Анализ графов с использованием формальных языков в качестве ограничений на пути

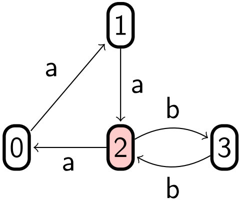
Заготовки для докторской диссертации

Семён Григорьев

Санкт-Петербургский Государственный Университет

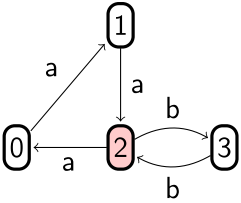
!!! !!!! 2026

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 $\Sigma \cap L \neq \varnothing$

Формальные языки и ограничения на пути в графе

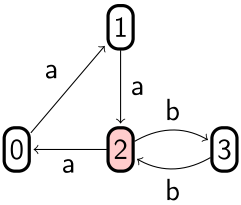


- $G = \langle V, E, L \rangle$ — (ориентированный) граф с метками на рёбрах
- Путь π задаёт слово: $\omega({}_2\pi_1) = \omega(2 \xrightarrow{a} 0 \xrightarrow{a} 1) = aa$
- Ищем пути, задающие слова определённого вида
  - Например, слова вида a^*b
- Множество слов — язык $\mathcal{L}$ над алфавитом Σ
  - Для удобства будем считать, что $\Sigma \cap L \neq \varnothing$

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



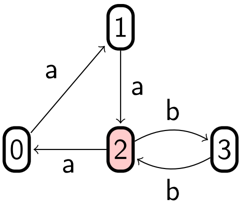
- $G = \langle V, E, L \rangle$ — (ориентированный) граф с метками на рёбрах
- Путь π задаёт слово: $\omega({}_2\pi_1) = \omega(2 \xrightarrow{a} 0 \xrightarrow{a} 1) = aa$
- Ищем пути, задающие слова определённого вида
  - Например, слова вида a^*b
- Множество слов — язык $\mathcal{L}$ над алфавитом Σ
  - Для удобства будем считать, что $\Sigma \cap L \neq \varnothing$

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



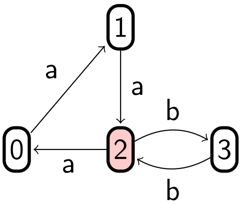
- $G = \langle V, E, L \rangle$ — (ориентированный) граф с метками на рёбрах
- Путь π задаёт слово: $\omega({}_2\pi_1) = \omega(2 \xrightarrow{a} 0 \xrightarrow{a} 1) = aa$
- Ищем пути, задающие слова определённого вида
  - Например, слова вида a^*b
- Множество слов — язык $\mathcal{L}$ над алфавитом Σ
  - Для удобства будем считать, что $\Sigma \cap L \neq \varnothing$

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: $\omega({}_2\pi_1) = \omega(2 \xrightarrow{a} 0 \xrightarrow{a} 1) = aa$
- Ищем пути, задающие слова определённого вида
  - Например, слова вида a^*b
- Множество слов — язык $\mathcal{L}$ над алфавитом Σ
  - Для удобства будем считать, что $\Sigma \cap L \neq \varnothing$

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

Области применения

- Регулярные языки
  - Графовые базы данных (Regular Path Queries, RPQ), ISO, !!!!
  - Параметризация алгебраическими структурами: полукольцами, моноидами и т.д.

Области применения

- Регулярные языки
  - Графовые базы данных (Regular Path Queries, RPQ), ISO, !!!!
  - Параметризация алгебраическими структурами: полукольцами, моноидами и т.д.
- Контекстно-свободные языки
  - Статический анализ кода, унификация !!!
  - Графовые базы данных
  - !!!

Области применения

- Регулярные языки
  - Графовые базы данных (Regular Path Queries, RPQ), ISO, !!!!
  - Параметризация алгебраическими структурами: полукольцами, моноидами и т.д.
- Контекстно-свободные языки
  - Статический анализ кода, унификация !!!
  - Графовые базы данных
  - !!!
- За пределами контекстно-свободных
  - Многокомпонентные контекстно-свободные !!!
  - Линейные конъюнктивные !!!

Состояние дел

- API для создания алгоритмов анализа графов на основе линейной алгебры
  - Различные операции над матрицами и векторами (**разреженными**)
  - Параметризация алгебраическими структурами: полукольцами, моноидами и т.д.

[1] https://graphblas.org/GraphBLAS-Pointers/
[2] https://github.com/DrTimothyAldenDavis/GraphBLAS
[3] https://github.com/GraphBLAS/LAGraph

Состояние дел

- API для создания алгоритмов анализа графов на основе линейной алгебры
  - Различные операции над матрицами и векторами (**разреженными**)
  - Параметризация алгебраическими структурами: полукольцами, моноидами и т.д.
- Позволяет выражать **различные** алгоритмы
  - Обход в ширину, поиск кратчайших путей, достижимость, ...
  - Подсчёт треугольников, PageRank, остовные деревья, кластеризация, ...
  - Навигационные запросы: **RPQ, CFPQ,** ...

