On the notion of syntax concept lattice

Danya Rogozin

MSU

May, 2019

Structural rules

- In structural proof theory, one often have a deal with so-called logical and structural rules.
- Logical rule is a rule that allows one to manipulate with formulae and connections somehow.
- Structural rule is a rule that describes ways of manipulation with your assumptions, but it doesn't refer to any logical connective from considered language.

Structural rules

Let us consider main types of structural rules.

- Weakening is an inference rule that allows one to add new formulae into the context:

$$\frac{\Gamma \vdash A}{\Gamma, B \vdash A}\ \mathbf{W}$$

- Permutation (or, exchange): result of derivation doesn't depend on the order of assumptions.

$$\frac{\Gamma, A, B \vdash C}{\Gamma, B, A \vdash C}\ \mathbf{Ex}$$

- Contraction: if we have a formula repeating twice, we may remove one of them.

$$\frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\ \mathbf{C}$$

Structural rules

What if we reject one of these rules? If we lack one of them, then there is a substructural logic.

- If we reject weakening rule, we obtain so-called relevant logic widely used in philosophy and philosophical logic;
- Logic without weakening and contraction is a linear logic. Linear logic took its place in computer science and programming language design;
- What if we lack all of them? Non-commutative linear logic is logic without weakening, contraction, and exchange. What's the point?

See:

- Girard, Jean-Yves. (1987). Linear logic. Theoretical Computer Science, Volume 50, Issue 1, 1-101.
- Lambek, J. (1958). The mathematics of sentence structure. American mathematical monthly, vol. 65, No. 3, 154-170.

Categorial grammar

John runs.
$np \quad np\backslash s$
$\vdash np, np\backslash s \rightarrow s$

John loves Mary
$np \quad (np\backslash s)/np \quad np$
$\vdash np, (np\backslash s)/np, np \rightarrow s$

November spawned a monster
$np \quad (np\backslash s)/np \quad np/n \quad n$
$\vdash np, (np\backslash s)/np, np/n, n \rightarrow s$

Reduction rules in categorical grammar: $A, A\backslash B \rightarrow B$ and $B/A, A \rightarrow B$.

Categorial grammar

- Categorial grammar describes parsing in a natural language logically.
- On the other hand, we have context-free grammars (widely used in compiler development), where we may generate words (finite sequences of strings) via recursive rewriting rules
- Well, Lambek categorial grammars and context-free grammars are equivalent, that is, any Lamber categorial grammar is a context-free and vice versa.

See:
M. Pentus, "Lambek grammars are context free," [1993] Proceedings Eighth Annual IEEE Symposium on Logic in Computer Science, Montreal, Quebec, Canada, 1993, pp. 429-433.

But we have the same question. If categorical grammar is a logical approach to natural language, then how we can axiomatise inference rule for these grammars? And here comes the Lambek calculus.

Lambek calculus in itself

$$\frac{}{A \to A}\ \text{ax}$$

$$\frac{\Gamma \to A \qquad \Delta, B, \Theta \to C}{\Delta, \Gamma, A\backslash B, \Theta \to C}\ \backslash \to \qquad \frac{A, \Pi \to B}{\Pi \to A\backslash B} \to \backslash$$

$$\frac{\Gamma \to A \qquad \Delta, B, \Theta \to C}{\Delta, B/A, \Gamma, \Theta \to C}\ / \to \qquad \frac{\Pi, A \to B}{\Pi \to B/A} \to /$$

$$\frac{\Gamma, A, B, \Delta \to C}{\Gamma, A \bullet B, \Delta \to C}\ \bullet \to \qquad \frac{\Gamma \to A \qquad \Delta \to B}{\Gamma, \Delta \to A \bullet B} \to \bullet$$

Lambek calculus. Example

$$\Sigma = \{\text{November}, \text{spawned}, \text{a}, \text{monster}\}$$

$$\begin{aligned} v(\text{November}) &= np \\ v(\text{spawned}) &= (np\backslash s)/np \\ v(\text{a}) &= np/n \\ v(\text{monster}) &= n \end{aligned}$$

$$\dfrac{\dfrac{n \to n \qquad \dfrac{np \to np \qquad \dfrac{s \to s \qquad np, np\backslash s \to s}{np, (np\backslash s)/np, np \to s}/\to}{np, (np\backslash s)/np, np/n, n \to s}/\to}{np, (np\backslash s)/np, np/n, n \to s}\backslash\to$$

Thus, we proved that "November spawned a monster" is a sentence, that is, $v(\text{November spawned a monster}) = s$.

