Commutative Lambek Grammars are **not** Context Free

Tikhon Pshenitsyn
ptihon@yandex.ru
Supervisor: prof. Mati Pentus

Department of Mathematical Logic and Theory of Algorithms
Faculty of Mathematics and Mechanics
Lomonosov Moscow State University
Russia

November 9, 2022

Context-Free Grammars

Context-free grammar (Chomsky 1956, 1958):

Example

Productions:

1. $S \to N\ V$,
2. $V \to VTr\ N$,
3. $N \to \text{John}$, $N \to \text{Mary}$,
4. $V \to \text{sleeps}$,
5. $VTr \to \text{loves}$.

The start symbol: S.
Derivation: $S \Rightarrow N\ V \Rightarrow N\ VTr\ N \Rightarrow^* \text{John loves Mary}$.

Categorial Grammars and Lambek Calculus

- Lambek 1958: a logical calculus L (Lambek calculus) as a basis for describing a language.

Categorial Grammars and Lambek Calculus

- Lambek 1958: a logical calculus L (Lambek calculus) as a basis for describing a language.
- Lambek calculus L is a substructural non-commutative logic. Formulas: $Fm_{\mathrm{L}} := n, s, \ldots \mid Fm_{\mathrm{L}} \backslash Fm_{\mathrm{L}} \mid Fm_{\mathrm{L}} / Fm_{\mathrm{L}} \mid Fm_{\mathrm{L}} \cdot Fm_{\mathrm{L}}$.

Categorial Grammars and Lambek Calculus

- Lambek 1958: a logical calculus L (Lambek calculus) as a basis for describing a language.
- Lambek calculus L is a substructural non-commutative logic. Formulas: $Fm_{\mathrm{L}} := n, s, \ldots \mid Fm_{\mathrm{L}} \backslash Fm_{\mathrm{L}} \mid Fm_{\mathrm{L}} / Fm_{\mathrm{L}} \mid Fm_{\mathrm{L}} \cdot Fm_{\mathrm{L}}$.
- The Lambek calculus in the Gentzen style deals with sequents; there is 1 axiom and 6 inference rules.

Categorial Grammars and Lambek Calculus

- Lambek 1958: a logical calculus L (Lambek calculus) as a basis for describing a language.
- Lambek calculus L is a substructural non-commutative logic. Formulas: $Fm_{\mathrm{L}} := n, s, \ldots \mid Fm_{\mathrm{L}} \backslash Fm_{\mathrm{L}} \mid Fm_{\mathrm{L}} / Fm_{\mathrm{L}} \mid Fm_{\mathrm{L}} \cdot Fm_{\mathrm{L}}$.
- The Lambek calculus in the Gentzen style deals with sequents; there is 1 axiom and 6 inference rules.
- L has nice semantics (residuated semigroups; languages; relations).

Categorial Grammars and Lambek Calculus

- Lambek 1958: a logical calculus L (Lambek calculus) as a basis for describing a language.
- Lambek calculus L is a substructural non-commutative logic. Formulas: $Fm_{\mathrm{L}} := n, s, \ldots \mid Fm_{\mathrm{L}} \backslash Fm_{\mathrm{L}} \mid Fm_{\mathrm{L}} / Fm_{\mathrm{L}} \mid Fm_{\mathrm{L}} \cdot Fm_{\mathrm{L}}$.
- The Lambek calculus in the Gentzen style deals with sequents; there is 1 axiom and 6 inference rules.
- L has nice semantics (residuated semigroups; languages; relations).
- A grammar consists of an assignment of a finite number of formulas to each terminal unit and of a distinguished formula S.

Categorial Grammars and Lambek Calculus

- Lambek 1958: a logical calculus L (Lambek calculus) as a basis for describing a language.
- Lambek calculus L is a substructural non-commutative logic. Formulas: $Fm_{\mathrm{L}} := n, s, \dots \mid Fm_{\mathrm{L}} \backslash Fm_{\mathrm{L}} \mid Fm_{\mathrm{L}} / Fm_{\mathrm{L}} \mid Fm_{\mathrm{L}} \cdot Fm_{\mathrm{L}}$.
- The Lambek calculus in the Gentzen style deals with sequents; there is 1 axiom and 6 inference rules.
- L has nice semantics (residuated semigroups; languages; relations).
- A grammar consists of an assignment of a finite number of formulas to each terminal unit and of a distinguished formula S.
- A string $a_1 \dots a_n$ is accepted by a grammar if we can replace each a_i by a formula T_i assigned to it in such a way that $\mathrm{L} \vdash T_1, \dots, T_n \to S$.

Lambek Calculus

Axiom: $A \to A$.
Rules:

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, \Pi, A\backslash B, \Delta \to C}\ (\backslash \to) \qquad \frac{A, \Pi \to B}{\Pi \to A\backslash B}\ (\to \backslash)$$

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, B/A, \Pi, \Delta \to C}\ (/ \to) \qquad \frac{\Pi, A \to B}{\Pi \to B/A}\ (\to /)$$

$$\frac{\Gamma, A, B, \Delta \to C}{\Gamma, A \cdot B, \Delta \to C}\ (\cdot \to) \qquad \frac{\Pi \to A \quad \Psi \to B}{\Pi, \Psi \to A \cdot B}\ (\to \cdot)$$

A, B, C are formulas; Γ, Δ are sequences of formulas; Π, Ψ are nonempty sequences of formulas.

