WoLLIC 2019

Bar-Hillel Theorem Mechanization in Coq

Sergey Bozhko, Leyla Khatbullina, **Semyon Grigorev**

JetBrains Research, Programming Languages and Tools Lab
Saint Petersburg University

July 05, 2019

Automated Theorem Proving

$$\underbrace{\texttt{CACATGGAGAGTTTGA...CTGGATCACCTCCTTT}}_{\sim 1500 \text{ symbols}}$$

- Classification

Automated Theorem Proving

$$\underbrace{\texttt{CACATGGAGAGTTTGA...CTGGATCACCTCCTTT}}_{\sim 1500 \text{ symbols}}$$

- Classification
  - Secondary structure handling

Automated Theorem Proving

$$\underbrace{\texttt{CACATGGAGAGTTTGA...CTGGATCACCTCCTTT}}_{\sim 1500 \text{ symbols}}$$

- Classification
  - Secondary structure handling
- Metagenomic assembly processing
  - Filter out chimeric sequences
  - Secondary structure handling

Formal Language Theory Mechanization

- Use parsing to extract features, not to model secondary structure
  - As compared to the classical way of probabilistic CF grammars utilization

Formal Language Theory Mechanization

- Use parsing to extract features, not to model secondary structure
  - As compared to the classical way of probabilistic CF grammars utilization
- Formal grammars as secondary structure description
- Parsing as features extraction
- Artificial neural network as probabilistic model for features processing

The Bar-Hillel Teorem

Theorem (Bar-Hillel)

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 the Proof

1. We can assume that there is a contextfree grammar G_{CNF} in Chomsky normal form, such that $L(G_{CNF}) = L_1$

Sketch of the Proof

1. We can assume that there is a contextfree grammar G_{CNF} in Chomsky normal form, such that $L(G_{CNF}) = L_1$
2. We can assume that there is a set of regular languages $\{A_1 \dots A_n\}$ where each A_i is recognized by a DFA with precisely one final state and $L_2 = A_1 \cup \dots \cup A_n$

Sketch of the Proof

1. We can assume that there is a contextfree grammar G_{CNF} in Chomsky normal form, such that $L(G_{CNF}) = L_1$
2. We can assume that there is a set of regular languages $\{A_1 \dots A_n\}$ where each A_i is recognized by a DFA with precisely one final state and $L_2 = A_1 \cup \dots \cup A_n$
   - If $L \neq \varnothing$ and L is regular then L is the union of regular language $A_1, \dots, A_n$ where each A_i is accepted by a DFA with precisely one final state

Sketch of the Proof

1. We can assume that there is a contextfree grammar G_{CNF} in Chomsky normal form, such that $L(G_{CNF}) = L_1$
2. We can assume that there is a set of regular languages $\{A_1 \dots A_n\}$ where each A_i is recognized by a DFA with precisely one final state and $L_2 = A_1 \cup \dots \cup A_n$
   - If $L \neq \varnothing$ and L is regular then L is the union of regular language $A_1, \dots, A_n$ where each A_i is accepted by a DFA with precisely one final state
3. For each A_i we can explicitly define a grammar of the intersection: $L(G_{CNF}) \cap A_i$

Sketch of the Proof

1. We can assume that there is a contextfree grammar G_{CNF} in Chomsky normal form, such that $L(G_{CNF}) = L_1$
2. We can assume that there is a set of regular languages $\{A_1 \ldots A_n\}$ where each A_i is recognized by a DFA with precisely one final state and $L_2 = A_1 \cup \ldots \cup A_n$
   - 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 precisely one final state
3. For each A_i we can explicitly define a grammar of the intersection: $L(G_{CNF}) \cap A_i$
4. Finally, we join them together with the operation of the union

Hofmann's Results Generalization

Jana Hofmann provides mechanization of the part of CFL in the Coq

- Basic definitions: terminal, nonterminal, grammar, word, ...

Hofmann's Results Generalization

Jana Hofmann provides mechanization of the part of CFL in the Coq

- Basic definitions: terminal, nonterminal, grammar, word, ...
- **Context-Free grammar to the Chomsky Normal Form convertion**

Hofmann's Results Generalization

Jana Hofmann provides mechanization of the part of CFL in the Coq

- Basic definitions: terminal, nonterminal, grammar, word, ...
- **Context-Free grammar to the Chomsky Normal Form convertion**

```
Inductive ter : Type :=
 | T : nat -> ter.
```

Yana Hofmann

```
Inductive ter : Type :=
 | T : Tt -> ter.
```

We need an arbitrary type for terminals and nontermianls!

Hofmann's Results Generalization

Jana Hofmann provides mechanization of the part of CFL in the Coq

- Basic definitions: terminal, nonterminal, grammar, word, ...
- **Context-Free grammar to the Chomsky Normal Form convertion**

```
Inductive ter : Type :=
 | T : nat -> ter.
```

Yana Hofmann

```
Inductive ter : Type :=
 | T : Tt -> ter.
```

We need an arbitrary type for terminals and nontermianls!

