



# Структурное представление результата поиска путей с контекстно-свободными ограничениями

Семён Григорьев

JetBrains Research, лаборатория языковых инструментов  
Санкт-Петербургский государственный университет

21.10.2017

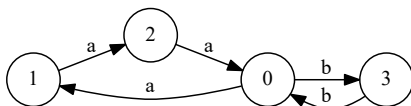
- Анализ графов
  - ▶ Запросы к графовым базам данных
  - ▶ Анализ сетей (социальных, интернет и т.д.)
- Статический анализ программ
  - ▶ Анализ алиасов
  - ▶ Taint analysis
  - ▶ Статический анализ динамически формируемого кода
- ...

# Поиск путей с контекстно-свободными ограничениями

- $\mathbb{G} = (\Sigma, N, P)$  — контекстно-свободная грамматика
- $p = v_0 \xrightarrow{l_0} v_1 \xrightarrow{l_1} \dots v_{n-1} \xrightarrow{l_{n-1}} v_n$  — путь в графе  $G$
- $w(p) = w(v_0 \xrightarrow{l_0} v_1 \xrightarrow{l_1} \dots v_{n-1} \xrightarrow{l_{n-1}} v_n) = l_0 l_1 \dots l_{n-1}$
- $G = (V, E, L)$  — ориентированный граф,  $E \subseteq V \times L \times V$ ,  $L \subseteq \Sigma$
- $R = \{p \mid \exists N_i \in N (w(p) \in L(\mathbb{G}, N_i))\}$ 
  - ▶ Стартовый нетерминал можно зафиксировать заранее
  - ▶ **Проблема:** множество  $R$  может быть бесконечным

## Пример

Входной граф



Запрос — грамматика  $G$  для языка  $L = \{a^n b^n; n \geq 1\}$  с явным выделением середины пути

0 :  $S \rightarrow a S b$

1 :  $S \rightarrow Middle$

2 :  $Middle \rightarrow a b$

Ответ — бесконечное множество путей

•  $p_1 = 0 \xrightarrow{a} 1 \xrightarrow{a} 2 \xrightarrow{a} 0 \xrightarrow{b} 3 \xrightarrow{b} 0 \xrightarrow{b} 3$

•  $p_2 = 0 \xrightarrow{a} 1 \xrightarrow{a} 2 \xrightarrow{a} 0 \xrightarrow{a} 1 \xrightarrow{a} 2 \xrightarrow{a} 0 \xrightarrow{b} 3 \xrightarrow{b} 0 \xrightarrow{b} 3 \xrightarrow{b} 0 \xrightarrow{b} 3 \xrightarrow{b} 0 \xrightarrow{b} 3 \xrightarrow{b} 0$

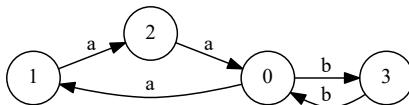
• ...

- В качестве основы используют алгоритм СΥК
  - ▶ Необходимо преобразовывать грамматику в нормальную форму Хомского
  - ▶ Только проверка наличия пути (*Zhang X. et al.* “Context-free path queries on RDF graphs.” 2016)
  - ▶ Грамматика в качестве представления результата (*Hellings J.* “Conjunctive context-free path queries.” 2014)

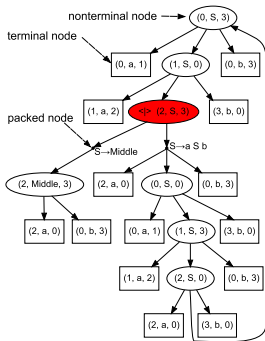
# Алгоритм выполнения КС запросов к графам

- Основан на обобщённом LL (Generalized GLL, GLL)
  - ▶ *Scott E., Johnstone A. “GLL parsing”*
- Поддерживает произвольные контекстно-свободные грамматики (неоднозначные, леворекурсивные)
- Не требует преобразования грамматики в нормальную форму Хомского
- Строит сжатое представление леса разбора (Sharep Packed Parse Forest, SPPF) — конечное представление бесконечного ответа

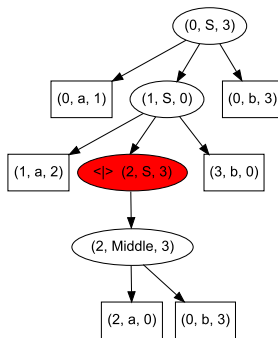
# Структурное представление результата запроса



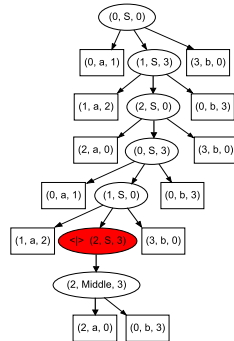
Входной граф



Результат (SPPF)