[1] https://graphblas.org/GraphBLAS-Pointers/
[2] https://github.com/DrTimothyAldenDavis/GraphBLAS
[3] https://github.com/GraphBLAS/LAGraph

Состояние дел

- API для создания алгоритмов анализа графов на основе линейной алгебры
  - Различные операции над матрицами и векторами (**разреженными**)
  - Параметризация алгебраическими структурами: полукольцами, моноидами и т.д.
- Позволяет выражать **различные** алгоритмы
  - Обход в ширину, поиск кратчайших путей, достижимость, . . .
  - Подсчёт треугольников, PageRank, остовные деревья, кластеризация, . . .
  - Навигационные запросы: **RPQ, CFPQ,** . . .
- Подробнее
  - GraphBLAS Pointers[1]
  - **SuiteSparse:GraphBLAS**[2] — **эталон** на чистом C
  - **LAGraph**[3] — коллекция **прикладных** алгоритмов анализа графов

[1] https://graphblas.org/GraphBLAS-Pointers/
[2] https://github.com/DrTimothyAldenDavis/GraphBLAS
[3] https://github.com/GraphBLAS/LAGraph

Проблемы

$R_1 = \{(a, y); (b, x); (b, z)\}$

$R_2 = \{(x, u); (x, w); (y, v); (y, w); (z, u)\}$

$|S_A| = n, |S_B| = k, |S_C| = m$

$\mathcal{N}_{\mathcal{A}} : [0 \dots (n-1)] \to S_A,$ биекция

$\mathcal{N}_{\mathcal{B}} : [0 \dots (k-1)] \to S_B,$ биекция

$\mathcal{N}_{\mathcal{C}} : [0 \dots (m-1)] \to S_C,$ биекция

$R_1 \subseteq S_A \times S_B, R_2 \subseteq S_B \times S_C$

$R_3 = \{(x, z) \mid (x, y) \in R_1, (y, z) \in R_2\}$

$M^{R_1}_{n \times k} = 1 \iff (\mathcal{N}_{\mathcal{A}}(i), \mathcal{N}_{\mathcal{B}}(j)) \in R_1,$ иначе $M^{R_1} = 0$

$M^{R_2}_{k \times m} = 1 \iff (\mathcal{N}_{\mathcal{B}}(i), \mathcal{N}_{\mathcal{C}}(j)) \in R_2,$ иначе $M^{R_2} = 0$

$M^{R_3}_{n \times m} = M^{R_1} \times M^{R_2}$

Проблемы

$$R_1 = \{(a, y); (b, x); (b, z)\}$$
$$R_2 = \{(x, u); (x, w); (y, v); (y, w); (z, u)\}$$
$$R_3 = \{(i, j) \mid (i, k) \in R_1, (k, j) \in R_2\}$$

$|S_A| = n, |S_B| = k, |S_C| = m$

$\mathcal{N}_{\mathcal{A}} : [0 \ldots (n-1)] \to S_A,$ биекция

$\mathcal{N}_{\mathcal{B}} : [0 \ldots (k-1)] \to S_B,$ биекция

$\mathcal{N}_{\mathcal{C}} : [0 \ldots (m-1)] \to S_C,$ биекция

$R_1 \subseteq S_A \times S_B, R_2 \subseteq S_B \times S_C$

$R_3 = \{(x, z) \mid (x, y) \in R_1, (y, z) \in R_2\}$

$M^{R_1}_{n \times k} = 1 \iff (\mathcal{N}_{\mathcal{A}}(i), \mathcal{N}_{\mathcal{B}}(j)) \in R_1,$ иначе $M^{R_1} = 0$

$M^{R_2}_{k \times m} = 1 \iff (\mathcal{N}_{\mathcal{B}}(i), \mathcal{N}_{\mathcal{C}}(j)) \in R_2,$ иначе $M^{R_2} = 0$

$M^{R_3}_{n \times m} = M^{R_1} \times M^{R_2}$

Проблемы

$R_1 = \{(a,y);(b,x);(b,z)\}$

$R_2 = \{(x,u);(x,w);(y,v);(y,w);(z,u)\}$

$R_3 = \{(i,j) \mid (i,k) \in R_1, (k,j) \in R_2\}$

$$\begin{array}{c c} & \begin{array}{ccc} x & y & z \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \times \begin{array}{c c} & \begin{array}{ccc} u & v & w \end{array} \\ \begin{array}{c} x \\ y \\ z \end{array} & \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array} = \begin{array}{c c} & \begin{array}{ccc} u & v & w \end{array} \\ \begin{array}{c} a \\ b \end{array} & \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array}$$

$|S_A| = n, |S_B| = k, |S_C| = m$

$\mathcal{N_A} : [0 \dots (n-1)] \to S_A,$ биекция

$\mathcal{N_B} : [0 \dots (k-1)] \to S_B,$ биекция

$\mathcal{N_C} : [0 \dots (m-1)] \to S_C,$ биекция

$R_1 \subseteq S_A \times S_B, R_2 \subseteq S_B \times S_C$

$R_3 = \{(x,z) \mid (x,y) \in R_1, (y,z) \in R_2\}$

$M^{R_1}_{n \times k} = 1 \iff (\mathcal{N_A}(i), \mathcal{N_B}(j)) \in R_1,$ иначе $M^{R_1} = 0$

$M^{R_2}_{k \times m} = 1 \iff (\mathcal{N_B}(i), \mathcal{N_C}(j)) \in R_2,$ иначе $M^{R_2} = 0$