And now we should carefully rewrite all existing stuff ...

DFA Splitting

```
Lemma correct_split:
   forall dfa w,
     dfa_language dfa w <->
     exists sdfa,
        In sdfa (split_dfa dfa) /\ s_dfa_language sdfa w.
```

Chomsky Induction

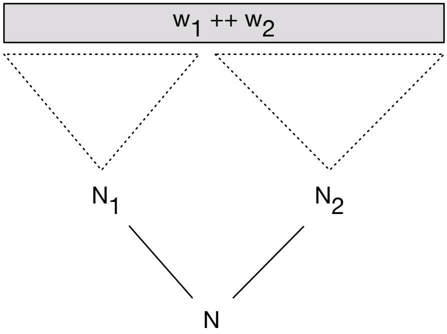
Lemma

Let G be a grammar in CNF. Consider an arbitrary nonterminal $N \in G$ and phrase which consists only of terminals w. If w is derivable from N and $|w| \geq 2$, then there exists two nonterminals N_1, N_2 and two phrases w_1, w_2 such that: $N \to N_1 N_2 \in G$, $der(G, N_1, w_1)$, $der(G, N_2, w_2)$, $|w_1| \geq 1$, $|w_2| \geq 1$ and $w_1 + +w_2 = w$.

Chomsky Induction

Lemma

Let G be a grammar in CNF. Consider an arbitrary nonterminal $N \in G$ and phrase which consists only of terminals w. If w is derivable from N and $|w| \geq 2$, then there exists two nonterminals N_1, N_2 and two phrases w_1, w_2 such that: $N \to N_1 N_2 \in G$, $der(G, N_1, w_1)$, $der(G, N_2, w_2)$, $|w_1| \geq 1$, $|w_2| \geq 1$ and $w_1 ++ w_2 = w$.



Chomsky Induction in Coq

```
Definition syntactic_analysis_is_possible :=
forall (G : grammar) (A : var) (w : phrase),
   der G A w -> (R A w \in G)
                \/
                (exists rhs, R A rhs \in G /\ derf G rhs w).
```

Languges Union

```
Variable grammars: seq (var * grammar).

Theorem correct_union:
forall word,
  language (grammar_union grammars) (V (start Vt))
           (to_phrase word)
  <->
  exists s_l,
    language (snd s_l) (fst s_l) (to_phrase word)
    /\
    In s_l grammars.
```

The Final Theorem

Theorem

For any two decidable types ***Tt*** *and* ***Nt*** *for types of terminals and nonterminals correspondingly. If there exists a bijection from* ***Nt*** *to* $\mathbb{N}$ *and syntactic analysis is possible (in the sense of our definition), then for any DFA* ***dfa*** *and any context-free grammar* ***G****, there exists the context-free grammar* G_{INT}*, such that* $L(G_{INT}) = L(G) \cap L(dfa)$*.*

The Final Theorem in Coq

```
Theorem grammar_of_intersection_exists:
  exists
   (NewNonterminal: Type)
   (IntersectionGrammar: @grammar Terminal NewNonterminal)
   St,
  forall word,
    dfa_language dfa word /\ language G S (to_phrase word)
    <->
    language IntersectionGrammar St (to_phrase word).
```

Conclusion

- We present mechanized in Coq proof of the Bar-Hillel theorem on the closure of context-free languages under intersection with the regular languages
- We generalize the results of Jana Hofmann and Gert Smolka
  - The definition of the terminal and nonterminal alphabets in context-free grammar were made generic
  - All related definitions and theorems were adjusted to work with the updated definition
- All results are published at GitHub and are equipped with automatically generated documentation

Future work

- Ruy J. G. B. de Queiroz vs Jana Hifmann
  - We use results of Jana Hofman
  - Results of Ruy J. G. B. de Queiroz looks more mature
  - Is it even possible to cretae one "true" solution in this area?

Future work

- Ruy J. G. B. de Queiroz vs Jana Hifmann
  - We use results of Jana Hofman
  - Results of Ruy J. G. B. de Queiroz looks more mature
  - Is it even possible to cretae one "true" solution in this area?
- Mechanization of practical algorithms which are just implementation of the Bar-Hillel theorem
  - Context-free path querying algorithm, based on CYK or even on GLL parsing algorithm
  - Certified algorithm for context-free constrained path querying for graph databases

Contact Information

- Semyon Grigorev:
  - s.v.grigoriev@spbu.ru
  - Semen.Grigorev@jetbrains.com
- Sergey Bozhko:
  - Max Planck Institute for Software Systems (MPI-SWS), Saarbrcken, Germany
  - sbozhko@mpi-sws.com
- Leyla Khatbullina:
  - St.Petersburg Electrotechnical University "LETI", St.Petersburg, Russia
  - leila.xr@gmail.com

- Sources: https://github.com/YaccConstructor/YC_in_Coq

Thanks!