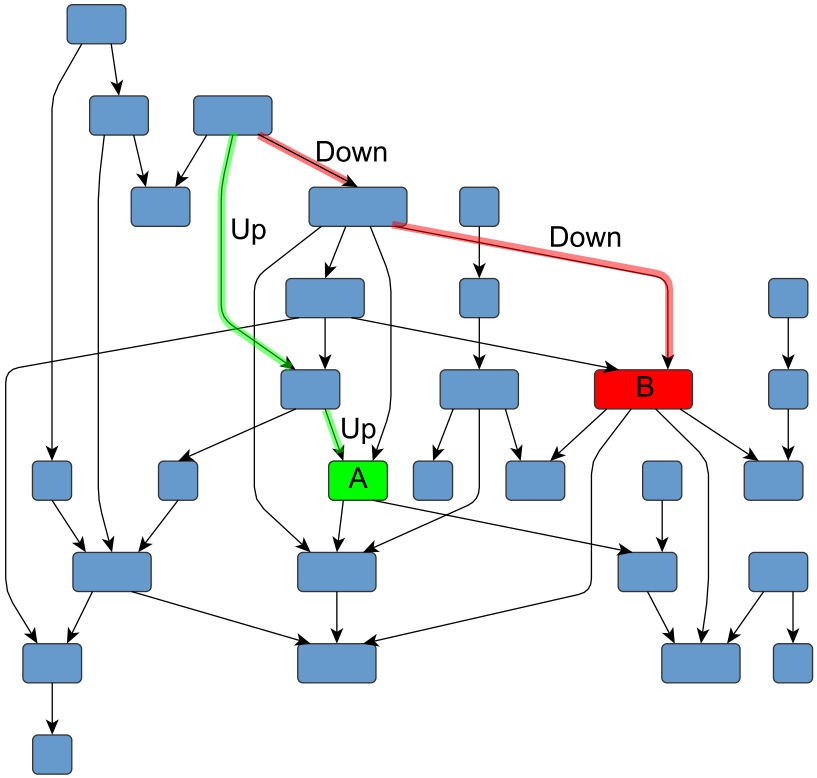
Parsing techniques for graph analysis

Semyon Grigorev, **Kate Verbitskaia**

JetBrains Research, Programming Languages and Tools Lab
Saint Petersburg University

October 22, 2017

Language-constrained paths filtering



Navigation through a graph

- Are nodes A and B on the same level of hierarchy?
- Is there a path of form $\mathbf{Up}^n\,\mathbf{Down}^n$?
- Find all paths of form $\mathbf{Up}^n\,\mathbf{Down}^n$ which start from the node A

- (How) Can an automaton generate phrases in some specific (context-free) language?
- (How) Can a program produce some specific chain of the subprogram calls?

Language-constrained paths filtering: more formal

- $\mathbb{G} = (\Sigma, N, P)$ — context-free grammar
- $G = (V, E, L)$ — directed graph
  - $v \xrightarrow{l} u \in E$
  - $L \subseteq \Sigma$
- $\omega(p) = \omega(v_0 \xrightarrow{l_0} v_1 \xrightarrow{l_1} \cdots \xrightarrow{l_{n-2}} v_{n-1} \xrightarrow{l_{n-1}} v_n) = l_0 l_1 \cdots l_{n-1}$
- $R = \{p \mid \text{exists } N_i \in N \text{ such that } \omega(p) \in L(\mathbb{G}, N_i)\}$

Applications

- Graph database querying (Yannakakis. 1990; Hellings. 2014; Zhang. 2016)
- Code analysis
  - Static analysis via context-free and linear conjunctive language reachability
    - ★ alias analysis (Zhang, Su. 2017)
    - ★ points-to analysis (Xu, Rountev, Sridharan. 2009)
  - Dynamically generated strings analysis (Verbitskaia, Grigorev, Avdyukhin. 2015)
  - Multiple input parsing (Scott, Johnstone. 2016)
- ...

