



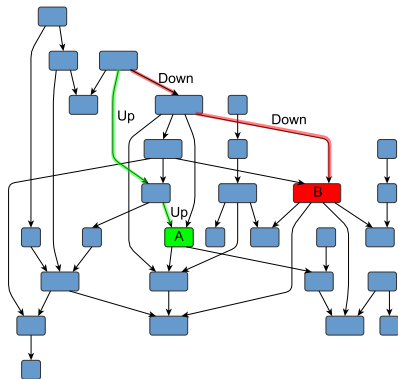
## Parsing techniques for graph analysis

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  **$Up^n Down^n$** ?
- Find all paths of form  **$Up^n Down^n$**  which start from a node A.

- (How) Can this automaton generate phrases in some specific (context-free) language?
- (How) Can this program produce some specific chain of subprograms calls?

# Language-constrained paths filtering: more formal

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

- Graph analysis
  - ▶ Graph database querying
  - ▶ Network graph analysis
- Code analysis
  - ▶ Static analysis CFL(linear conjunctive) reachability
    - ★ alias analysis
    - ★ points-to analysis
  - ▶ Dynamically generated strings analysis
  - ▶ Multiple input parsing
- ...

- Do not use the power of advanced parsing techniques
  - ▶ Mostly based on CYK  
(Xiaowang Zhang, et al. “Context-free path queries on RDF graphs.”;  
Jelle Hellings. “Conjunctive context-free path queries.” )
  - ▶ Do not provide useful structural representation of result
- Impose restrictions on input
  - ▶ Problems with cycles in the input graph  
(Petteri Sevon, Lauri Eronen. “Subgraph queries by context-free grammars.”)

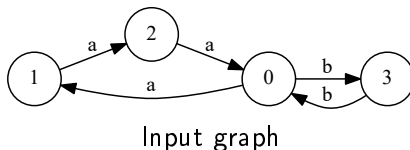
# Open problems

- Development of effective algorithms
- Result representation for debugging; further processing
- GPGPU utilization
- Processing of different types of grammars (ECFG, conjunctive, etc)

# Bar-Hillel theorem

- Context-free languages are closed under intersection with regular languages
- Parsing algorithms are constructive proof of Bar-Hillel theorem for one simple case ...
- ... so, classical parsing can be generalized for arbitrary regular language processing

## Example



0 :  $S \rightarrow a S b$

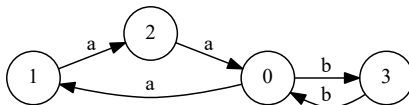
1 :  $S \rightarrow \textit{Middle}$

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

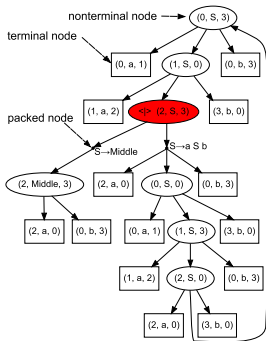
Query: a grammar for the language  $L = \{a^n b^n; n \geq 1\}$  with an additional marker for the middle of a path



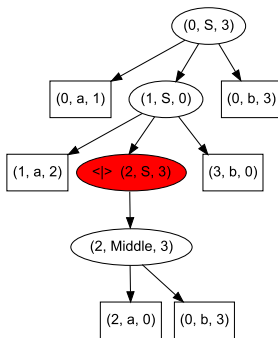
# Example



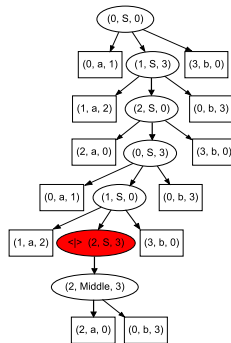
Input graph



Query result: SPPF

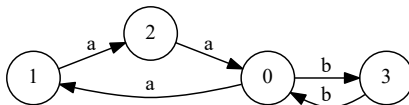


Tree for path from 0 to 3

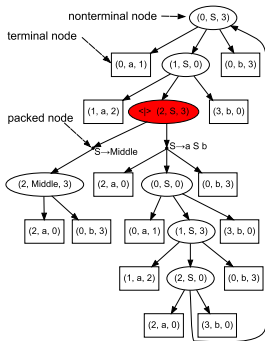


Tree for path from 0 to 0

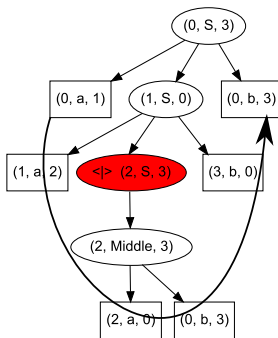
# Example



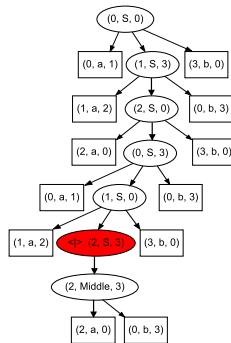
Input graph



Query result: SPPF



Tree for path from 0 to 3



Tree for path from 0 to 0

- Relaxed parsing of dynamically generated SQL-queries
  - ▶ Based on RNLGR parsing algorithm (Elizabeth Scott, Adrian Johnstone)
- Context-free path querying with structural representation of result
  - ▶ Based on GLL parsing algorithm (Elizabeth Scott, Adrian Johnstone)
- Combinators for context-free path querying
  - ▶ Based on the Meerkat: a general parser combinator library for Scala (Ali Afroozeh, Anastasia Izmaylova)
- Context-free path querying by matrix multiplication
  - ▶ Inspired by Leslie Valiant and Alexander Okhotin

- Other grammars and language classes intersection
  - ▶ Context-free grammars intersection: Mark-Jan Nederhof, “The language intersection problem for non-recursive context-free grammars”
  - ▶ Approximated intersection of regular and conjunctive/boolean languages
  - ▶ ...
- Mechanization in Coq
  - ▶ Bar-Hillel theorem
  - ▶ GLL-based algorithms
  - ▶ ...
- New areas for application

# Contact information

- Semyon Grigorev: [semen.grigorev@jetbrains.com](mailto:semen.grigorev@jetbrains.com)
- Kate Verbitskaia: [kajigor@gmail.com](mailto:kajigor@gmail.com)
- YaccConstructor: <https://github.com/YaccConstructor>