

Parser-Combinators for Context-Free Path Querying*

Smolina

Institute for Clarity in Documentation
Dublin, Ohio
trovato@corporation.com

Ilia Kirillov

The Thørvöld Group
Hekla, Iceland
larst@affiliation.org

Ekaterina Verbitskaia

Institute for Clarity in Documentation
Dublin, Ohio
webmaster@marysville-ohio.com

Semyon Grigorev

Institute for Clarity in Documentation
Dublin, Ohio
trovato@corporation.com

ABSTRACT

[illegible]

CCS CONCEPTS

- **Computer systems organization** → **Embedded systems**; *Redundancy*; **Robotics**; • **Networks** → Network reliability;

KEYWORDS

ACM proceedings, L^AT_EX, text tagging

ACM Reference Format:

Smolina, Ekaterina Verbitskaia, Ilia Kirillov, and Semyon Grigorev. 2018. Parser-Combinators for Context-Free Path Querying. In *Proceedings of ACM Woodstock conference (GRADES-DNA'18)*. ACM, New York, NY, USA, Article 4, 2 pages. <https://doi.org/10.475/123> 4

1 INTRODUCTION

Graph data bases

Path querying and context-free path querying. Same generation query is not a regular.

*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-DNA'18, July 2018, El Paso, 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

Integration with general purpose programming languages. Special DSL vs Combinators (LINQ, etc) [4]

Contribution:

- Combinators for CF path querying with structural representation of result
- Implementation in Scala. Generalization of linear parsing. Integration with Neo4J graph data base. Available on gitHub:<https://github.com/YaccConstructor/Meerkat>
- 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 [7]. Hellings [1, 2], RDF [8], etc

Special graph query languages. SPARQL, cypher

Language integration problem: special DSLs for SQL, ORM, Linq
Parser-combinators

Scala combinators for graph [4]

GLL [6] arbitrary context-free grammars, SPPF [5]

Meerkat ¹ [3] – combi

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.

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 bata base

Extensible solution

An architecture of the solution.

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

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 [4]

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] Jelle Hellings. 2014. Conjunctive context-free path queries. (2014).
- [2] Jelle Hellings. 2015. Path Results for Context-free Grammar Queries on Graphs. *CoRR* abs/1502.02242 (2015). <http://arxiv.org/abs/1502.02242>
- [3] 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>
- [4] 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>
- [5] Joan Gerard Rekers. 1992. *Parser generation for interactive environments*. Ph.D. Dissertation. Universiteit van Amsterdam.
- [6] Elizabeth Scott and Adrian Johnstone. 2010. GLL parsing. *Electronic Notes in Theoretical Computer Science* 253, 7 (2010), 177–189.
- [7] 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.
- [8] 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.