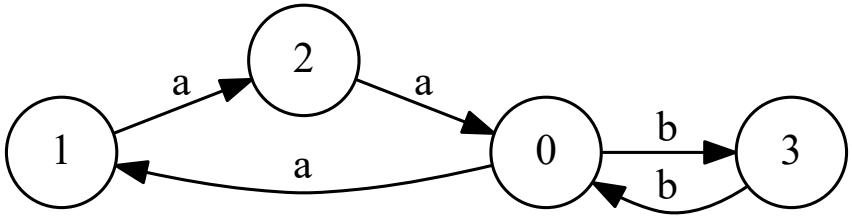
Existing approaches

- Do not use the power of advanced parsing techniques
  - Are mostly based on CYK
    (Zhang, et al. “Context-free path queries on RDF graphs.”;
    Hellings. “Conjunctive context-free path queries.”)
  - Do not provide useful structural representation of result
- Impose restrictions on input
  - Do not process input graphs with cycles
    (Sevon, Eronen. “Subgraph queries by context-free grammars.”)
  - Are restricted to certain grammar classes

Open problems

- Development of efficient algorithms
- Result representation for query debugging and further processing
- Processing of wider (??) types of grammars (ECFG, conjunctive, etc)

Example

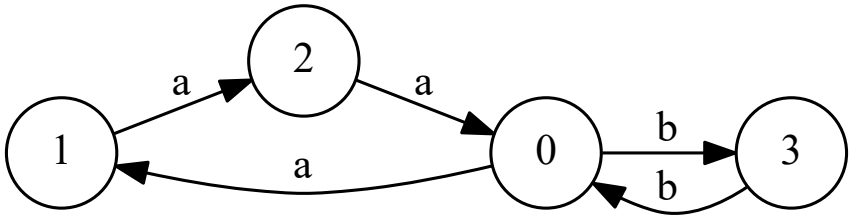


Input graph

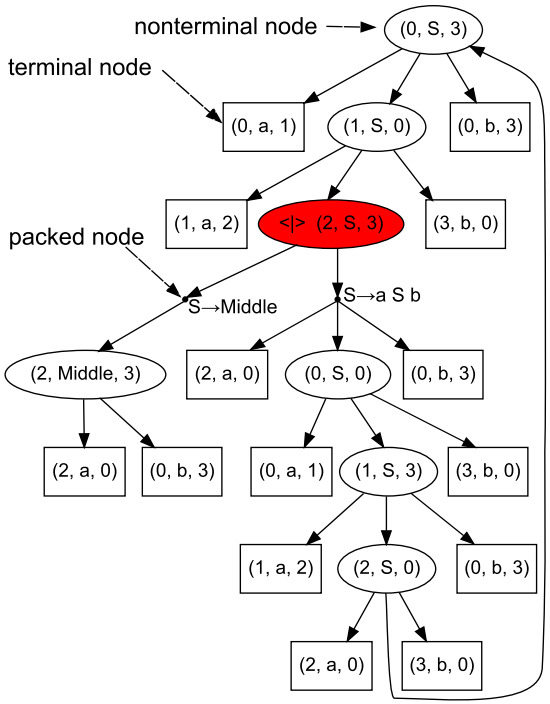
$$
\begin{aligned}
S &\rightarrow a\ S\ b \\
S &\rightarrow \mathit{Middle} \\
\mathit{Middle} &\rightarrow a\ b
\end{aligned}
$$

Query: a grammar for the language $L = \{a^n b^n \mid n \geq 1\}$ with an additional marker for the middle of the path