Lambek Grammars and Context-Free Grammars

Example

A natural way of modelling the string *John loves Mary* via a categorial grammar is as follows:

John	loves	Mary
n	$(n\backslash s)/n$	n

Lambek Grammars and Context-Free Grammars

Example

A natural way of modelling the string *John loves Mary* via a categorial grammar is as follows:

John n	loves $(n\backslash s)/n$	Mary n

Indeed, it can be verified that $\mathrm{L} \vdash n, (n\backslash s)/n, n \to s$.
Note that $\mathrm{L} \nvdash (n\backslash s)/n, n, n \to s$ (cf. *"loves John Mary"*).

Lambek Grammars and Context-Free Grammars

Example

A natural way of modelling the string *John loves Mary* via a categorial grammar is as follows:

John	loves	Mary
n	$(n\backslash s)/n$	n

Indeed, it can be verified that $\mathrm{L} \vdash n, (n\backslash s)/n, n \to s$.
Note that $\mathrm{L} \not\vdash (n\backslash s)/n, n, n \to s$ (cf. *"loves John Mary"*).

Theorem (Pentus, 1993)

Lambek grammars generate exactly context-free languages without the empty word.

Commutative Lambek Calculus LP

- van Benthem 1983, 1991: the commutative Lambek calculus LP (i.e. the order of formulas in an antecedent does not matter).
- LP = the multiplicative fragment of the intuitionistic linear logic.
- Two operations: product $\otimes$ and linear implication $\multimap$. Fm = the set of formulas.
- A sequent is of the form $\Pi \to A$ where Π is a nonempty finite multiset of formulas and A is a formula.

Commutative Lambek Calculus LP

Commutative Lambek Calculus LP

Axiom: $A \to A$ for $A \in Fm$.

Rules:

$$\frac{\Pi \to A \quad \Gamma, B \to C}{\Gamma, \Pi, A \multimap B \to C}\ (L \multimap) \qquad \frac{\Pi, A \to B}{\Pi \to A \multimap B}\ (R \multimap)$$

$$\frac{\Gamma, A, B \to C}{\Gamma, A \otimes B \to C}\ (L\otimes) \qquad \frac{\Pi \to A \quad \Psi \to B}{\Pi, \Psi \to A \otimes B}\ (R\otimes)$$

Commutative Lambek Calculus LP

Commutative Lambek Calculus LP

Axiom: $A \to A$ for $A \in Fm$.

Rules:

$$\frac{\Pi \to A \quad \Gamma, B \to C}{\Gamma, \Pi, A \multimap B \to C}\ (L \multimap) \qquad \frac{\Pi, A \to B}{\Pi \to A \multimap B}\ (R \multimap)$$

$$\frac{\Gamma, A, B \to C}{\Gamma, A \otimes B \to C}\ (L\otimes) \qquad \frac{\Pi \to A \quad \Psi \to B}{\Pi, \Psi \to A \otimes B}\ (R\otimes)$$

Example

$$\cfrac{\cfrac{\cfrac{q \to q \quad p \to p}{q, p \to q \otimes p}\ (\otimes R) \qquad p \to p}{(q \otimes p) \multimap p, q, p \to p}\ (\multimap L)}{(q \otimes p) \multimap p, q \to p \multimap p}\ (\multimap R)$$

Recognizing Power of Commutative Lambek Grammars

Definition

A commutative Lambek grammar consists of a finite alphabet Σ, of a distinguished formula $S \in Fm$, and of a finite binary relation $\triangleright \subseteq \Sigma \times Fm$ between symbols of the alphabet and formulas of LP.

Recognizing Power of Commutative Lambek Grammars

Definition

A commutative Lambek grammar consists of a finite alphabet Σ, of a distinguished formula $S \in Fm$, and of a finite binary relation $\triangleright \subseteq \Sigma \times Fm$ between symbols of the alphabet and formulas of LP.

Definition

The language generated by such a grammar is the set of strings $a_1 \dots a_n$ over Σ such that for some formulas $T_1, \dots, T_n$ of LP it holds that:

1. $a_i \triangleright T_i$ $(i = 1, \dots, n)$;
2. $\mathrm{LP} \vdash T_1, \dots, T_n \to S$.

Recognizing Power of Commutative Lambek Grammars

Definition

The language generated by such a grammar is the set of strings $a_1 \dots a_n$ over Σ such that for some formulas $T_1, \dots, T_n$ of LP it holds that:

1. $a_i \triangleright T_i$ $(i = 1, \dots, n)$;
2. $\mathrm{LP} \vdash T_1, \dots, T_n \to S$.

