вершин, **быстро** решаем задачу для конкретных вершин
- Разрешимость проверки эквивалентности результатов для задачи поиска всех путей
 - ▶ Проверка эквивалентности конечных автоматов в общем случае **разрешима**
 - ▶ Проверка эквивалентности КС грамматик в общем случае **неразрешима**
 - ▶ Проверка эквивалентности MCFG в общем случае **неразрешима**

Набор данных CFPQ_Data

- Для экспериментального исследования алгоритмов FLPQ
 - ▶ Регулярные языки
 - ▶ Контекстно-свободные языки
 - ⚙️ Многокомпонентные контекстно-свободные языки

Набор данных CFPQ_Data

- Для экспериментального исследования алгоритмов FLPQ
 - ▶ Регулярные языки
 - ▶ Контекстно-свободные языки
 -  Многокомпонентные контекстно-свободные языки
- Синтетические данные
 - ▶ Крайние случаи
 - ▶ «Случайные» графы
- Реальные данные из различных областей (граф, запросы, стартовые вершины)
 - ▶ Статический анализ кода
 - ▶ Семантические сети
 -  Биоинформатика
 -  Анализ происхождения данных

Набор данных CFPQ_Data

- Для экспериментального исследования алгоритмов FLPQ
 - ▶ Регулярные языки
 - ▶ Контекстно-свободные языки
 - ⚙️ Многокомпонентные контекстно-свободные языки
- Синтетические данные
 - ▶ Крайние случаи
 - ▶ «Случайные» графы
- Реальные данные из различных областей (граф, запросы, стартовые вершины)
 - ▶ Статический анализ кода
 - ▶ Семантические сети
 - ⚙️ Биоинформатика
 - ⚙️ Анализ происхождения данных
- Метаданные
 - ▶ Описание: источник, количество вершин, рёбер, метки рёбер
 - ▶ Контрольные значения: ответы на конкретные запросы

Инженерные решения на основе разработанных алгоритмов

- Разработаны универсальные решатели
 - ▶ UCFS и CFPQ_GLL: контекстно-свободные языки, на основе GLL
 - ▶ CFPQ_PyAlgo: контекстно-свободные языки, на основе линейной алгебры

¹Коллекция алгоритмов анализа графов на основе линейной алгебры

Инженерные решения на основе разработанных алгоритмов

- Разработаны универсальные решатели
 - ▶ UCFS и CFPQ_GLL: контекстно-свободные языки, на основе GLL
 - ▶ CFPQ_PyAlgo: контекстно-свободные языки, на основе линейной алгебры
- Интеграция с инструментами статического анализа кода
 - ⚙️ C SVF — инструмент для анализа LLVM IR
 - ⚙️ C Qilin — инструмент для анализа Java

¹Коллекция алгоритмов анализа графов на основе линейной алгебры

Инженерные решения на основе разработанных алгоритмов

- Разработаны универсальные решатели
 - ▶ UCFS и CFPQ_GLL: контекстно-свободные языки, на основе GLL
 - ▶ CFPQ_PyAlgo: контекстно-свободные языки, на основе линейной алгебры
- Интеграция с инструментами статического анализа кода
 - ⚙️ C SVF — инструмент для анализа LLVM IR
 - ⚙️ C Qilin — инструмент для анализа Java
- Интеграция в графовые базы данных
 - ▶ Первая полноценная поддержка запросов с ограничениями в виде КС языков
 - ★ От языка запросов (Cypher) до алгоритма выполнения в инфраструктуре БД (RedisGraph)
 - ▶ Самый высокопроизводительный алгоритм запросов с КС ограничениями для Neo4j

¹Коллекция алгоритмов анализа графов на основе линейной алгебры

Инженерные решения на основе разработанных алгоритмов

- Разработаны универсальные решатели
 - ▶ UCFS и CFPQ_GLL: контекстно-свободные языки, на основе GLL
 - ▶ CFPQ_PyAlgo: контекстно-свободные языки, на основе линейной алгебры
- Интеграция с инструментами статического анализа кода
 - ⚙️ C SVF — инструмент для анализа LLVM IR
 - ⚙️ C Qilin — инструмент для анализа Java
- Интеграция в графовые базы данных
 - ▶ Первая полноценная поддержка запросов с ограничениями в виде КС языков
 - ★ От языка запросов (Cypher) до алгоритма выполнения в инфраструктуре БД (RedisGraph)
 - ▶ Самый высокопроизводительный алгоритм запросов с КС ограничениями для Neo4j
- Часть алгоритмов интегрирована в библиотеку LAGraph¹
 - ▶ Для регулярных и контекстно-свободных языков

