

Синтаксический анализ графов через умножение матриц

В рамках проекта лаборатории JetBrains

Автор: Рустам Шухратуллович Азимов, 646 группа

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

Рецензент: преподаватель, научный координатор Академии Або
и Центра Компьютерных Наук TUCS М.Л. Бараш

Санкт-Петербургский Государственный Университет
Кафедра системного программирования

28 марта 2018г.

Синтаксический анализ графов

- Вход:
 - ▶ Ориентированный граф $D = (V, E)$ с метками на ребрах из алфавита Σ
 - ▶ КС-грамматика (запрос к графу) $G = (\Sigma, N, P)$ над тем же алфавитом
- Выход для реляционной семантики запросов:
 - ▶ Множество всех троек (A, m, n) , где существует путь из вершины m в вершину n , метки на ребрах которого образуют строку, выводимую из нетерминала A
- Выход для single-path семантики запросов:
 - ▶ Дополнительно предоставить один такой путь для каждой тройки (A, m, n)

Пример

- 0 : $S \rightarrow subClassOf^{-1} S subClassOf$
- 1 : $S \rightarrow type^{-1} S type$
- 2 : $S \rightarrow subClassOf^{-1} subClassOf$
- 3 : $S \rightarrow type^{-1} type$

Рис.: Пример входной грамматики

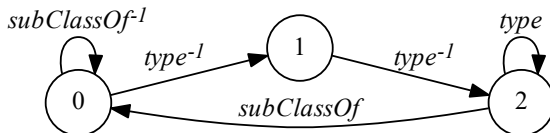


Рис.: Пример входного графа

- Запросы к графовым базам данных
- Анализ RDF файлов
- Биоинформатика
- Статический анализ программ

- Реляционная семантика запросов
 - ▶ Основанный на CYK (J. Hellings, 2014)
 - ▶ Тот же алгоритм, реализованный для анализа RDF файлов (X. Zhang, Z. Feng, X. Wang et al., 2016)
- Single-path семантика запросов
 - ▶ Основанный на методе динамического программирования (J. Hellings, 2015)
 - ▶ Основанный на GLL (Григорьев Семен, Рогозина Анастасия, 2016)

- Низкая производительность на больших графах
- Существующие алгоритмы не позволяют эффективно применить такие техники, как вычисление на графическом процессоре, параллельное вычисление
- Возможность создания матричного алгоритма синтаксического анализа графов является открытой проблемой

Постановка задачи

Цель: Разработать эффективный матричный алгоритм синтаксического анализа графов

Задачи:

- Разработать матричный алгоритм синтаксического анализа графов для реляционной семантики запросов
- Разработать матричный алгоритм синтаксического анализа графов для single-path семантики запросов
- Показать практическую применимость предложенных алгоритмов на общепринятом наборе данных

- Предложены алгоритмы синтаксического анализа графов, вычисляющие матричное транзитивное замыкание
- Доказана корректность предложенных алгоритмов
- Алгоритмы реализованы с использованием комбинаций таких оптимизаций, как:
 - ▶ Разреженное представление матриц
 - ▶ Умножение матриц на графическом процессоре
 - ▶ Параллельное умножение матриц
- Проведена апробация на общепринятом наборе RDF файлов

Пример работы алгоритма

0 : $S \rightarrow subClassOf^{-1} S subClassOf$
1 : $S \rightarrow type^{-1} S type$
2 : $S \rightarrow subClassOf^{-1} subClassOf$
3 : $S \rightarrow type^{-1} type$

Рис.: Пример входной грамматики

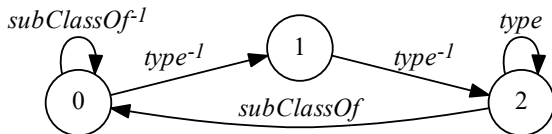


Рис.: Пример входного графа

Пример: Грамматика в нормальной форме

0 : $S \rightarrow S_1 S_5$
1 : $S \rightarrow S_3 S_6$
2 : $S \rightarrow S_1 S_2$
3 : $S \rightarrow S_3 S_4$
4 : $S_5 \rightarrow S S_2$
5 : $S_6 \rightarrow S S_4$
6 : $S_1 \rightarrow \text{subClassOf}^{-1}$
7 : $S_2 \rightarrow \text{subClassOf}$
8 : $S_3 \rightarrow \text{type}^{-1}$
9 : $S_4 \rightarrow \text{type}$

Рис.: Входная грамматика в нормальной форме Хомского

Пример: Начальная матрица и первая итерация

$$T_0 = \begin{pmatrix} \{S_1\} & \{S_3\} & \emptyset \\ \emptyset & \emptyset & \{S_3\} \\ \{S_2\} & \emptyset & \{S_4\} \end{pmatrix}$$