Lambek calculus models

Definition

Let $\mathcal{A} = \langle A, \cdot, \leqslant \rangle$ be an ordered semigroup (monoid), then $\mathcal{A}$ is a residual semigroup if there exists operation $\backslash$ (left division) and $/$ (right division), such that:

1. $(\forall a, b, c \in A)\ a \cdot b \leqslant c \Rightarrow a \leqslant b/c$;
2. $(\forall a, b, c \in A)\ a \cdot b \leqslant c \Rightarrow b \leqslant a\backslash b$

Theorem

1. *Logic L is a logic of all residuated semigroups;*
2. *Logic L* is a logic of all residuated monoids.*

Language models (L-модели)

Let $\mathcal{S} = \langle S, \cdot \rangle$ be a free semigroup. Let us consider the subsets of S, $\mathcal{P}(S)$, with the following operations:

- $A \bullet B = \{a \cdot b \mid a \in A \,\&\, b \in B\}$
- $A \backslash B = \{c \mid A \bullet \{c\} \subseteq B\}$
- $B / A = \{c \mid \{c\} \bullet A \subseteq B\}$

This residuated semigroup $\langle \mathcal{P}(S), \bullet, \backslash, / \rangle$ is called L-model.

Language models

Theorem

- *L is complete w.r.t L-models.*
- *L is complete w.r.t. L-models on infinitely generated semigroups.*
- *L is complete w.r.t. L-models on 2-generated semigroups.*

- Пентус М. Р. Полнота синтаксического исчисления Ламбека // Фундаментальная и прикладная математика. – 1999. – Т. 5, No 1. – С. 193—219.
- Buszkowski W. Completeness Results for Lambek Syntactic Calculus // Zeitschrift fur mathematische Logik und Grundlagen der Mathematik. – 1986. – Vol. 32. – P. 13-28.

What else?

One may extend Lambek calculus with so-called additive connections:

$$\frac{\Gamma, A_i, \Delta \to B}{\Gamma, A_1 \wedge A_2, \Delta \to B} \wedge \to, i = 1, 2 \qquad \frac{\Gamma \to A \qquad \Gamma \to B}{\Gamma \to A \wedge B} \to \wedge$$

$$\frac{\Gamma, A, \Delta \to C \qquad \Gamma, B, \Delta \to C}{\Gamma, A \vee B, \Delta \to C} \wedge \vee \qquad \frac{\Gamma \to A_i}{\Gamma \to A_1 \vee A_2} \to \vee$$

Informally, additive conjunction and disjunction are Boolean intersection and union of languages.

(Im)completeness

We have the following list of distributivities:

- $A \wedge (B \vee C) \to (A \wedge B) \vee (A \wedge C)$
- $(A \wedge B) \vee (A \wedge C) \to A \wedge (B \vee C)$
- $A \vee (B \wedge C) \to (A \vee B) \wedge (A \vee C)$
- $(A \vee B) \wedge (A \vee C) \to A \vee (B \wedge C)$

It is clearly, that these sequents are true in each L-model with intersection and union. On the other hand, the first one isn't provable in Lambek calculus. Thus, $L(\bullet, \backslash, /, \wedge, \vee)$ isn't complete w.r.t L-models with intersection and union.

Wurm models

If we want reasonable "linguistical" semantics for $L(\bullet, \backslash, /, \wedge, \vee)$, we should reconsider our approach to language models.
The solution was proposed by Christian Wurm.

See:

- Wurm, C. Language-Theoretic and Finite Relation Models for the (Full) Lambek Calculus. J of Log Lang and Inf (2017) 26: 179.

Wurm models

Definition

Let $\mathcal{L}$ be a finite alphabet and $L \subseteq \mathcal{L}^*$ be a language.