¹Коллекция алгоритмов анализа графов на основе линейной алгебры

Инженерные решения на основе разработанных алгоритмов

- Разработаны универсальные решатели
 - ▶ UCFS и CFPQ_GLL: контекстно-свободные языки, на основе GLL
 - ▶ CFPQ_PyAlgo: контекстно-свободные языки, на основе линейной алгебры
- Интеграция с инструментами статического анализа кода
 - ⚙️ C SVF — инструмент для анализа LLVM IR
 - ⚙️ C Qilin — инструмент для анализа Java
- Интеграция в графовые базы данных
 - ▶ Первая полноценная поддержка запросов с ограничениями в виде КС языков
 - ★ От языка запросов (Cypher) до алгоритма выполнения в инфраструктуре БД (RedisGraph)
 - ▶ Самый высокопроизводительный алгоритм запросов с КС ограничениями для Neo4j
- Часть алгоритмов интегрирована в библиотеку LAGraph¹
 - ▶ Для регулярных и контекстно-свободных языков
- Отдельные реализации
 - ▶ Многокомпонентные контекстно-свободные языки в качестве ограничений
 - ▶ Конъюнктивные и булевы языки в качестве ограничений

¹Коллекция алгоритмов анализа графов на основе линейной алгебры

Библиотеки разреженной линейной алгебры

- Спроектированы и реализованы библиотеки разреженной линейной алгебры, использующие возможности GPGPU
 - ▶ Специализированные для булевой алгебры: CuBool (Nvidia Cuda), Spbla (OpenCL)
 - ▶ Обобщённые: Spla (OpenCL), Brahma.FSharp + GraphBLAS# (F# + OpenCL)

Библиотеки разреженной линейной алгебры

- Спроектированы и реализованы библиотеки разреженной линейной алгебры, использующие возможности GPGPU
 - ▶ Специализированные для булевой алгебры: CuBool (Nvidia Cuda), Spbla (OpenCL)
 - ▶ Обобщённые: Spla (OpenCL), Brahma.FSharp + GraphBLAS# (F# + OpenCL)
- Показана применимость разработанных библиотек для решения задач анализа графов
 - ▶ В том числе, некоторых задач FLPQ

Библиотеки разреженной линейной алгебры

- Спроектированы и реализованы библиотеки разреженной линейной алгебры, использующие возможности GPGPU
 - ▶ Специализированные для булевой алгебры: CuBool (Nvidia Cuda), Spbla (OpenCL)
 - ▶ Обобщённые: Spla (OpenCL), Brahma.FSharp + GraphBLAS# (F# + OpenCL)
- Показана применимость разработанных библиотек для решения задач анализа графов
 - ▶ В том числе, некоторых задач FLPQ
- Сформулированы рекомендации и ограничения
 - ▶ Выделение булевых операций для улучшения производительности
 - ▶ Баланс между трансфером данных и вычислениями (в контексте FLPQ)
 - ▶ Востребованность операций и их модификаций: Kronecker с маской и фильтрами
 - ▶ Использование возможностей функциональных языков программирования (полукольца, Option для матриц смежности)

Методические рекомендации по преподаванию

- Теория формальных языков
 - ▶ Примеры прикладных областей и конкретных задач
 - ▶ Непосредственное использование теоретических результатов
 - ▶ Свежий взгляд на известные задачи (инкрементальный анализ, неоднозначные грамматики)

Методические рекомендации по преподаванию

- Теория формальных языков
 - ▶ Примеры прикладных областей и конкретных задач
 - ▶ Непосредственное использование теоретических результатов
 - ▶ Свежий взгляд на известные задачи (инкрементальный анализ, неоднозначные грамматики)
- Теория графов
 - ▶ Использования и обобщения классических алгоритмов (обход в ширину, транзитивное замыкание)
 - ▶ Использования и обобщения классических структур данных (матрица смежности)
 - ▶ Пример класса задач с требованиями к производительности