$M^{R_3}_{n \times m} = M^{R_1} \times M^{R_2}$

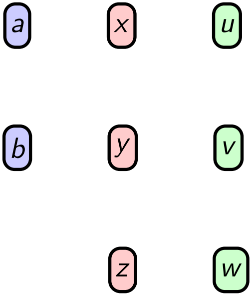
Проблемы

$$R_1 = \{(a, y); (b, x); (b, z)\}$$

$$R_2 = \{(x, u); (x, w); (y, v); (y, w); (z, u)\}$$

$$R_3 = \{(i, j) \mid (i, k) \in R_1, (k, j) \in R_2\}$$

$$\begin{array}{c} \\ a \\ b \end{array}\begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \times \begin{array}{c} \\ x \\ y \\ z \end{array}\begin{array}{c} \begin{array}{ccc} u & v & w \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array} = \begin{array}{c} \\ a \\ b \end{array}\begin{array}{c} \begin{array}{ccc} u & v & w \end{array} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array}$$



$|S_A| = n, |S_B| = k, |S_C| = m$

$\mathcal{N}_{\mathcal{A}} : [0 \ldots (n-1)] \to S_A,$ биекция

$\mathcal{N}_{\mathcal{B}} : [0 \ldots (k-1)] \to S_B,$ биекция

$\mathcal{N}_{\mathcal{C}} : [0 \ldots (m-1)] \to S_C,$ биекция

$R_1 \subseteq S_A \times S_B, R_2 \subseteq S_B \times S_C$

$R_3 = \{(x, z) \mid (x, y) \in R_1, (y, z) \in R_2\}$

$M^{R_1}_{n \times k} = 1 \iff (\mathcal{N}_{\mathcal{A}}(i), \mathcal{N}_{\mathcal{B}}(j)) \in R_1,$ иначе $M^{R_1} = 0$

$M^{R_2}_{k \times m} = 1 \iff (\mathcal{N}_{\mathcal{B}}(i), \mathcal{N}_{\mathcal{C}}(j)) \in R_2,$ иначе $M^{R_2} = 0$

$M^{R_3}_{n \times m} = M^{R_1} \times M^{R_2}$

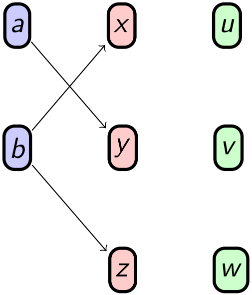
Проблемы

$R_1 = \{(a, y); (b, x); (b, z)\}$

$R_2 = \{(x, u); (x, w); (y, v); (y, w); (z, u)\}$

$R_3 = \{(i, j) \mid (i, k) \in R_1, (k, j) \in R_2\}$

$$\begin{array}{c} \\ a \\ b \end{array}\!\!\begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \times \begin{array}{c} \\ x \\ y \\ z \end{array}\!\!\begin{array}{c} \begin{array}{ccc} u & v & w \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array} = \begin{array}{c} \\ a \\ b \end{array}\!\!\begin{array}{c} \begin{array}{ccc} u & v & w \end{array} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array}$$



$|S_A| = n, |S_B| = k, |S_C| = m$

$\mathcal{N}_{\mathcal{A}} : [0 \dots (n-1)] \to S_A,$ биекция

$\mathcal{N}_{\mathcal{B}} : [0 \dots (k-1)] \to S_B,$ биекция

$\mathcal{N}_{\mathcal{C}} : [0 \dots (m-1)] \to S_C,$ биекция

$R_1 \subseteq S_A \times S_B, R_2 \subseteq S_B \times S_C$

$R_3 = \{(x, z) \mid (x, y) \in R_1, (y, z) \in R_2\}$

$M^{R_1}_{n \times k} = 1 \iff (\mathcal{N}_{\mathcal{A}}(i), \mathcal{N}_{\mathcal{B}}(j)) \in R_1,$ иначе $M^{R_1} = 0$

$M^{R_2}_{k \times m} = 1 \iff (\mathcal{N}_{\mathcal{B}}(i), \mathcal{N}_{\mathcal{C}}(j)) \in R_2,$ иначе $M^{R_2} = 0$

$M^{R_3}_{n \times m} = M^{R_1} \times M^{R_2}$

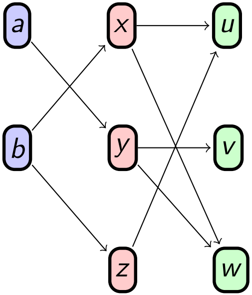
Проблемы

$R_1 = \{(a, y); (b, x); (b, z)\}$

$R_2 = \{(x, u); (x, w); (y, v); (y, w); (z, u)\}$

$R_3 = \{(i, j) \mid (i, k) \in R_1, (k, j) \in R_2\}$

$$\begin{array}{c} \\ a \\ b \end{array}\begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \times \begin{array}{c} \\ x \\ y \\ z \end{array}\begin{array}{c} \begin{array}{ccc} u & v & w \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array} = \begin{array}{c} \\ a \\ b \end{array}\begin{array}{c} \begin{array}{ccc} u & v & w \end{array} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array}$$



$|S_A| = n, |S_B| = k, |S_C| = m$

$\mathcal{N}_{\mathcal{A}} : [0 \dots (n-1)] \to S_A,$ биекция

$\mathcal{N}_{\mathcal{B}} : [0 \dots (k-1)] \to S_B,$ биекция

$\mathcal{N}_{\mathcal{C}} : [0 \dots (m-1)] \to S_C,$ биекция

$R_1 \subseteq S_A \times S_B, R_2 \subseteq S_B \times S_C$

$R_3 = \{(x, z) \mid (x, y) \in R_1, (y, z) \in R_2\}$

$M^{R_1}_{n \times k} = 1 \iff (\mathcal{N}_{\mathcal{A}}(i), \mathcal{N}_{\mathcal{B}}(j)) \in R_1,$ иначе $M^{R_1} = 0$

