Verified Algorithms for Context-Free Grammars in Coq

Final Bachelor Talk

Jana Hofmann
Advisor: Prof. Dr. Gert Smolka

SAARLAND UNIVERSITY
COMPUTER SCIENCE

29th August 2016

Introduction

Formalization

Finite Iteration

Word Problem

Binarization

Conclusion

What is a Context-Free Grammar?

Example

$$A \backslash aAc$$
$$A \backslash aBc$$
$$B \backslash BB$$
$$B \backslash b$$

- Describe context-free languages
  e.g. $\{a^n b^m c^n \mid m, n > 0\}$
- Are used to describe (programming) languages

Important Terms

Example

$A \backslash aAc$

$A \backslash aBc$

$B \backslash BB$

$B \backslash b$

- **Grammar** G consists of:
  - **symbols** s
    characters $a, b, c, \ldots$
    variables $A, B, C, \ldots$
  - **phrases** $u, v, w, \ldots$
  - **rules** $A \backslash u$
- **Words** are phrases containing only characters
- u **derives** a word $w$by rewriting rules of the grammar
- A **language** of a grammar is the set of all words we can derive starting with some variable ($\mathcal{L}_G^A$ or $\mathcal{L}_G^B$)

Derivations

Example

$$A \backslash aAc$$
$$A \backslash aBc$$
$$B \backslash BB$$
$$B \backslash b$$

Example

Derivation of *aaabccc*

$$\begin{aligned} A &\Rightarrow aAc & A\backslash aAc \in G \\ &\Rightarrow aaAcc & A\backslash aAc \in G \\ &\Rightarrow aaaBccc & A\backslash aBc \in G \\ &\Rightarrow aaabccc & B\backslash b \in G \end{aligned}$$

Decidability Problems

1. Empty language problem: $\mathcal{L}_G^A \equiv \emptyset$?
2. Word problem: $w \in \mathcal{L}_G^A$?
3. Finiteness: Is $\mathcal{L}_G^A$ finite?

} decidable

4. Equality problem: $\mathcal{L}_G^A \equiv \mathcal{L}_{G'}^{A'}$?
5. Regularity: Is $\mathcal{L}_G^A$ regular?

} undecidable

Chomsky Normal Form (CNF)

A grammar G is in CNF, if for all rules $A \backslash u \in G$ holds:

- $u \neq \varepsilon$
- $u = a$ or
- $|u| \leq 2$ and all symbols in u are variables

All grammars can be transformed into CNF

Serves as basis for CYK algorithm to decide the word problem

Previous Work

- Denis Firsov and Tarmo Uustalu
  Certified Normalization of Context-Free Grammars
  Institute of Cybernetics at TUT, 2015
- Denis Firsov and Tarmo Uustalu
  Certified CYK parsing of context-free languages
  Journal of Logical and Algebraic Methods in Programming 83.5 (2014): 459-468
- Aditi Barthwal
  A formalization of the theory of context-free languages in higher order logic
  The Australian National University, Ph.D. thesis, 2010
- Marcus V. M. Ramos
  Formalization of context-free language theory
  Universidade Federal de Pernambuco, Ph.D. thesis, 2016

Contributions

Decidability results

- Decidability of word problem ($w \in \mathcal{L}_G^A$?)
- Decidability of emptiness problem ($\mathcal{L}_G^A \equiv \emptyset$?)

Grammar transformations

- Elimination of ε-**rules** ($A \backslash \varepsilon$)
- Elimination of **unit-rules** ($A \backslash B$)
- Binarization (every rule of the form $A \backslash s_1 s_2$)
- Separation (every rule of the form $A \backslash a$ or $A \backslash B_1 \ldots B_n$)

} yield grammar in CNF

\+ Elimination of deterministic variables

Definitions

We use lists for grammars and derivations

$$
\begin{aligned}
\mathit{var} &:= n && n \in \mathbb{N} \\
\mathit{char} &:= n && n \in \mathbb{N} \\
\mathit{symbol} &:= \mathit{var} \mid \mathit{char}
\end{aligned}
$$