Методические рекомендации по преподаванию

- Теория формальных языков
 - ▶ Примеры прикладных областей и конкретных задач
 - ▶ Непосредственное использование теоретических результатов
 - ▶ Свежий взгляд на известные задачи (инкрементальный анализ, неоднозначные грамматики)
- Теория графов
 - ▶ Использования и обобщения классических алгоритмов (обход в ширину, транзитивное замыкание)
 - ▶ Использования и обобщения классических структур данных (матрица смежности)
 - ▶ Пример класса задач с требованиями к производительности
- Алгебра
 - ▶ Применение операций линейной алгебры и их свойств
 - ▶ Нетривиальные алгебраические структуры (неассоциативные полукольца)

Методические рекомендации по преподаванию

- Теория формальных языков
 - ▶ Примеры прикладных областей и конкретных задач
 - ▶ Непосредственное использование теоретических результатов
 - ▶ Свежий взгляд на известные задачи (инкрементальный анализ, неоднозначные грамматики)
- Теория графов
 - ▶ Использования и обобщения классических алгоритмов (обход в ширину, транзитивное замыкание)
 - ▶ Использования и обобщения классических структур данных (матрица смежности)
 - ▶ Пример класса задач с требованиями к производительности
- Алгебра
 - ▶ Применение операций линейной алгебры и их свойств
 - ▶ Нетривиальные алгебраические структуры (неассоциативные полукольца)
- Анализ программ, графовые базы данных
 - ▶ Теоретическая основа
 - ▶ Прикладные алгоритмы, инструменты

Курс по теории формальных языков

- Для инженеров-программистов
 - ▶ Основной акцент на теории, которая важна на практике
 - ▶ Прикладные алгоритмы, особенности их реализации
 - ▶ Связь с другими прикладными областями
- Для каждого класса языков: задача → теоретическая основа → алгоритмы
- Активное использование знаний, полученных на теоретических курсах
 - ▶ Алгебра
 - ▶ Дискретная математика, теория графов
- Классические задачи с нового ракурса
 - ▶ Инкрементальный синтаксический анализ → обработка изменяющихся графов
 - ▶ Параллельный синтаксический анализ → параллельная обработка графов
 - ▶ Работа с неоднозначными грамматиками → неоднозначные запросы

Границы применимости

- Прикладная направленность: основной фокус на получении алгоритмов, имеющих практическую ценность
 - ▶ Возможно ли улучшить оценки сложности для задачи или класса задач?

Границы применимости

- Прикладная направленность: основной фокус на получении алгоритмов, имеющих практическую ценность
 - ▶ Возможно ли улучшить оценки сложности для задачи или класса задач?
- Не обсуждаются частные/крайние случаи
 - ▶ Что делать, если известно, что граф — дерево?
 - ▶ Можно ли улучшить решение, если известно, что язык принадлежит специальному узкому классу?

Границы применимости

- Прикладная направленность: основной фокус на получении алгоритмов, имеющих практическую ценность
 - ▶ Возможно ли улучшить оценки сложности для задачи или класса задач?
- Не обсуждаются частные/крайние случаи
 - ▶ Что делать, если известно, что граф — дерево?
 - ▶ Можно ли улучшить решение, если известно, что язык принадлежит специальному узкому классу?
- Не обсуждаются некоторые особенности прикладных областей
 - ▶ Реализуем ли мы статический анализ для IDE или же для компилятора
 - ▶ Разрабатываем мы алгоритм для графовой базы данных с OLAP нагрузкой или же с OLTP

Границы применимости

- Прикладная направленность: основной фокус на получении алгоритмов, имеющих практическую ценность
 - ▶ Возможно ли улучшить оценки сложности для задачи или класса задач?
- Не обсуждаются частные/крайние случаи
 - ▶ Что делать, если известно, что граф — дерево?
 - ▶ Можно ли улучшить решение, если известно, что язык принадлежит специальному узкому классу?
- Не обсуждаются некоторые особенности прикладных областей
 - ▶ Реализуем ли мы статический анализ для IDE или же для компилятора
 - ▶ Разрабатываем мы алгоритм для графовой базы данных с OLAP нагрузкой или же с OLTP
- Не обсуждается удобство для прикладного пользователя
 - ▶ Как естественным образом встраивать рассматриваемые ограничения в язык запросов к графикам?
 - ▶ Как должен выглядеть интерфейс унифицированного решателя задач FLPQ?