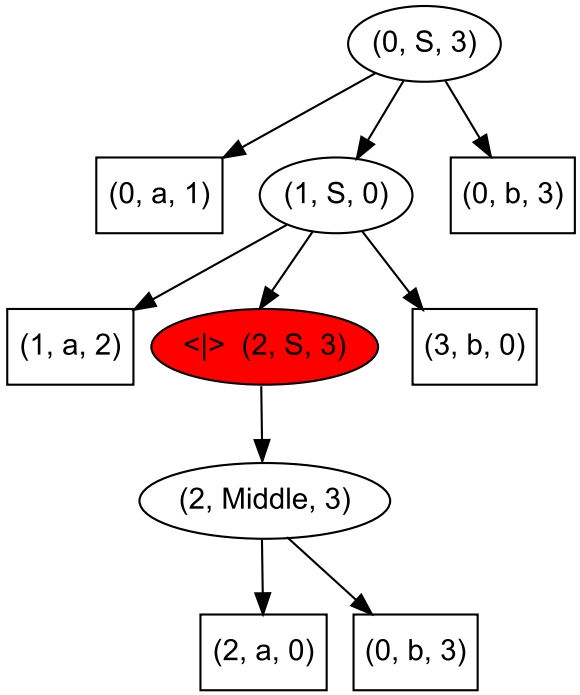
Example



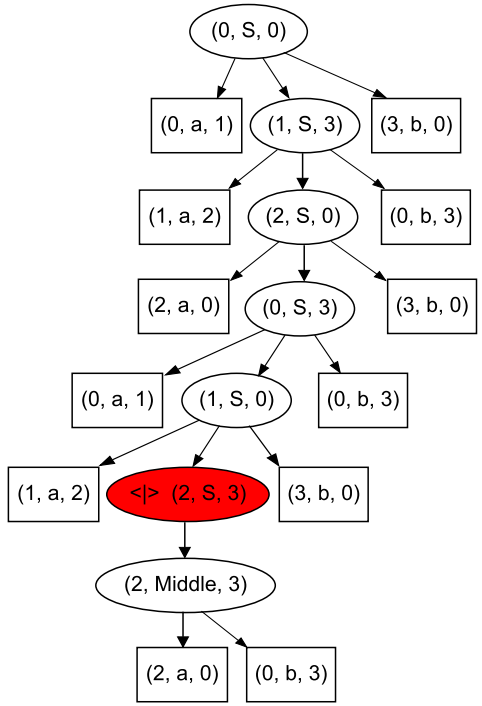
Input graph



Query result: SPPF

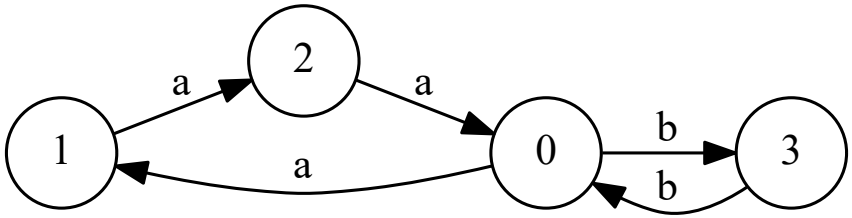


Tree for the path $0 \rightsquigarrow 3$

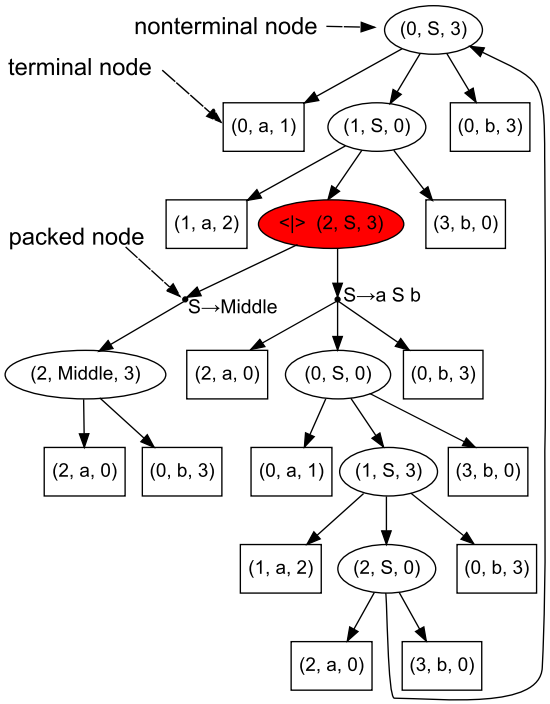


Tree for the path $0 \rightsquigarrow 0$

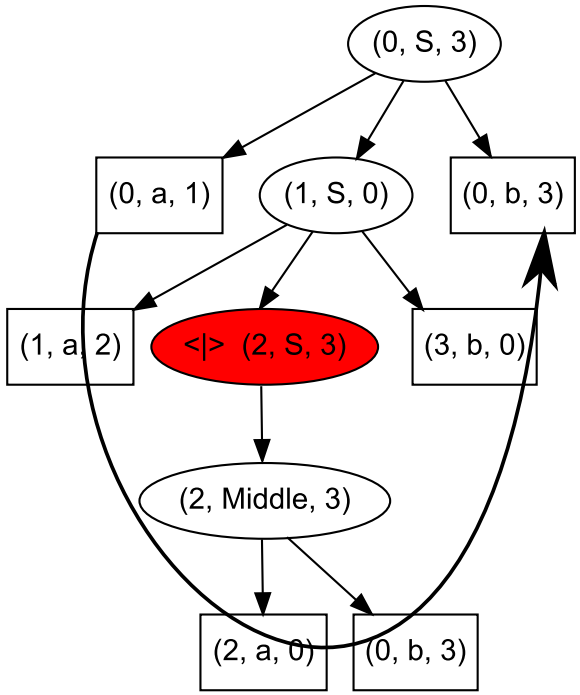
Example



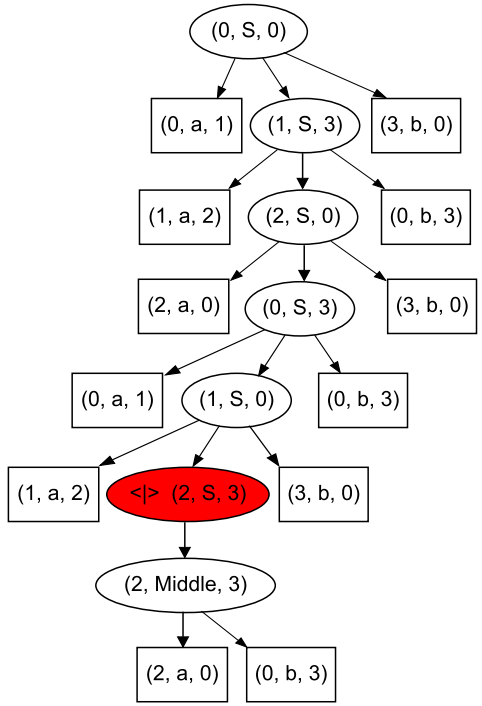
Input graph



