

# Parser-Combinators for Context-Free Path Querying\*

Smolina

Saint Petersburg State University  
St. Petersburg, Russia  
trovato@corporation.com

Ilia Kirillov

Saint Petersburg State University  
St. Petersburg, Russia  
larst@affiliation.org

Ekaterina Verbitskaia

Saint Petersburg State University  
St. Petersburg, Russia  
webmaster@marysville-ohio.com

Semyon Grigorev

Saint Petersburg State University  
St. Petersburg, Russia  
semen.grigorev@jetbrains.com

## ABSTRACT

[illegible]

## CCS CONCEPTS

- **Information systems** → Graph-based database models; Query languages for non-relational engines;
- **Theory of computation** → *Grammars and context-free languages*;

## KEYWORDS

Graph data bases, Language-constrained path problem, Context-Free path querying, Parser Combinators, Generalized LL, GLL, Neo4J, Scala

### ACM Reference Format:

Smolina, Ekaterina Verbitskaia, Ilia Kirillov, and Semyon Grigorev. 2018. Parser-Combinators for Context-Free Path Querying. In *Proceedings of Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA) 2018 (GRADES-NDA'18)*. ACM, New York, NY, USA, Article 4, 2 pages. [https://doi.org/10.475/123\\_4](https://doi.org/10.475/123_4)

---

\*Produces the permission block, and copyright information

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GRADES-NDA'18, June 2018, Houston, Texas USA

© 2018 Copyright held by the owner/author(s).

ACM ISBN 123-4567-24-567/08/06...\$15.00

[https://doi.org/10.475/123\\_4](https://doi.org/10.475/123_4)

## 1 INTRODUCTION

Graph as data model, Graph data bases.

Navigation queries. Path querying and context-free path querying. Same generation query is not a regular. Static code analysis.

Integration with general purpose programming languages is a problem. String-embedded DSLs. Special DSL vs Combinators (LINQ [3, 8], etc) [7]

We propose !!! and we make the following contributions in this paper.

- (1) Combinators for CF path querying with structural representation of result
- (2) Implementation in Scala. Generalization of linear parsing. Integration with Neo4J graph data base. Available on gitHub:<https://github.com/YaccConstructor/Meerkat>
- (3) Evaluation on realistic data, which shows that it is applicable. Comparison with other tools for CF path querying.

## 2 RELATED WORK

Language-constrained path querying, Yannakakis [14]. Hellings [4, 5], RDF [15], etc [1, 2, 9, 11, 13]

Special graph query languages. SPARQL, cypher

Language integration problem: special DSLs for SQL, ORM, Linq  
Parser-combinators is one of classical approach for parsing!!!

Scala combinators for graph [7] — one of attempt to adopt combinators technique for graph processing. But language class is not discussed.

Classical combinators has restrictions: left-recursive grammars. GLL [12] can handle arbitrary context-free grammars, SPPF [10]

Parser combinators library based on GLL – Meerkat<sup>1</sup> [6].

etc

### 3 PARSER-COMBITATORS FOR PATH QUERING

In this section we present our implementation of and describe some details.

Our implementation is based on Meerkat library. We need only some steps for generalization.

As far as linear input is a one of case of graph, it is possible to provide input abstraction which make possible to generalize combinators.

<sup>1</sup><https://github.com/meerkat-parser/Meerkat>

SPPF may be an arbitrary graph in opposite of linear input parsing.

Let us introduce an example. Graph. Grammar. In terms of combinators.

- Interface for Neo4J data base
- Extensible solution
- An architecture of the solution.

## 4 EVALUATION

Some experiments on real data and comparison with existing solutions

- Classical RDFs
- Integration with Neo4J
- Static code analysis
- Comparison with GLL
- Comparison with [7]

## 5 CONCLUSION

We propose a native way to integrate language for language-constrained path querying into general purpose programming language. We implement it and show that our implementation can be applied for real problems.

- Code is available on GitHub:
- Future work is
- SPPF processing for debugging and results processing
- Attributed grammars processing to provide mechanism for semantics calculation

## REFERENCES

- [1] Pablo Barceló, Gaëlle Fontaine, and Anthony Widjaja Lin. 2013. Expressive Path Queries on Graphs with Data. In *International Conference on Logic for Programming Artificial Intelligence and Reasoning*. Springer, 71–85.
- [2] Chris Barrett, Riko Jacob, and Madhav Marathe. 2000. Formal-language-constrained path problems. *SIAM J. Comput.* 30, 3 (2000), 809–837.
- [3] James Cheney, Sam Lindley, and Philip Wadler. 2013. A Practical Theory of Language-integrated Query. *SIGPLAN Not.* 48, 9 (Sept. 2013), 403–416. <https://doi.org/10.1145/2544174.2500586>
- [4] Jelle Hellings. 2014. Conjunctive context-free path queries. (2014).
- [5] Jelle Hellings. 2015. Path Results for Context-free Grammar Queries on Graphs. *CoRR* abs/1502.02242 (2015). <http://arxiv.org/abs/1502.02242>
- [6] Anastasia Izmaylova, Ali Afroozeh, and Tijs van der Storm. 2016. Practical, General Parser Combinators. In *Proceedings of the 2016 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation (PEPM '16)*. ACM, New York, NY, USA, 1–12. <https://doi.org/10.1145/2847538.2847539>
- [7] Daniel Kröni and Raphael Schweizer. 2013. Parsing Graphs: Applying Parser Combinators to Graph Traversals. In *Proceedings of the 4th Workshop on Scala (SCALA '13)*. ACM, New York, NY, USA, Article 7, 4 pages. <https://doi.org/10.1145/2489837.2489844>
- [8] Erik Meijer, Brian Beckman, and Gavin Bierman. 2006. LINQ: Reconciling Object, Relations and XML in the .NET Framework. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data (SIGMOD '06)*. ACM, New York, NY, USA, 706–706. <https://doi.org/10.1145/1142473.1142552>
- [9] A. Mendelzon and P. Wood. 1995. Finding Regular Simple Paths in Graph Databases. *SIAM J. Computing* 24, 6 (1995), 1235–1258.
- [10] Joan Gerard Rekers. 1992. *Parser generation for interactive environments*. Ph.D. Dissertation. Universiteit van Amsterdam.
- [11] Juan L Reutter, Miguel Romero, and Moshe Y Vardi. 2015. Regular queries on graph databases. *Theory of Computing Systems* (2015), 1–53.
- [12] Elizabeth Scott and Adrian Johnstone. 2010. GLL parsing. *Electronic Notes in Theoretical Computer Science* 253, 7 (2010), 177–189.
- [13] Petteri Sevon and Lauri Eronen. 2008. Subgraph queries by context-free grammars. *Journal of Integrative Bioinformatics* 5, 2 (2008), 100.
- [14] Mihalis Yannakakis. 1990. Graph-theoretic methods in database theory. In *Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. ACM, 230–242.
- [15] Xiaowang Zhang, Zhiyong Feng, Xin Wang, Guozheng Rao, and Wenrui Wu. 2016. Context-free path queries on RDF graphs. In *International Semantic Web Conference*. Springer, 632–648.