

# 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<sup>A</sup>T<sub>E</sub>X, text tagging

### ACM Reference Format:

Smolina, Ekaterina Verbitskaia, Ilia Kirillov, and Semyon Grigorev. 1997. Parser-Combinators for Context-Free Path Querying. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97)*, Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 2 pages. [https://doi.org/10.475/123\\_4](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.

Integration with general purpose programming languages. Special DSL vs Combinators (LINQ, etc) [2]

Contribution:

- Combinators for CF path querying with structural representation of result
- Implementation in Scala. Available on gitHub:<https://github.com/YaccConstructor/Meerkat>
- Evaluation on realistic data, which shows that it is applicable.

## 2 RELATED WORK

Hellings,etc

Scala combinators for graph [2]

GLL

Meerkat <sup>1</sup> [1]

etc

### 3 PARSER-COMBITATORS FOR PATH QUERING

Based on Meerkat.

SPPF

### Input abstraction

### Example

WTF???

## 4 EVALUATION

Classical RDFs

## Integration with Neo4J

## Static code analysis

### Comparison with GLL

### Comparison with [2]

## 5 CONCLUSION

We propose and show

Future work:

SPPF processing

Semantics calculation

---

\*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).

WOODSTOCK'97, July 1997, El Paso, Texas USA

© 2016 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)

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

## REFERENCES

- [1] 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>
- [2] 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>