We define maps $[.]^{\triangleright} : \mathcal{P}(\mathcal{L}^*) \to \mathcal{P}(\mathcal{L}^* \times \mathcal{L}^*)$ and $[.]^{\triangleleft} : \mathcal{P}(\mathcal{L}^* \times \mathcal{L}^*) \to \mathcal{P}(\mathcal{L}^*)$ as follows:

1. $M \subseteq \mathcal{L}^*$, $M^{\triangleright} = \{(x, y) \mid \forall w \in M, xwy \in L\}$;
2. $C \subseteq \mathcal{L}^* \times \mathcal{L}^*$, $C^{\triangleleft} = \{w \mid \forall (x, y) \in C, xwy \in L\}$

Note that compositions $[.]^{\triangleleft\triangleright}$ and $[.]^{\triangleright\triangleleft}$ form closure operators, by the way $[.]^{\triangleleft}$ and $[.]^{\triangleright}$ are connected via contravariant Galois connection.

Wurm models

Definition

A syntactic concept is a pair $\langle S, C \rangle$*, where* $S \subseteq \mathcal{L}^*$ *and* $C \subseteq \mathcal{L}^* \times \mathcal{L}^*$*, such that* $S^{\triangleright} = C$ *and* $C^{\triangleleft} = S$*.*

Following to Wurm, by the concept we mean a closed set of strings, that is, A is a concept iff $A^{\triangleright\triangleleft} = A$. Moreover, $\langle \mathcal{B}_{\mathcal{L}}, \vee, \wedge \rangle$ is a complete lattice, where $\mathcal{B}_{\mathcal{L}}$ is the set of $\triangleright\triangleleft$-closed subsets of $\mathcal{L}^*$. Note that $A \vee B = (A \vee B)^{\triangleright\triangleleft}$

Furthermore, we define a product of concepts as $A \circ B = (A \cdot B)^{\triangleright\triangleleft} = \{ab \mid a \in A, b \in B\}^{\triangleright\triangleleft}$.

Wurm models

Residuals are defined explicitly as follows:

$$A\backslash B = \{(aB, b) \mid (a, b) \in A^{\triangleright}\}^{\triangleleft}$$
$$B/A = \{(a, Bb) \mid (a, b) \in A^{\triangleright}\}^{\triangleleft}$$

Definition

The syntax concept lattice of a language L is a structure $\langle \mathcal{B}_L, \circ, \bigvee, \bigwedge, /, \backslash \rangle$, where $\mathcal{B}_L$ is the set of $[.]^{\bowtie\triangleleft}$-closed subsets of $\mathcal{L}^*$.

Theorem

$L(\bullet, \backslash, /, \wedge, \vee)$ is complete w.r.t syntax concept lattices.

Lambek calculus and subexponentials

Let us consider these quite ordinary examples:

The girl whom Danya met (?) yesterday

The paper that Danya signed (?) without reading (?)

In these situations, one need to extract from the middle and multiply. In other words, we need to apply exchange and contraction in a limited way. Lambek calculus might be extended via subexponential modality for this purpose:

$$\frac{\Gamma, A, \Delta \to B}{\Gamma, !A, \Delta \to B}\, ! \to \qquad \frac{!\Gamma \to !A}{!\Gamma \to !A} \to !$$

$$\frac{\Gamma, !A, \Theta, \Delta \to B}{\Gamma, \Theta, !A, \Delta, \to B}\, \mathbf{ex} \qquad \frac{\Gamma, !A, \Theta, !A, \Delta \to B}{\Gamma, !A, \Theta, \Delta \to B}\, \mathbf{con}$$

Lambek calculus and subexponentials

Lambek calculus with additives and subexponentials is well studied from a proof-theoretical point of view, that is, such aspects as cut-elimination, etc. But semantics for noncommutative subexponentials has not implemented yet.

- Max Kanovich, Stepan Kuznetsov, Vivek Nigam, Andre Scedrov, "Subexponentials in non-commutative linear logic", Math. Structures Comput. Sci., 2018, 1–33.

Open question (my current task, as a matter of fact): provide the way to interpret noncommutative subexponentials within syntax concept lattices.

Thank you!