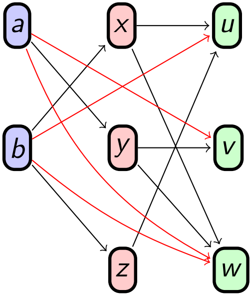
$M^{R_2}_{k \times m} = 1 \iff (\mathcal{N}_{\mathcal{B}}(i), \mathcal{N}_{\mathcal{C}}(j)) \in R_2,$ иначе $M^{R_2} = 0$

$M^{R_3}_{n \times m} = M^{R_1} \times M^{R_2}$

Проблемы

$$R_1 = \{(a, y); (b, x); (b, z)\}$$
$$R_2 = \{(x, u); (x, w); (y, v); (y, w); (z, u)\}$$
$$R_3 = \{(i, j) \mid (i, k) \in R_1, (k, j) \in R_2\}$$

$$\begin{array}{c} \\ a \\ b \end{array}\!\!\begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \end{array} \times \begin{array}{c} \\ x \\ y \\ z \end{array}\!\!\begin{array}{c} \begin{array}{ccc} u & v & w \end{array} \\ \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array} = \begin{array}{c} \\ a \\ b \end{array}\!\!\begin{array}{c} \begin{array}{ccc} u & v & w \end{array} \\ \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \end{array}$$



$|S_A| = n, |S_B| = k, |S_C| = m$

$\mathcal{N}_{\mathcal{A}} : [0 \ldots (n-1)] \to S_A,$ биекция

$\mathcal{N}_{\mathcal{B}} : [0 \ldots (k-1)] \to S_B,$ биекция

$\mathcal{N}_{C} : [0 \ldots (m-1)] \to S_C,$ биекция

$R_1 \subseteq S_A \times S_B, R_2 \subseteq S_B \times S_C$

$R_3 = \{(x, z) \mid (x, y) \in R_1, (y, z) \in R_2\}$

$M^{R_1}_{n \times k} = 1 \iff (\mathcal{N}_{\mathcal{A}}(i), \mathcal{N}_{\mathcal{B}}(j)) \in R_1,$ иначе $M^{R_1} = 0$

$M^{R_2}_{k \times m} = 1 \iff (\mathcal{N}_{\mathcal{B}}(i), \mathcal{N}_{C}(j)) \in R_2,$ иначе $M^{R_2} = 0$

$M^{R_3}_{n \times m} = M^{R_1} \times M^{R_2}$

Цель работы

Целью диссертационной работы является создание общей методологии анализа графов с использованием формальных языков в качестве ограничений на пути. Для её достижения необходимо решить следующие задачи. Провести систематизацию имеющихся результатов, с одной стороны, для выявления мест, требующих более детального изучения и формализации, а с другой, для обнаружения общих закономерностей, позволяющих построить общее описание рассматриваемой области. В частности, необходимо изучить существующие алгоритмы решения задачи анализа графов с использованием формальных языков в качестве ограничений на пути. Кроме этого, необходимо провести анализ областей применения, выявление характерных особенностей задач, возникающих в них, выявить и структурировать основные требования к применяемым в данных областях алгоритмам. Разработать методологию анализа графов с использованием формальных языков в качестве ограничений на пути, предоставляющую средства для классификации задач, включающую методы построения и анализа алгоритмов решения соответствующих задач. Разработать метод создания алгоритмов, в том числе параллельных, на основе линейной алгебры для решения задач анализа графов с использованием формальных языков в качестве ограничений на пути

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. Разработаны 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. Разработаны 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. Разработаны 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. Разработаны методы конструирования алгоритмов решения FLPQ
   - Использующих операции линейной алгебры
   - Основанных на классических алгоритмах синтаксического анализа
3. Разработана методика проведения 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. Разработаны методы конструирования алгоритмов решения FLPQ
   - Использующих операции линейной алгебры
   - Основанных на классических алгоритмах синтаксического анализа
3. Разработана методика проведения 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. Разработаны методы конструирования алгоритмов решения FLPQ
   - Использующих операции линейной алгебры
   - Основанных на классических алгоритмах синтаксического анализа
3. Разработана методика проведения 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. Разработаны методы конструирования алгоритмов решения FLPQ
   - Использующих операции линейной алгебры
   - Основанных на классических алгоритмах синтаксического анализа
3. Разработана методика проведения 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

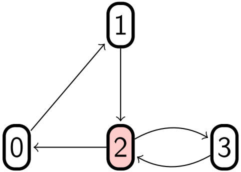
Положения, выносимые на защиту

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

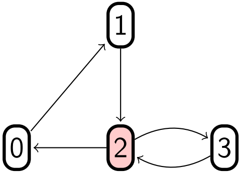
Положения, выносимые на защиту

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

Обход в ширину



Обход в ширину



Текущий фронт

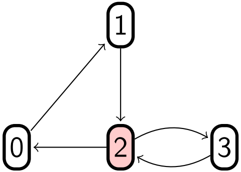
Матрица смежности

Новый фронт

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

Полукольцо

Обход в ширину



Текущий фронт

Матрица смежности

Новый фронт

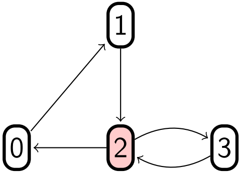
$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

