Bar-Hillel Theorem mechanization in Coq

Semyon Grigorev
Associate Professor
St.Petersburg State University
Universitetski pr., 28
St.Petersburg 198504, Russia
semen.grigorev@jetbrains.com
Researcher
JetBrains Research
Universitetskaya emb., 7-9-11/5A
St.Petersburg 199034, Russia
semen.grigorev@jetbrains.com

Sergey Bozhko
Student
St.Petersburg State University
Universitetski pr., 28
St.Petersburg 198504, Russia
gkerfimf@gmail.com

Ley
Position1
Department1
Institution1
Street1 Address1
City1, State1 Post-Code1
Saint-Petersburg
gkerfimf@gmail.com

Abstract

Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract. Text of abstract is very abstract.

....

Keywords Formal languages, Coq, Bar-Hillel, closure, intersection, regulalr language, context-free language

1 Introduction

Different on languages intersection is a one of fundamental problem in formal languages theory. Many different problems: Emptiness of intersection, closure under intersection, constructing of intersection

It is the well-known fact tat context-free languages are closed under intersection with regular languages. Theoretical result is Bar-Hillel [1] theorem which provide construction for resulting language description.

Language intersection problem is a foundation in many areas. Parsing, program analysis, graph analysis [6, 7]. Method proposed by Hellings is B-H theorem. All-path semantics. Foundation in some areas: graphs, code analysis, etc. Bar-Hillel theorem is a main on .

Mechanization (formalization) is important and many work done on formal languages theory mechanization. Parsing algorithms and reasoning about other problems on languages intersection.

Conference'17, July 2017, Washington, DC, USA
2017. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Short overview of current results. Many different parts of formal languages are mechanized. Algorithms and basic results.

The main contribution of this paper may be summarized as follows.

- We provide constructive proof of the Bar-Hillel theorem in Coq.
- We generalize Smolka's CFL results: terminals is abstract types....
- ...

2 Related Work

All results you use in your work. All relevant results in this field (excluded this work). Smolka, smb else [2–4].

As a result of this section we should conclude, that (1) this problem is open (2) it is important to solve this problem.

3 Bar-Hillel Theorem

Original B-H theorem and proof which we use as base. We work with the next formulation of the theorem.

Lemma 3.1. *If L is a context free language and $\varepsilon \notin L$ then there is grammar in Chomsky Normal Form that generates L.*

Lemma 3.2. *If $L \neq \varnothing$ and L is regular then L is the union of regular language $A_1, \ldots, A_n$ where each A_i is accepted by a DFA with exactly one final state.*

Theorem 3.3. *If L_1 is a context free language and L_2 is a regular language then $L_1 \cap L_2$ is context free.*

Sketch of proof:

1. By lemma 3.1 we can assume that there is a context-free grammar G in Chomsky normal form, such that $L(G) = L_1$
2.
3.
4.

4 CNF

One of importatnt part of proof is the fact that any context-free language can be described with grammar in CNF.

We want to reuse existing proof of convertion of original context-free grammar to CNF.

We choose Smolka's version.

5 B-H in Coq

Main part. All code are published on GitHub [1] What did you do and how. And, possible, why. Problems, nontrivial solutions, stc.

5.1 Smolka's code generalization

First we nwwd to generalize code of ...

5.2 General scheme of proof

General scheme of our proof is based on constructive proof presented by [**?**]

5.3 Part one: regular language and automata

First step is !!!!

5.4 Part two

5.5 Part N: final step

Finally we should proof main statement!

6 Conclusion

Short resume of main part (main results formulation). We present mechanization of Bar-Hillel theorem on closure of contex-free languages under intersection with regular.

Other algorithms on regular and context-free languages intersection. One of direction of future reserch is mechanization of practical algorithms which are just implementation of Bar-Hillel theorem. For example, context-free path querying algorithm, based on GLL [10] parsing algorithm [5].

Other problems on language intersection [8, 9].

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. nnnnnnn and Grant No. mmmmmmm. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

[1] Yehoshua Bar-Hillel, Micha Perles, and Eli Shamir. 1961. On formal properties of simple phrase structure grammars. *Sprachtypologie und Universalienforschung* 14 (1961), 143–172.

[2] Christian Doczkal, Jan-Oliver Kaiser, and Gert Smolka. 2013. A constructive theory of regular languages in Coq. In *International Conference on Certified Programs and Proofs*. Springer, 82–97.

[3] Christian Doczkal and Gert Smolkab. 2017. Regular Language Representations in the Constructive Type Theory of Coq. (2017).

[4] Denis Firsov. 2016. Certification of Context-Free Grammar Algorithms. (2016).

[5] Semyon Grigorev and Anastasiya Ragozina. 2016. Context-Free Path Querying with Structural Representation of Result. *arXiv preprint arXiv:1612.08872* (2016).

[6] J. Hellings. 2014. Conjunctive context-free path queries. (2014).

[7] Jelle Hellings. 2015. Querying for Paths in Graphs using Context-Free Path Queries. *arXiv preprint arXiv:1502.02242* (2015).

[8] Mark-Jan Nederhof and Giorgio Satta. 2002. Parsing non-recursive context-free grammars. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 112–119.

[9] Mark-Jan Nederhof and Giorgio Satta. 2004. The language intersection problem for non-recursive context-free grammars. *Information and Computation* 192, 2 (2004), 172–184.

[10] Elizabeth Scott and Adrian Johnstone. 2010. GLL parsing. *Electronic Notes in Theoretical Computer Science* 253, 7 (2010), 177–189.

A Appendix

Text of appendix ...

[1] https://github.com/YaccConstructor/YC_in_Coq