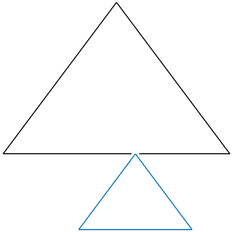
$$
\begin{aligned}
\mathit{phrase} &:= \mathscr{L}(\mathit{symbol}) \\
\mathit{rule} &:= \mathit{var} \times \mathit{phrase} \\
\mathit{grammar} &:= \mathscr{L}(\mathit{rule})
\end{aligned}
$$

Definitions

We use lists for grammars and derivations

The notion of derivability can be defined inductively:

$$\frac{}{A \overset{G}{\Rightarrow} A} \qquad \frac{A \backslash u \in G}{A \overset{G}{\Rightarrow} u} \qquad \frac{A \overset{G}{\Rightarrow} uBw \quad B \overset{G}{\Rightarrow} v}{A \overset{G}{\Rightarrow} uvw}$$

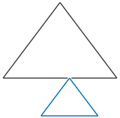


Languages of a grammar are defined in terms of derivability:

$$\mathcal{L}_G^A\ w := A \overset{G}{\Rightarrow} w \ \wedge \ w \textit{ is a word}$$

Alternative Derivation Predicates

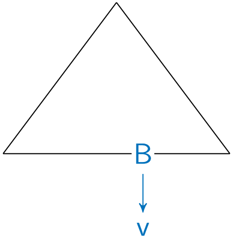
$$\frac{}{A \overset{G}{\Rightarrow} A} \qquad \frac{A \backslash u \in G}{A \overset{G}{\Rightarrow} u} \qquad \frac{A \overset{G}{\Rightarrow} uBw \quad B \overset{G}{\Rightarrow} v}{A \overset{G}{\Rightarrow} uvw}$$



- Give several derivation predicates for different purposes
- Heart of the work

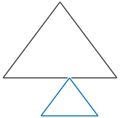
$\Rightarrow_{\mathcal{L}}$ is a right-linear variant of $\Rightarrow$

$$\frac{}{A \overset{G}{\Rightarrow}_{\mathcal{L}} A} \qquad \frac{A \overset{G}{\Rightarrow}_{\mathcal{L}} uBw \quad B \backslash v \in G}{A \overset{G}{\Rightarrow}_{\mathcal{L}} uvw}$$



Alternative Derivation Predicates

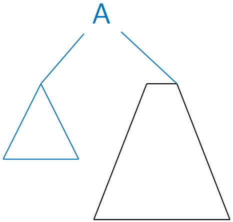
$$\frac{}{A \overset{G}{\Rightarrow} A} \qquad \frac{A \backslash u \in G}{A \overset{G}{\Rightarrow} u} \qquad \frac{A \overset{G}{\Rightarrow} uBw \quad B \overset{G}{\Rightarrow} v}{A \overset{G}{\Rightarrow} uvw}$$



- Give several derivation predicates for different purposes
- Heart of this work

$\Rightarrow_{\mathcal{F}}$ is symmetric and resembles a derivation tree

$$\frac{}{u \overset{G}{\Rightarrow}_{\mathcal{F}} u} \qquad \frac{A \backslash u \in G \quad u \overset{G}{\Rightarrow}_{\mathcal{F}} v}{A \overset{G}{\Rightarrow}_{\mathcal{F}} v} \qquad \frac{s \overset{G}{\Rightarrow}_{\mathcal{F}} u \quad v \overset{G}{\Rightarrow}_{\mathcal{F}} w}{sv \overset{G}{\Rightarrow}_{\mathcal{F}} uw}$$



Finite Fixed Point Iteration (FFPI) (ICL 2014)

$f : X \to X$
$x : X$

> x is a fixed point of a function f, if $f\ x = x$.

Is $f^n x$ a fixed point of f?