Рис.: Начальная матрица

$$T_0 \cdot T_0 = \begin{pmatrix} \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \{S\} \\ \emptyset & \emptyset & \emptyset \end{pmatrix}$$

$$T_1 = T_0 \cup (T_0 \cdot T_0) = \begin{pmatrix} \{S_1\} & \{S_3\} & \emptyset \\ \emptyset & \emptyset & \{S_3, S\} \\ \{S_2\} & \emptyset & \{S_4\} \end{pmatrix}$$

Пример: Остальные итерации

$$T_2 = \begin{pmatrix} \{S_1\} & \{S_3\} & \emptyset \\ \{S_5\} & \emptyset & \{S_3, S, S_6\} \\ \{S_2\} & \emptyset & \{S_4\} \end{pmatrix}$$

$$T_3 = \begin{pmatrix} \{S_1\} & \{S_3\} & \{S\} \\ \{S_5\} & \emptyset & \{S_3, S, S_6\} \\ \{S_2\} & \emptyset & \{S_4\} \end{pmatrix}$$

$$T_4 = \begin{pmatrix} \{S_1, S_5\} & \{S_3\} & \{S, S_6\} \\ \{S_5\} & \emptyset & \{S_3, S, S_6\} \\ \{S_2\} & \emptyset & \{S_4\} \end{pmatrix}$$

$$T_5 = \begin{pmatrix} \{S_1, S_5, S\} & \{S_3\} & \{S, S_6\} \\ \{S_5\} & \emptyset & \{S_3, S, S_6\} \\ \{S_2\} & \emptyset & \{S_4\} \end{pmatrix}$$

Пример: Результирующие отношения по матрице

$T_6 = T_5$

$$R_S = \{(0, 0), (0, 2), (1, 2)\},$$

$$R_{S_1} = \{(0, 0)\},$$

$$R_{S_2} = \{(2, 0)\},$$

$$R_{S_3} = \{(0, 1), (1, 2)\},$$

$$R_{S_4} = \{(2, 2)\},$$

$$R_{S_5} = \{(0, 0), (1, 0)\},$$

$$R_{S_6} = \{(0, 2), (1, 2)\}.$$

Рис.: Результирующие КС-отношения

Апробация: Запрос 1

0 : $S \rightarrow subClassOf^{-1} S subClassOf$
1 : $S \rightarrow type^{-1} S type$
2 : $S \rightarrow subClassOf^{-1} subClassOf$
3 : $S \rightarrow type^{-1} type$

Рис.: Грамматика для запроса 1

Апробация: результаты для запроса 1

Ontology	edgs	result	GLL	dGPU	sCPU	sGPU
skos	252	810	10	56	14	12
generations	273	2164	19	62	20	13
travel	277	2499	24	69	22	30
univ-bench	293	2540	25	81	25	15
atom-primitive	425	15454	255	190	92	22
biomedical	459	15156	261	266	113	20
foaf	631	4118	39	154	48	9
people-pets	640	9472	89	392	142	32
funding	1086	17634	212	1410	447	36
wine	1839	66572	819	2047	797	54
pizza	1980	56195	697	1104	430	24
g_1	8688	141072	1926	—	26957	82
g_2	14712	532576	6246	—	46809	185
g_3	15840	449560	7014	—	24967	127

Апробация: Запрос 2

0 : $S \rightarrow B \text{ subClassOf}$
1 : $S \rightarrow \text{subClassOf}$
2 : $B \rightarrow \text{subClassOf}^{-1} B \text{ subClassOf}$
3 : $B \rightarrow \text{subClassOf}^{-1} \text{subClassOf}$

Рис.: Грамматика для запроса 2

Апробация: результаты для запроса 2

Ontology	edgs	result	GLL	dGPU	sCPU	sGPU
skos	252	1	1	10	2	1
generations	273	0	1	9	2	0
travel	277	63	1	31	7	10
univ-bench	293	81	11	55	15	9
atom-primitive	425	122	66	36	9	2
biomedical	459	2871	45	276	91	24
foaf	631	10	2	53	14	3
people-pets	640	37	3	144	38	6
funding	1086	1158	23	1246	344	27
wine	1839	133	8	722	179	6
pizza	1980	1262	29	943	258	23
g_1	8688	9264	167	—	21115	38
g_2	14712	1064	46	—	10874	21
g_3	15840	10096	393	—	15736	40

- Разработан матричный алгоритм синтаксического анализа графов для реляционной семантики запросов
- Разработан матричный алгоритм синтаксического анализа графов для single-path семантики запросов
- Показана практическая применимость предложенных алгоритмов на общепринятом наборе данных