Полукольцо

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$$

Посещённые вершины

Обход в ширину



Текущий фронт

Матрица смежности

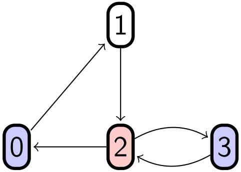
Новый фронт

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

Полукольцо

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$$

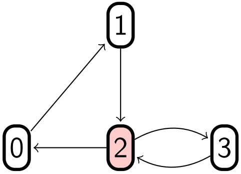
Посещённые вершины



$$\begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 1 & 1 \end{pmatrix}$$

Обход в ширину



Текущий фронт

Матрица смежности

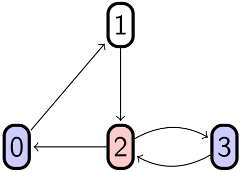
Новый фронт

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

Полукольцо

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$$

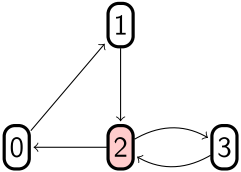
Посещённые вершины



$$\begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & \cancel{1} & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 1 & 1 \end{pmatrix}$$

Обход в ширину



Текущий фронт

Матрица смежности

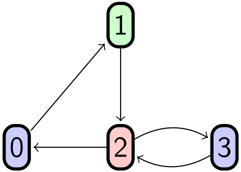
Новый фронт

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

Полукольцо

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$$

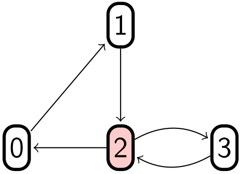
Посещённые вершины



$$\begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$$

Обход в ширину



Текущий фронт

Матрица смежности

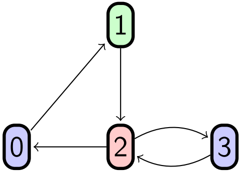
Новый фронт

$$\begin{pmatrix}0 & 0 & 1 & 0\end{pmatrix} \times \begin{pmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 1\end{pmatrix}$$

Полукольцо

$$\begin{pmatrix}0 & 0 & 1 & 0\end{pmatrix}$$

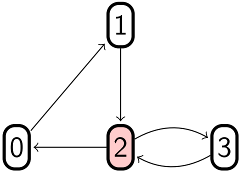
Посещённые вершины



$$\begin{pmatrix}0 & 1 & 0 & 0\end{pmatrix} \times \begin{pmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix} = \begin{pmatrix}0 & 0 & \cancel{1} & 0\end{pmatrix}$$

$$\begin{pmatrix}1 & 1 & 1 & 1\end{pmatrix}$$

Обход в ширину



Текущий фронт

Матрица смежности

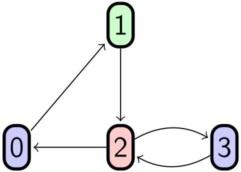
Новый фронт

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

Полукольцо

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$$

Посещённые вершины

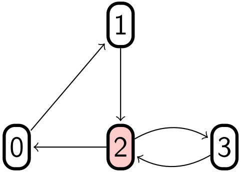


$$\begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cancel{1} & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$$

result: $\begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$

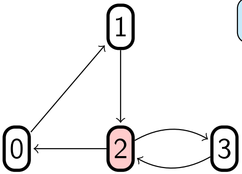
Обход в ширину с построением дерева обхода



Обход в ширину с построением дерева обхода

$p \otimes (i, j, \_) = i$: получаем предка

$i_1 \oplus i_2 = i_1$: из нескольких возможных предков возьмём первого



У стартовой нет предка

$$\begin{pmatrix} \_ & \_ & \boxed{-1} & \_ \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & \_ & \_ & 2 \end{pmatrix}$$

$$\begin{pmatrix} \_ & \_ & -1 & \_ \end{pmatrix}$$

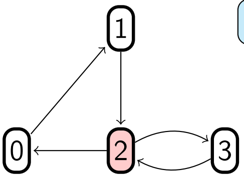
«Нейтральный элемент»

Стартовая — предок

Обход в ширину с построением дерева обхода

$p \otimes (i, j, \_) = i$: получаем предка

$i_1 \oplus i_2 = i_1$: из нескольких возможных предков возьмём первого



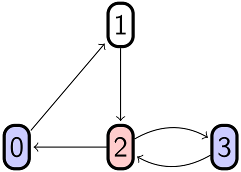
У стартовой нет предка

Стартовая — предок

$$\begin{pmatrix} \_ & \_ & -1 & \_ \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & \_ & \_ & 2 \end{pmatrix}$$

$$\begin{pmatrix} \_ & \_ & -1 & \_ \end{pmatrix}$$

«Нейтральный элемент»



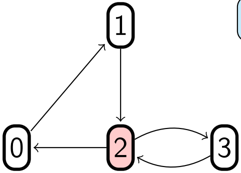
$$\begin{pmatrix} 2 & \_ & \_ & 2 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \_ & 0 & 3 & \_ \end{pmatrix}$$

$$\begin{pmatrix} 2 & \_ & -1 & 2 \end{pmatrix}$$

Обход в ширину с построением дерева обхода

$p \otimes (i, j, \_) = i$: получаем предка

$i_1 \oplus i_2 = i_1$: из нескольких возможных предков возьмём первого



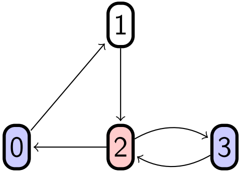
У стартовой нет предка

Стартовая — предок

$$\begin{pmatrix} \_ & \_ & -1 & \_ \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & \_ & \_ & 2 \end{pmatrix}$$

$$\begin{pmatrix} \_ & \_ & -1 & \_ \end{pmatrix}$$

«Нейтральный элемент»