Lemma (Fixed Point)

Let $\sigma : X \to \mathbb{N}$ such that for every number n either $\sigma(f^n\ x) > \sigma(f^{n+1}\ x)$ or $f^n\ x$ is a fixed point of f. Then $f^{\sigma x}\ x$ is a fixed point of f.

$$\sigma\ x > \sigma(f\ x) > \sigma(f^2\ x) > \cdots > \sigma(f^n\ x) = 0$$

max. $\sigma\ x$ times
$\Rightarrow n = \sigma\ x$

must be a fixed point of f

Finite Fixed Point Iteration (FFPI) (ICL 2014)

$f : X \to X$
$x : X$

> x is a fixed point of a function f, if $f\ x = x$.

Is $f^n x$ a fixed point of f?

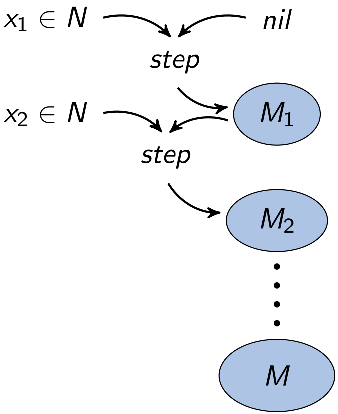
Lemma (Induction)

Let $p : X \to Prop$ and $x \in X$ such that $p\ x$ and $\forall z.\ p\ z \to p(f\ z)$. Then $p(f^n\ x)$ for every number n.

Finite Closure Iteration (FCI) (ICL 2014)

N : $list\ X$
$step$: $list\ X \to X \to Prop$ (decidable)

Wanted: $M \subseteq N$ s.t. M is closed with respect to $step$



Finite Closure Iteration (FCI) (ICL 2014)

$N : list\ X$
$step : list\ X \to X \to Prop$ (decidable)

Lemma

If step is decidable, then we can construct a list M, s.t.

1. *Closure:* If $step\ M\ x$ and $x \in N$, then $x \in M$.
2. *Induction:* Let $p : X \to Prop$ such that $step\ xs\ x \to p\ x$ for all $xs \subseteq p$ and $x \in N$. Then $M \subseteq p$.

Decidability of Word Problem

$w \in \mathcal{L}_G^A$?

More general: $A \overset{G}{\Longrightarrow} u$?

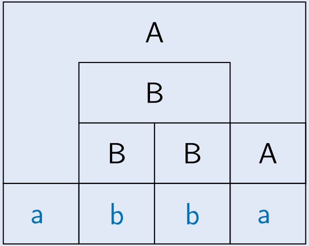
Existing solutions: CYK algorithm (bottom-up), Earley algorithm (top-down)

We give a generalized CYK-algorithm (bottom-up chart parsing algorithm)

Example

Let G and u be given as

$$\begin{aligned} G &:= A \backslash aBA \qquad B \backslash BB \\ & A \backslash a \qquad\;\;\;\; B \backslash b \\ u &:= abba \end{aligned}$$



Decidability of Word Problem

Let G and u be fixed. We define:

- $\mathit{item} : \mathit{Type} := \mathit{symbol} \times \mathit{phrase}$
- Segments: $v \precsim_s u := \exists u_1\ u_2.\ u = u_1 v u_2$

Aim: Construct $D : \mathit{list\ item}$, such that

$$(s, v) \in D \leftrightarrow v \precsim_s u \land s \xRightarrow{G} v$$

We use FCI:

- $N := \mathit{items}\ (G, u)$ ($\rightsquigarrow$ "all symbols of G $\times$ all segments of u")
- $\mathit{step}\ M\ (s, v) := v = s\ \lor\ s = A \land \exists M' \subseteq M.\ A \backslash (\pi_1\ M') \in G \land v = \mathrm{concat}(\pi_2\ M')$
- $D := \mathrm{FCI}\ N\ \mathit{step}$

Decidability of Word Problem

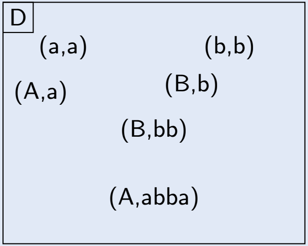
We use FCI:

- $N :=$ *items* (G, u) ($\rightsquigarrow$ "all symbols of G $\times$ all segments of u")
- $step\ M\ (s, v) \ := \ v = s \ \vee \ s = A \wedge$
  $\exists M' \subseteq M.\ A \backslash (\pi_1\ M') \in G \wedge v = \text{concat}(\pi_2\ M')$
- $D :=$ FCI N *step*

Example

Let G and u be given as

$$G \ := \ A \backslash aBA \qquad B \backslash BB$$
$$A \backslash a \qquad B \backslash b$$
$$u \ := \ abba$$



Decidability of Word Problem

Lemma (FCI)
If *step* is decidable, then we can construct a list M, s.t.

