The Journal of Open Source Software

LangToGroup: A Haskell package for group presentation construction from formal language

Maksym Shamrai[3], **Sergey Kuzivanov**[1], **Ekaterina Vinnik**[1], **Marin Gleb**[4], **and Semyon Grigorev**[1, 2]

1 Saint Petersburg State University **2** JetBrains Research **3** Kyiv Academic University **4** National Research University Higher School of Economics

DOI: 10.21105/joss.04184

Software

- Review
- Repository
- Archive

Editor: Nikoleta Glynatsi

Reviewers:

- @ocramz
- @JoyMitra

Submitted: 04 February 2022
Published: 20 February 2022

Summary

The formal language analysis approach has begun new development in the 50s of the last century (Chomsky, 1961, 2002). During this time, it found application not only in areas directly related to programming but, among other things, in the analysis of graph data models (Azimov & Grigorev, 2018; Hellings, 2015), which are used, for example, in bioinformatics (Huber et al., 2007) and social networks (Scott, 1988). Development entails new problems and challenges. Several of them are associated with the advent of new language classes (Okhotin, 2001, 2004), but not restricted with that.

Sometimes it happens that researchers lack the traditional methods of the field and come up with new ones. It seems like the most obvious way to invent a new approach is to connect your field with another field of study which already well known and try to apply methods of that field. So, Sapir, Birget, and Rips did like that (Sapir et al., 2002). They linked group theory and computational complexity theory by constructing a group presentation from a Turing machine. And had proved that group presentation can be treated like a computational model. Thus, they presented a new approach for representing computation, and accordingly, this can be used to take a fresh look at open problems.

Inspired by this, we wondered what if we treat formal language as a group presentation? This will allow us to use various group theoretical methods for formal languages. Moreover, it allows experimenting with computational group theory algorithms (*GAP – Groups, Algorithms, and Programming, Version 4.11.1*, 2021). But construction by hand like that takes a lot of time and paper. So it requires an efficient tool for automatic group presentation construction from formal language.

Statement of need

Modern research methods should not be limited to the traditional framework but should use all possible means to achieve a breakthrough result. The LangToGroup provides new opportunities for researchers in research related to computational models. Accordingly, the tool adds ways to analyze a problem, and sometimes understanding is a key that can lead to some insights about the task.

Mainly LangToGroup package has been developed for research in disciplines such as official languages, computational theory, and group theory, but this may be of interest to researchers in many other fields. The tool connects "computation" with group theory. That allows us to think mathematically about computational processes and use the concepts of group theory to analyze and solve problems.

Solution description

LangToGroup is a Haskell console application for group presentation construction. Haskell was chosen due to its rich and convenient type system, which we apply to represent algorithmic types such as formal grammar, Turing machine, S-machine, semigroup presentation, or group presentation. Furthermore, it is worth noting that data types were intentionally described in the same way as in the definitions of the corresponding concepts. This was done to maintain the maximum similarity to the articles Rotman (1995).

The package provides the capability to obtain a presentation of a group from a Turing machine in two ways described in (Sapir et al., 2002) and (Rotman, 1995). All of them are building a presentation of a group for a Turing Machine with preservation of its language. The first one does it through Symmetric Turing Machine and S-machine sequential constructions. The second one is via building a semigroup presentation. Moreover, two modifications of the second approach which output smaller presentations were developed as well. The time algorithms take depends on input grammar, but usually, it is no more than a few minutes.

From the beginning, the tool has been developing for the study of formal languages because of what algorithms for constructing a Turing machine from context-free and Boolean languages have additionally developed. Moreover, LangToGroup supports several types of input and output and each step of the construction can be printed in a document of LaTeX format for convenient usage. Therefore, LangToGroup presents a full pipeline of group presentation construction from formal languages and also can be used for construction from Turing machines.

References

Azimov, R., & Grigorev, S. (2018). Context-free path querying by matrix multiplication. *Proceedings of the 1st ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*. https://doi.org/10.1145/3210259.3210264

Chomsky, N. (1961). *Some methodological remarks on generative grammar*. Aspects of the Theory of Syntax. https://doi.org/10.1080/00437956.1961.11659755

Chomsky, N. (2002). *Syntactic structures*. Walter de Gruyter. https://doi.org/10.1515/9783110218329

GAP – Groups, Algorithms, and Programming, Version 4.11.1. (2021). The GAP Group. https://www.gap-system.org

Hellings, J. (2015). Querying for paths in graphs using context-free path queries. *arXiv: Databases*.

Huber, W., Carey, V. J., Long, L., Falcon, S., & Gentleman, R. (2007). Graphs in molecular biology. *BMC Bioinformatics*, *8*(6), S8. https://doi.org/10.1186/1471-2105-8-S6-S8

Okhotin, A. (2001). Conjunctive grammars. *J. Autom. Lang. Comb.*, *6*(4), 519–535. https://doi.org/10.25596/jalc-2001-519

Okhotin, A. (2004). Boolean grammars. *Information and Computation*, *194*(1), 19–48. https://doi.org/10.1016/j.ic.2004.03.006

Rotman, J. J. (1995). The word problem. In *An introduction to the theory of groups* (pp. 418–470). Springer. https://doi.org/10.1007/978-1-4612-4176-8

Sapir, M. V., Birget, J.-C., & Rips, E. (2002). Isoperimetric and isodiametric functions of groups. *Annals of Mathematics*, *156*(2), 345–466. https://doi.org/10.2307/3597195

Scott, J. (1988). Social network analysis. *Sociology*, *22*(1), 109–127. https://doi.org/10.1177/0038038588022001007