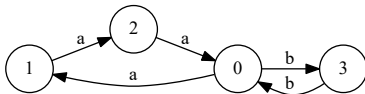


Дерево вывода пути  $p_1$



Дерево вывода пути  $p_2$

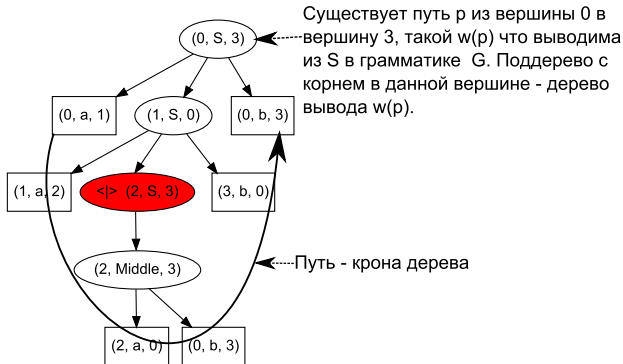
## Пример: извлечение путей



0 :  $S \rightarrow a S b$

1 :  $S \rightarrow \textit{Middle}$

2 :  $\textit{Middle} \rightarrow a b$

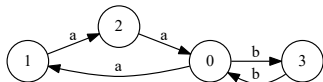


Путь:  $0 \xrightarrow{a} 1 \xrightarrow{a} 2 \xrightarrow{a} 0 \xrightarrow{b} 3 \xrightarrow{b} 0 \xrightarrow{b} 3$



# Пример: почему это работает

Замкнутость КС языков относительно пересечения с регулярными



Регулярный язык

0 :  $S \rightarrow a S b$

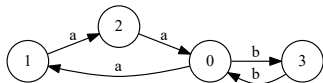
1 :  $S \rightarrow \textit{Middle}$

2 :  $\textit{Middle} \rightarrow a b$

Контекстно-свободный язык

# Пример: почему это работает

Замкнутость КС языков относительно пересечения с регулярными



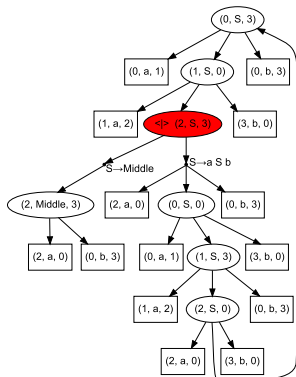
Регулярный язык

0 :  $S \rightarrow a S b$

1 :  $S \rightarrow \text{Middle}$

2 :  $\text{Middle} \rightarrow a b$

Контекстно-свободный язык



$(0, S, 3) \rightarrow (0, a, 1) (1, S, 0) (0, b, 3)$

$(1, S, 0) \rightarrow (1, a, 2) (2, S, 3) (3, b, 0)$

$(2, S, 3) \rightarrow (2, a, 0) (0, S, 0) (0, b, 3)$

$(2, S, 3) \rightarrow (2, \text{Middle}, 3)$

$(0, S, 0) \rightarrow (0, a, 1) (1, S, 3) (3, b, 0)$

$(1, S, 3) \rightarrow (1, a, 2) (2, S, 0) (0, b, 3)$

$(2, S, 0) \rightarrow (2, a, 0) (0, S, 3) (3, b, 0)$

$(0, \text{Middle}, 3) \rightarrow (2, a, 0) (0, b, 3)$

# Характеристики алгоритма

Пусть на входе граф  $M = (V, E, L)$ , тогда

- Пространственная сложность предложенного алгоритма  
 $O(|V|^3 + |E|)$
- Временная сложность предложенного алгоритма  
 $O\left(|V|^3 * \max_{v \in V} (deg^+(v))\right)$
- Результирующий SPPF имеет размер  $O(|V'|^3 + |E'|)$ , где  $M' = (V', E', L')$  — подграф  $M$ , содержащий только искомые пути

## Экспериментальное исследование: запросы

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

Грамматика для запроса Query 1

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

Грамматика для запроса Query 2

## Экспериментальное исследование: результаты

Ontology	#edg	Query 1			Query 2	
		time CYK (ms)	time (ms)	#result	time (ms)	#result
skos	252	1044	10	810	1	1
generations	273	6091	19	2164	1	0
travel	277	13971	24	2499	1	63
univ-bench	293	20981	25	2540	11	81
foaf	631	–	39	4118	2	10
people-pets	640	82081	89	9472	3	37
funding	1086	–	212	17634	23	1158
atom-primitive	425	515285	255	15454	66	122
biomedical- measure-primitive	459	420604	261	15156	45	2871
pizza	1980	3233587	697	56195	29	1262
wine	1839	4075319	819	66572	8	133

- Почта: `semen.grigorev@jetbrains.com`
- GitHub-сообщество YaccConstructor:  
`https://github.com/YaccConstructor`