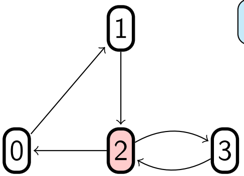
$$\begin{pmatrix} 2 & \_ & \_ & 2 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \_ & 0 & \cancel{3} & \_ \end{pmatrix}$$

$$\begin{pmatrix} 2 & \_ & -1 & 2 \end{pmatrix}$$

Обход в ширину с построением дерева обхода

$p \otimes (i, j, \_) = i$: получаем предка

$i_1 \oplus i_2 = i_1$: из нескольких возможных предков возьмём первого



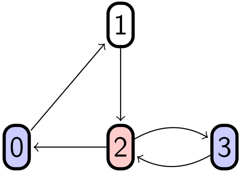
У стартовой нет предка

$$\begin{pmatrix} \_ & \_ & -1 & \_ \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & \_ & \_ & 2 \end{pmatrix}$$

$$\begin{pmatrix} \_ & \_ & -1 & \_ \end{pmatrix}$$

Стартовая — предок

«Нейтральный элемент»



$$\begin{pmatrix} 2 & \_ & \_ & 2 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \_ & 0 & \cancel{3} & \_ \end{pmatrix}$$

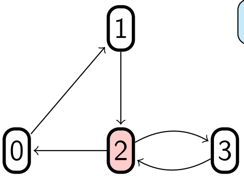
$$\begin{pmatrix} 2 & \_ & -1 & 2 \end{pmatrix}$$

result: $\begin{pmatrix} 2 & 0 & -1 & 2 \end{pmatrix}$

Обход в ширину с построением дерева обхода

$p \otimes (i, j, \_) = i$: получаем предка

$i_1 \oplus i_2 = i_1$: из нескольких возможных предков возьмём первого



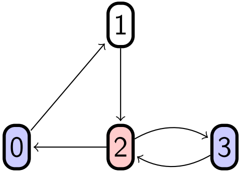
У стартовой нет предка

Стартовая — предок

«Нейтральный элемент»

$$(\_\ \ \_\ \ -1\ \ \_) \times \begin{pmatrix}0&1&0&0\\0&0&1&0\\1&0&0&1\\0&0&1&0\end{pmatrix} = (2\ \ \_\ \ \_\ \ 2)$$

$$(\_\ \ \_\ \ -1\ \ \_)$$



$$(2\ \ \_\ \ \_\ \ 2) \times \begin{pmatrix}0&1&0&0\\0&0&1&0\\1&0&0&1\\0&0&1&0\end{pmatrix} = (\_\ \ 0\ \ \cancel{3}\ \ \_)$$

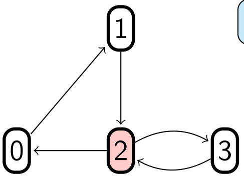
$$(2\ \ \_\ \ -1\ \ 2)$$

result: $(2\ \ 0\ \ -1\ \ 2)$

Обход в ширину с построением дерева обхода

$p \otimes (i, j, \_) = i$: получаем предка

$i_1 \oplus i_2 = i_1$: из нескольких возможных предков возьмём первого



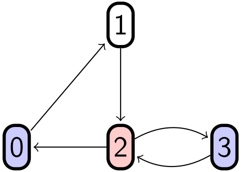
У стартовой нет предка

Стартовая — предок

$$\begin{pmatrix} \_ & \_ & -1 & \_ \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & \_ & \_ & 2 \end{pmatrix}$$

$$\begin{pmatrix} \_ & \_ & -1 & \_ \end{pmatrix}$$

«Нейтральный элемент»



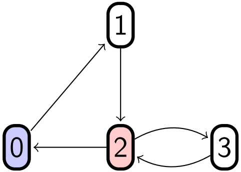
$$\begin{pmatrix} 2 & \_ & \_ & 2 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} \_ & 0 & \cancel{3} & \_ \end{pmatrix}$$

$$\begin{pmatrix} 2 & \_ & -1 & 2 \end{pmatrix}$$

result: $\begin{pmatrix} 2 & 0 & -1 & 2 \end{pmatrix}$

Выбор входящих и исходящих (инцидентных) рёбер

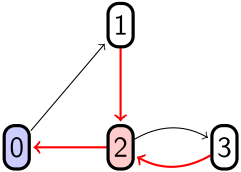
Выбор входящих рёбер — умножение на диагональную матрицу справа



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} =$$

Выбор входящих и исходящих (инцидентных) рёбер

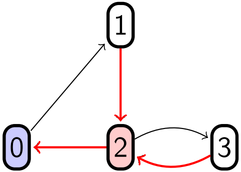
Выбор входящих рёбер — умножение на диагональную матрицу справа



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

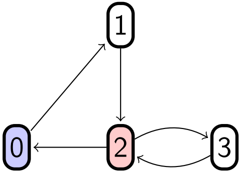
Выбор входящих и исходящих (инцидентных) рёбер

Выбор входящих рёбер — умножение на диагональную матрицу справа



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

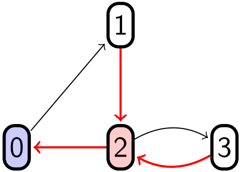
Выбор исходящих рёбер — умножение на диагональную матрицу слева