Query result: SPPF



Tree for the path $0 \rightsquigarrow 3$



Tree for the path $0 \rightsquigarrow 0$

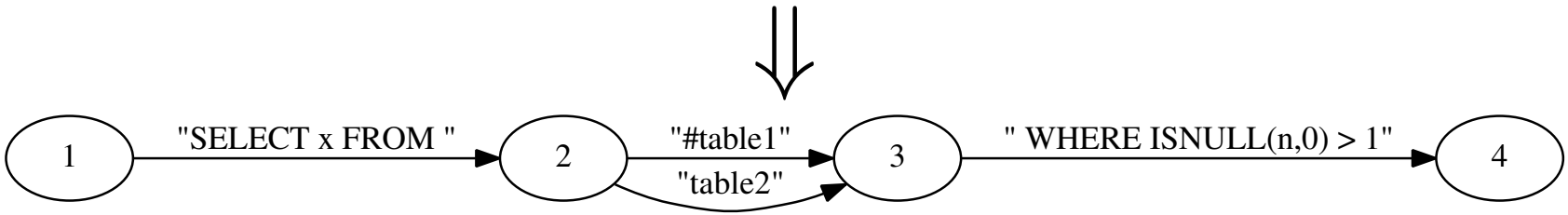
Our solutions

- Relaxed parsing of dynamically generated SQL-queries
- Context-free path querying with structural representation of result
- Context-free path querying by matrix multiplication
- Parser combinators for context-free path querying