Результаты

1 !!!

Соответствие паспорту специальности 2.3.5

- Пункту 1: модели, методы и алгоритмы проектирования, анализа, трансформации, верификации и тестирования программ и программных систем
- Пункту 4: интеллектуальные системы машинного обучения, управления базами данных и знаний, инструментальные средства разработки цифровых продуктов
- Пункту 8: модели и методы создания программ и программных систем для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования
- Пункту 10: оценка качества, стандартизация и сопровождение программных систем

Основные публикации

- ① Epelbaum I.V., Azimov R.Sh., Grigorev S.V. Multiple Context-Free Path Querying by Matrix Multiplication // Научно-технический вестник информационных технологий, механики и оптики. 2023.
- ② Egor Orachev, Maria Karpenko, Pavel Alimov, Semyon Grigorev // SPbLA: The Library of GPGPU-powered Sparse Boolean Linear Algebra Operations. Journal of Open Source Software, 2022
- ③ Shemetova, E.N., Grigorev, S.V. // Path Querying on Acyclic Graphs Using Boolean Grammars. Program Comput Soft, 2021
- ④ Azimov, R., Grigorev, S. // Path Querying with Conjunctive Grammars by Matrix Multiplication. Program Comput Soft, 2019
- ⑤ Shemetova, E., Okhotin, A., Grigorev, S. // Rational Index of Languages Defined by Grammars with Bounded Dimension of Parse Trees. Theory Comput Syst, 2025

Основные публикации

- ⑥ Азимов Р. Ш., Григорьев С. В. Алгоритм поиска всех путей в графе с заданными контекстно-свободными ограничениями с использованием матриц с множествами промежуточных вершин // Научно-технический вестник информационных технологий, механики и оптики, 2021
- ⑦ С. В. Григорьев и др., Инstrumentальная поддержка встроенных языков в интегрированных средах разработки // Модел. и анализ информ. систем, 2014

⑧ 

⑨ 

⑩ 

⑪ 

Конференции, семинары, гранты

- Конференции (более 10)
 - ▶ Анализ данных и базы данных: EDBT-2021, ADBIS-2020, DAMDID-2025
 - ▶ Анализ графов: GRADES-NDA-2018/2019/2020/2021, Grapl-2021, LSGDA-2025
 - ▶ Статический анализ кода: SOAP-2025, SIGPLAN International Symposium on Scala 2018
 - ▶ Иное: SECR-2017, PSI-2015, PaCT-2025, SYRCoSE-2025

Конференции, семинары, гранты

- Конференции (более 10)
 - ▶ Анализ данных и базы данных: EDBT-2021, ADBIS-2020, DAMDID-2025
 - ▶ Анализ графов: GRADES-NDA-2018/2019/2020/2021, Grapl-2021, LSGDA-2025
 - ▶ Статический анализ кода: SOAP-2025, SIGPLAN International Symposium on Scala 2018
 - ▶ Иное: SECR-2017, PSI-2015, PaCT-2025, SYRCoSE-2025
- Семинары
 - ▶ Семинар кафедры Системного программирования
 - ▶ Data.Meetup: графовые решения. Финтех и телеком

Конференции, семинары, гранты

- Конференции (более 10)
 - ▶ Анализ данных и базы данных: EDBT-2021, ADBIS-2020, DAMDID-2025
 - ▶ Анализ графов: GRADES-NDA-2018/2019/2020/2021, Grapl-2021, LSGDA-2025
 - ▶ Статический анализ кода: SOAP-2025, SIGPLAN International Symposium on Scala 2018
 - ▶ Иное: SECR-2017, PSI-2015, PaCT-2025, SYRCoSE-2025
- Семинары
 - ▶ Семинар кафедры Системного программирования
 - ▶ Data.Meetup: графовые решения. Финтех и телеком
- Гранты
 - ▶ Грант РНФ 18-11-00100 «Логические и алгебраические методы в теории формальных языков»
 - ▶ Грант РФФИ 19-37-90101 «Поиск путей с ограничениями в терминах формальных языков»
 - ▶ Грант СПбГУ 116636233 «Методы искусственного интеллекта в задачах механики сплошных сред»