1. Closure: If $step\ M\ x$ and $x \in N$, then $x \in M$.
2. Induction: Let $p : X \to Prop$ such that $step\ xs\ x \to p\ x$ for all $xs \subseteq p$ and $x \in N$. Then $M \subseteq p$.

Lemma

$(s, v) \in D \leftrightarrow v \precsim_s u \land s \overset{G}{\Rightarrow} v$

Proof

- $\rightarrow$ *Using the induction lemma of FCI.*
- $\leftarrow$ *Using the closure lemma of FCI.*

Theorem (The word problem of context-free languages is decidable)

Let G and w be given. $\forall A.\ (A, w) \in D_G, w$ *iff* $A \overset{G}{\Rightarrow} w$.

Binarization

Example

$$G := A\backslash aBCa \quad B\backslash b \quad C\backslash c$$

$$G^2 := A\backslash aA_0 \quad A_0\backslash BCa \quad A_0\backslash BA_1 \quad A_1\backslash Ca \quad B\backslash b \quad C\backslash c$$

We use FFPI to compute G^2

Binarization

Lemma (FFPI - Fixed Point)
Let $\sigma : X \to \mathbb{N}$ such that for every number n either $\sigma(f^n\ x) > \sigma(f^{n+1}\ x)$ or $f^n\ x$ is a fixed point of f. Then $f^{\sigma x}\ x$ is a fixed point of f.

We use FFPI to compute G^2

$$
\begin{aligned}
&\text{step'} : \textit{grammar} \to \textit{grammar} \to \textit{grammar}\\
&\text{step'}\ G'\ [\,] && := [\,]\\
&\text{step'}\ G'\ (A\backslash[\,] :: G) && := A\backslash[\,] :: \text{step'}\ G'\ G\\
&\text{step'}\ G'\ (A\backslash[s_0] :: G) && := A\backslash[s_0] :: \text{step'}\ G'\ G\\
&\text{step'}\ G'\ (A\backslash[s_0;s_1] :: G) && := A\backslash[s_0;s_1] :: \text{step'}\ G'\ G\\
&\text{step'}\ G'\ (A\backslash(s_0::u) :: G) && := \text{let } B := \text{fresh } G'\\
& && \quad \text{in } A\backslash[s_0;B] :: B\backslash u :: G\\
&\text{step } G && := \text{step'}\ G\ G
\end{aligned}
$$

step function

$$
\begin{aligned}
&\text{count} : \textit{grammar} \to \mathbb{N}\\
&\text{count } [\,] && := 0\\
&\text{count } (A\backslash u :: G) && := \text{if } |u| \le 2 \text{ then count } G\\
& && \quad \text{else } |u| + \text{count } G
\end{aligned}
$$

size function

$G^2 := \text{FFPI } \textit{step count}$

Binarization

Lemma (FFPI - Fixed Point)
Let $\sigma : X \to \mathbb{N}$ such that for every number n either $\sigma(f^n\ x) > \sigma(f^{n+1}\ x)$ or $f^n\ x$ is a fixed point of f. Then $f^{\sigma x}\ x$ is a fixed point of f.

Lemma (FFPI - Induction)
Let $p : X \to Prop$ and $x \in X$ such that $p\ x$ and $\forall z.\ p\ z \to p(f\ z)$. Then $p(f^n\ x)$ for every number n.