$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} =$$

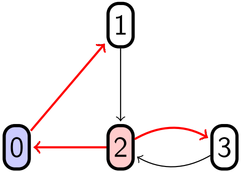
Выбор входящих и исходящих (инцидентных) рёбер

Выбор входящих рёбер — умножение на диагональную матрицу справа



$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Выбор исходящих рёбер — умножение на диагональную матрицу слева



$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Пример[4]

- Граф с двумя типами вершин: пользователи и карты
  - «Пользователь»: пол, возраст
  - «Карта»: тип (МИР, VISA, MASTERCARD), лимит средств

[4] Код и описание: https://github.com/SparseLinearAlgebra/PageRankBenchmark

Пример[4]

- Граф с двумя типами вершин: пользователи и карты
  - «Пользователь»: пол, возраст
  - «Карта»: тип (МИР, VISA, MASTERCARD), лимит средств
- Два типа ориентированных рёбер
  - «Перевод»: соединяет две карты (откуда и куда перевод)
    - Метка: общая сумма и «количество транзакций»
  - «Владеет»: соединяет пользователя и карту (от владельца карты к карте)
    - Не имеет меток

[4] Код и описание: https://github.com/SparseLinearAlgebra/PageRankBenchmark

Пример[4]

- Граф с двумя типами вершин: пользователи и карты
  - «Пользователь»: пол, возраст
  - «Карта»: тип (МИР, VISA, MASTERCARD), лимит средств
- Два типа ориентированных рёбер
  - «Перевод»: соединяет две карты (откуда и куда перевод)
    - Метка: общая сумма и «количество транзакций»
  - «Владеет»: соединяет пользователя и карту (от владельца карты к карте)
    - Не имеет меток

1. Выбрать хотим все карты системы «МИР», которыми владеют люди старше заданного возраста
2. Посчитать PageRank на подграфе, заданном переводами между отобранными картами

[4] Код и описание: https://github.com/SparseLinearAlgebra/PageRankBenchmark

Представление графа

- Пользовательские типы (структуры) для описания пользователя, карты, информации о переводах: `User`, `Card`, `Trans`

Представление графа

- Пользовательские типы (структуры) для описания пользователя, карты, информации о переводах: `User`, `Card`, `Trans`
- Булева матрица $\texttt{Own\_Edges}_{12\times12}$ задаёт отношение «Владеет»

Представление графа

- Пользовательские типы (структуры) для описания пользователя, карты, информации о переводах: `User`, `Card`, `Trans`
- Булева матрица $\texttt{Own\_Edges}_{12\times 12}$ задаёт отношение «Владеет»
- Матрица $\texttt{Trans\_Edges}_{12\times 12}$ типа `Matrix<Trans>` задаёт отношение «Перевод»

Представление графа

- Пользовательские типы (структуры) для описания пользователя, карты, информации о переводах: `User`, `Card`, `Trans`
- Булева матрица $\texttt{Own\_Edges}_{12\times12}$ задаёт отношение «Владеет»
- Матрица $\texttt{Trans\_Edges}_{12\times12}$ типа `Matrix<Trans>` задаёт отношение «Перевод»
- Вектор пользователей $\texttt{Users}_{12}$ хранит данные о пользователях

Представление графа

- Пользовательские типы (структуры) для описания пользователя, карты, информации о переводах: `User`, `Card`, `Trans`
- Булева матрица $\texttt{Own\_Edges}_{12\times 12}$ задаёт отношение «Владеет»
- Матрица $\texttt{Trans\_Edges}_{12\times 12}$ типа `Matrix<Trans>` задаёт отношение «Перевод»
- Вектор пользователей $\texttt{Users}_{12}$ хранит данные о пользователях
- Вектор карт $\texttt{Cards}_{12}$ хранит данные о картах

Построение подграфа

- `filter` — пользовательский предикат

Построение подграфа

- `filter` — пользовательский предикат

Построение подграфа

- Выбираем карты пользователей
  - Один шаг обхода в ширину
  - `s_cards = s_users * Owns_Edges`
  - `s_cards` — не карты, а их индексы

- filter — пользовательский предикат

Построение подграфа

- Выбираем карты пользователей
  - Один шаг обхода в ширину
  - s_cards = s_users * Owns_Edges
  - s_cards — не карты, а их индексы
- Выбираем карты «МИР»
  - s_cards используем как маску
  - s_cards’ = Mask(s_cards, Choose(filter’, Cards))

- filter — пользовательский предикат

Построение подграфа

- Выбираем карты пользователей
  - Один шаг обхода в ширину
  - s_cards = s_users * Owns_Edges
  - s_cards — не карты, а их индексы
- Выбираем карты «МИР»
  - s_cards используем как маску
  - s_cards' = Mask(s_cards, Choose(filter', Cards))
- Выбираем переводы между отобранными картами
  - Рёбра, идущие **только** между выбранными картами
  - cards_subgraph = Diag(s_cards') * Trans_Edges * Diag(s_cards')

Получение матрицы весов по данным о переводах

- Преобразовываем структуры (метки рёбер) в веса: $f(w) = \frac{w.Sum}{w.Count * 1000}$