Relaxed parsing of dynamically generated SQL-queries

- Verbitskaia, Grigorev, Avdyukhin. 2015
- Based on RNGLR parsing algorithm (Scott, Johnstone)
- Static analysis of dynamically generated (SQL)code

```
IF @X = @Y SET @TABLE = '#table1'
ELSE       SET @TABLE = 'table2'
EXECUTE
     ('SELECT x FROM ' + @TABLE + ' WHERE ISNULL(n,0) > 1')
```



Context-free path querying with structural representation of result

- Grigorev, Ragozina. 2016
- Based on GLL parsing algorithm (Scott, Johnstone)
- General-purpose context-free path querying algorithm which can produce user-friendly representation of result
- Worst-case space complexity

$$O(|V|^3 + |E|)$$

- Worst-case time complexity

$$O\left(|V|^3 * \max_{v \in V}\left(deg^+(v)\right)\right)$$

Context-free path querying by matrix multiplication

- Azimov, Grigorev. 2017
- Inspired by works of Valiant and Okhotin
- GPGPU utilization for context-free path querying
- Worst-case time complexity

$$O(|V|^2|N|^3(BMM(|V|) + BMU(|V|)))$$

  - BMM — Boolean Matrix Multiplication
  - BMU — Boolean Matrix Union

Parser combinators for context-free path querying

- Smolina, Verbitskaia. 2017
- Based on the Meerkat: a general parser combinator library for Scala (Afroozeh, Izmaylova)
- Context-free path querying without DSLs
  - May be more friendly for static code analysis tools developers
  - Data-dependent parsing (in progress)

Evaluation: data

- Graphs — the set of ontologies
- Query is classical "same-generation query"

$$\mathbf{S} \to \mathit{subClassOf}^{-1}\ \mathbf{S}\ \mathit{subClassOf}$$
$$\mathbf{S} \to \mathit{type}^{-1}\ \mathbf{S}\ \mathit{type}$$
$$\mathbf{S} \to \mathit{subClassOf}^{-1}\ \mathit{subClassOf}$$
$$\mathbf{S} \to \mathit{type}^{-1}\ \mathit{type}$$

Evaluation: results

Ontology	#edg	time (ms)		
		CYK[1]	GLL	Matrix
skos	252	1044	10	12
generations	273	6091	19	13
travel	277	13971	24	30
univ-bench	293	20981	25	15
people-pets	640	82081	89	32
atom-primitive	425	515285	255	22
biomedical-measure-primitive	459	420604	261	20
pizza	1980	3233587	697	24
wine	1839	4075319	819	54
g1	8688	—	1926	82
g2	14712	—	6246	185
g3	15840	—	7014	127

[1] Zhang, et al. "Context-free path queries on RDF graphs."

Future work: Other grammars and language classes intersection

- Context-free grammars intersection: Nederhof, "The language intersection problem for non-recursive context-free grammars"
  - Compressed strings processing
  - Grammar-compressed graphs querying
- Approximated intersection of regular and conjunctive/boolean languages
  - More expressive query languages
- ...

Future work: Mechanization in Coq

- Bar-Hillel theorem
- GLL-based algorithms
- Other algorithms for grammars intersection

- Basic (parsing) algorithms verification
- Base for complex algorithms verification

Contact information

- Semyon Grigorev: semen.grigorev@jetbrains.com
- Kate Verbitskaia: ekaterina.verbitskaya@jetbrains.com

- YaccConstructor: https://github.com/YaccConstructor