Lemma

1. G^2 is binary
2. For every (non fresh) A: $\mathcal{L}^A_G \equiv \mathcal{L}^A_{G^2}$

Proof

1. G^2 is a fixed point of *step* (FFPI fixed point lemma) and every fixed point of *step* is binary.
2. (FFPI induction lemma) prove: For every (non fresh) A: $\mathcal{L}^A_G \equiv \mathcal{L}^A_{step\ G}$

Conclusion

What we did

- Decidability results
  - Decidability of word problem ($w \in \mathcal{L}_G^A$?)
  - Decidability of emptiness problem ($\mathcal{L}_G^A \equiv \emptyset$?)
- Grammar transformations
  - Elimination of ε-**rules** ($A\backslash\varepsilon$)
  - Elimination of **unit-rules** ($A\backslash B$)
  - Binarization (every rule of the form $A\backslash s_1 s_2$)
  - Separation (every rule of the form $A\backslash a$ or $A\backslash B_1 \dots B_n$)

  } yield grammar in CNF

  \+ Elimination of deterministic variables

Future Work

- Decidability of finiteness of context-free languages
- Elimination of useless symbols
- Closure properties of CFLs

Sources

- Dexter C. Kozen
  Automata and Computability
  Springer, 1997
- John E. Hopcroft and Jeffrey D. Ullman
  Introduction to Automata Theory, Languages and Computation
  Addison-Wesley, Reading, Ma., USA, 1997
- Gert Smolka and Chad E. Brown
  Introduction to Computational Logic
  Lecture Notes [PDF], 2014. Retrieved from
  https://www.ps.uni-saarland.de/courses/cl-ss14/script/icl.pdf

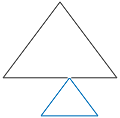
Closure Properties

Let $C, C_1,\ C_2$ be a context-free and R a regular language

- $C_1 \cup C_2$ is context-free
- $C_1 \cap C_2$ is in general not context-free
- $C \cup R$ is context-free
- $\overline{C}$ is not context-free

Alternative Derivation Predicates

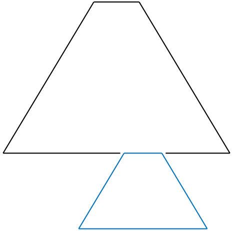
$$\frac{}{A \overset{G}{\Rightarrow} A} \qquad \frac{A \backslash u \in G}{A \overset{G}{\Rightarrow} u} \qquad \frac{A \overset{G}{\Rightarrow} uBw \quad B \overset{G}{\Rightarrow} v}{A \overset{G}{\Rightarrow} uvw}$$



- Give several derivation predicates for different purposes
- Heart of this work

$\Rightarrow_\mathcal{T}$ is a symmetric variant of $\Rightarrow$

$$\frac{}{u \overset{G}{\Rightarrow}_\mathcal{T} u} \qquad \frac{A \backslash u \in G}{A \overset{G}{\Rightarrow}_\mathcal{T} u} \qquad \frac{u \overset{G}{\Rightarrow}_\mathcal{T} u_1 v u_2 \quad v \overset{G}{\Rightarrow}_\mathcal{T} w}{u \overset{G}{\Rightarrow}_\mathcal{T} u_1 w u_2}$$



Use of Derivation Predicates

Decidability of Emptiness Problem	$\Rightarrow_{\mathcal{F}}$
Decidability of Word Problem	$\Rightarrow_{\mathcal{F}}$
Elimination of Epsilon Rules	$\Rightarrow_{\mathcal{T}}$, $\Rightarrow_{\mathcal{L}}$
Elimination of Unit Rules	$\Rightarrow_{\mathcal{F}}$
Elimination of Deterministic Variables	$\Rightarrow$, $\Rightarrow_{\mathcal{L}}$
Separation of Grammars	$\Rightarrow$
Binarization of Grammars	$\Rightarrow$

Use of FFPI and FCI

Decidability of Emptiness Problem	FCI
Decidability of Word Problem	FCI
Elimination of Epsilon Rules	-
Elimination of Unit Rules	FCI
Elimination of Deterministic Variables	-
Separation of Grammars	FFPI
Binarization of Grammars	FFPI