Получение матрицы весов по данным о переводах

- Преобразовываем структуры (метки рёбер) в веса: $f(w) = \frac{w.Sum}{w.Count * 1000}$
- Сумма весов исходящих рёбер равна 1: $SoftMax(\overrightarrow{z})_i = \frac{e^{f(z_i)}}{\sum\limits_{k=1}^{K} e^{f(z_k)}}$

Получение матрицы весов по данным о переводах

- Преобразовываем структуры (метки рёбер) в веса: $f(w) = \frac{w.Sum}{w.Count*1000}$
- Сумма весов исходящих рёбер равна 1: $SoftMax(\overrightarrow{z})_i = \frac{e^{f(z_i)}}{\sum\limits_{k=1}^{K} e^{f(z_k)}}$
- Знаменатель: `W = ReduceRows(+, Map(f,cards_subgraph))`

Получение матрицы весов по данным о переводах

- Преобразовываем структуры (метки рёбер) в веса: $f(w) = \frac{w.Sum}{w.Count * 1000}$
- Сумма весов исходящих рёбер равна 1: $SoftMax(\overrightarrow{z})_i = \frac{e^{f(z_i)}}{\sum\limits_{k=1}^{K} e^{f(z_k)}}$
- Знаменатель: `W = ReduceRows(+, Map(f,cards_subgraph))`
- Элементы `W` на нужных местах: `D = Mask(cards_subgraph,` $M_W \times M_1$`)`

$$\texttt{Mask}\left(\begin{pmatrix} 0 & c_0 & 0 \\ c_2 & 0 & c_1 \\ c_3 & 0 & 0 \end{pmatrix}, \overbrace{\begin{pmatrix} w_0 & 0 & 0 \\ w_1 & 0 & 0 \\ w_2 & 0 & 0 \end{pmatrix}}^{W} \times \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\right) = \begin{pmatrix} 0 & w_0 & 0 \\ w_1 & 0 & w_1 \\ w_2 & 0 & 0 \end{pmatrix}$$

Получение матрицы весов по данным о переводах

- Преобразовываем структуры (метки рёбер) в веса: $f(w) = \frac{w.Sum}{w.Count*1000}$
- Сумма весов исходящих рёбер равна 1: $SoftMax(\overrightarrow{z})_i = \frac{e^{f(z_i)}}{\sum\limits_{k=1}^{K} e^{f(z_k)}}$
- Знаменатель: `W = ReduceRows(+, Map(f,cards_subgraph))`
- Элементы `W` на нужных местах: `D = Mask(cards_subgraph,` $M_W \times M_1$`)`

$$\texttt{Mask}\left(\begin{pmatrix} 0 & c_0 & 0 \\ c_2 & 0 & c_1 \\ c_3 & 0 & 0 \end{pmatrix}, \overbrace{\begin{pmatrix} w_0 & 0 & 0 \\ w_1 & 0 & 0 \\ w_2 & 0 & 0 \end{pmatrix}}^{W} \times \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\right) = \begin{pmatrix} 0 & w_0 & 0 \\ w_1 & 0 & w_1 \\ w_2 & 0 & 0 \end{pmatrix}$$

- Матрица весов $\mathcal{W}$ — поэлементное деление матриц: `Map(f,cards_subgraph)/D`

PageRank

- Что: ранжирование вершин по их «важности»
  - Оригинальная идея: вероятность попасть в вершину при случайном блуждании по графу

PageRank

- Что: ранжирование вершин по их «важности»
  - Оригинальная идея: вероятность попасть в вершину при случайном блуждании по графу
- Зачем (например): выделение «подозрительных» вершин в графе транзакций

PageRank

- Что: ранжирование вершин по их «важности»
  - Оригинальная идея: вероятность попасть в вершину при случайном блуждании по графу
- Зачем (например): выделение «подозрительных» вершин в графе транзакций
- Как: по определению
  - Матрица весов рёбер: $\mathcal{W}$
  - Вектор (столбец) весов вершин: v
  - Итерируем $v = \mathcal{W} \times v$ пока изменения v значимы

Линейная алгебра и языки запросов

- FalkorDB и Cypher
  - FalkorDB — графовая БД на основе линейной алгебры (SuiteSparse:GraphBLAS)
  - Поддерживает подмножество Cypher
  - Транслятор подмножества Cypher в операции линейной алгебры

Линейная алгебра и языки запросов

- FalkorDB и Cypher
  - FalkorDB — графовая БД на основе линейной алгебры (SuiteSparse:GraphBLAS)
  - Поддерживает подмножество Cypher
  - Транслятор подмножества Cypher в операции линейной алгебры
- Разное про SQL
  - TenSQL: An SQL Database Built on GraphBLAS
  - TCUDB: Accelerating Database with Tensor Processors
  - ...

Линейная алгебра и языки запросов

- FalkorDB и Cypher
  - FalkorDB — графовая БД на основе линейной алгебры (SuiteSparse:GraphBLAS)
  - Поддерживает подмножество Cypher
  - Транслятор подмножества Cypher в операции линейной алгебры
- Разное про SQL
  - TenSQL: An SQL Database Built on GraphBLAS
  - TCUDB: Accelerating Database with Tensor Processors
  - ...
- Линейная алгебра как основа для анализа данных
  - Towards a linear algebra semantics for SQL
  - Fast Join Project Query Evaluation using Matrix Multiplication
  - Fast Matrix Multiplication for Query Processing
  - A linear algebra approach to OLAP
  - Expressive Power of Linear Algebra Query Languages
  - On the expressiveness of Lara: A proposal for unifying linear and relational algebra

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