Example

Let $\Sigma = \{a, b\}$, $S = p \multimap p$, and $a \triangleright (q \otimes p) \multimap p$, $b \triangleright q$. Then the string ab belongs to the language generated by this grammar:

$$\dfrac{\dfrac{\dfrac{q \to q \quad p \to p}{q, p \to q \otimes p}\,(\otimes R) \qquad p \to p}{(q \otimes p) \multimap p, q, p \to p}\,(\multimap L)}{(q \otimes p) \multimap p, q \to p \multimap p}\,(\multimap R)$$

Recognizing Power of Commutative Lambek Grammars

Definition

The language generated by such a grammar is the set of strings $a_1 \dots a_n$ over Σ such that for some formulas $T_1, \dots, T_n$ of LP it holds that:

1. $a_i \triangleright T_i$ $(i = 1, \dots, n)$;
2. $\mathrm{LP} \vdash T_1, \dots, T_n \to S$.

Remark

Clearly, if a string $a_1 \dots a_n$ is accepted by an commutative Lambek grammar, then each its permutation $a_{\sigma(1)} \dots a_{\sigma(n)}$ is accepted as well.

Recognizing Power of Commutative Lambek Grammars

Definition

The language generated by such a grammar is the set of strings $a_1 \dots a_n$ over Σ such that for some formulas $T_1, \dots, T_n$ of LP it holds that:

1. $a_i \triangleright T_i$ $(i = 1, \dots, n)$;
2. $\mathrm{LP} \vdash T_1, \dots, T_n \to S$.

Definition

A permutation closure of a language is the set of all permutations of all its strings.

Recognizing Power of Commutative Lambek Grammars

Theorem (Buszkowski, 1983; van Benthem, 1991)

Commutative Lambek grammars generate all permutation closures of context-free languages.

Recognizing Power of Commutative Lambek Grammars

Theorem (Buszkowski, 1983; van Benthem, 1991)

Commutative Lambek grammars generate all permutation closures of context-free languages.

- The converse statement is an open question. It is present in van Benthem's list of open problems (1993).
- Buszkowski's conjecture (1984): commutative Lambek grammars without product generate exactly permutation closures of context-free languages.
- Stepan Kuznetsov's conjecture (personal communication): the converse does not hold.

Recognizing Power of Commutative Lambek Grammars: New Results

Theorem

Commutative Lambek grammars generate more than permutation closures of context-free languages.

For example, they generate the permutation closure of the language $\{a^l b^n \mid 0 < n, 0 \leq l \leq n^2\}$.

Recognizing Power of Commutative Lambek Grammars: New Results

Theorem

Commutative Lambek grammars generate more than permutation closures of context-free languages.

For example, they generate the permutation closure of the language $\{a^l b^n \mid 0 < n, 0 \leq l \leq n^2\}$.

Theorem

Commutative Lambek grammars are equivalent to commutative Lambek grammars without product $\otimes$ (i.e. which use only formulas with $\multimap$).

Recognizing Power of Commutative Lambek Grammars: New Results

Theorem

Commutative Lambek grammars generate more than permutation closures of context-free languages.

For example, they generate the permutation closure of the language $\{a^l b^n \mid 0 < n, 0 \leq l \leq n^2\}$.

Theorem

Commutative Lambek grammars are equivalent to commutative Lambek grammars without product $\otimes$ (i.e. which use only formulas with $\multimap$).

The proof of both theorems is based on introducing another formalism (linearly-restricted branching vector addition systems with states, or lBVASSAM), which combines the categorial approach and the generative one, and showing that commutative Lambek grammars are equivalent to it.

Comments on the Results and Further Directions

- Pentus's techniques cannot be generalized to LP (the binary reduction lemma fails).

Comments on the Results and Further Directions

- Pentus's techniques cannot be generalized to LP (the binary reduction lemma fails).
- The concept of IBVASSAM and the proofs are inspired by the methods used for hypergraph Lambek grammars (P. 2022). We claim that they could be useful for other kinds of categorial grammars.

Comments on the Results and Further Directions

- Pentus's techniques cannot be generalized to LP (the binary reduction lemma fails).
- The concept of IBVASSAM and the proofs are inspired by the methods used for hypergraph Lambek grammars (P. 2022). We claim that they could be useful for other kinds of categorial grammars.
- A more simple proof using closure properties?

Comments on the Results and Further Directions

- Pentus's techniques cannot be generalized to LP (the binary reduction lemma fails).
- The concept of IBVASSAM and the proofs are inspired by the methods used for hypergraph Lambek grammars (P. 2022). We claim that they could be useful for other kinds of categorial grammars.
- A more simple proof using closure properties?

Comments on the Results and Further Directions

- Pentus's techniques cannot be generalized to LP (the binary reduction lemma fails).
- The concept of IBVASSAM and the proofs are inspired by the methods used for hypergraph Lambek grammars (P. 2022). We claim that they could be useful for other kinds of categorial grammars.
- A more simple proof using closure properties?

Thank you for attention! Any questions?