Adventures in Lambek Calculus

Max Kanovich, Stepan L. Kuznetsov, and Andre Scedrov

Logical Perspectives 2021

Introduction: Algebra of Formal Languages

- Let Σ be a finite alphabet.
- By Σ^+ we denote the set of all non-empty words over Σ.
- $\mathcal{P}(\Sigma^+)$ is the set of all formal languages over Σ without the empty word.
- We introduce the following algebraic operations on $\mathcal{P}(\Sigma^+)$:

$$\begin{aligned}
A \cdot B &= \{uv \mid u \in A, v \in B\} \\
A \setminus B &= \{u \in \Sigma^+ \mid (\forall v \in A)\; vu \in B\} \\
B \,/\, A &= \{u \in \Sigma^+ \mid (\forall v \in A)\; uv \in B\} \\
A \vee B &= A \cup B; \quad A \wedge B = A \cap B
\end{aligned}$$

- The most interesting operations are two divisions, $\setminus$ and $/$. They are connected to product in the following way:

$$B \subseteq A \setminus C \iff A \cdot B \subseteq C \iff A \subseteq C \,/\, B$$

Relational Algebras

- Another class of algebraic structures we are going to keep in mind is formed by the *algebras of binary relations.*
- Let W be a non-empty set. Fix a transitive binary relation $U \subseteq W \times W$, which we shall call the "universal" one.
- We take $\mathcal{P}(U)$, the set of all subrelations of U, and introduce algebraic operations in the same signature as on $\mathcal{P}(\Sigma^+)$:

$$
\begin{aligned}
&R \cdot S = R \circ S \\
&R \setminus S = \{\langle y, z\rangle \in U \mid (\forall \langle x, y\rangle \in R)\ \langle x, z\rangle \in S\} \\
&S \,/\, R = \{\langle x, y\rangle \in U \mid (\forall \langle y, z\rangle \in R)\ \langle x, z\rangle \in S\} \\
&R \vee S = R \cup S; \quad R \wedge S = R \cap S
\end{aligned}
$$

- Again,

$$S \subseteq R \setminus T \iff R \cdot S \subseteq T \iff R \subseteq T \,/\, S$$

Residuated Lattices

- Both algebras of languages and relational algebras are special kinds of a more general class of algebraic structures, *residuated lattices.*
- A residuated lattice is a tuple $\mathfrak{A} = (\mathcal{A}, \preceq, \cdot, \backslash, /, \vee, \wedge)$, where:
  - $\preceq$ is a partial order which forms a lattice, $\vee$ and $\wedge$ are lattice join and meet;
  - $(\mathcal{A}, \cdot)$ is a semigroup;
  - $b \preceq a \backslash c \iff a \cdot b \preceq c \iff a \preceq c \,/\, b$, for any $a, b, c \in \mathcal{A}$.
- Residuated lattices give algebraic semantics to substructural logics, like, for example, Heyting algebras do for intuitionism.

  N. Galatos, P. Jipsen, T. Kowalski, H. Ono. Residuated Lattices: An Algebraic Glimpse at Substructural Logics. Springer, 2007.
- The logic of residuated lattices is the multiplicative-additive Lambek calculus.

Multiplicative-Additive Lambek Calculus (MALC)

$$\frac{}{A \vdash A}\ \mathrm{Id}$$

$$\frac{\Pi \vdash A \quad \Gamma, B, \Delta \vdash C}{\Gamma, \Pi, A \setminus B, \Delta \vdash C}\ \mathrm{L}\setminus \qquad \frac{A, \Pi \vdash B}{\Pi \vdash A \setminus B}\ \mathrm{R}\setminus \quad (\Pi \text{ is not empty})$$

$$\frac{\Pi \vdash A \quad \Gamma, B, \Delta \vdash C}{\Gamma, B \,/\, A, \Pi, \Delta \vdash C}\ \mathrm{L}/ \qquad \frac{\Pi, A \vdash B}{\Pi \vdash B \,/\, A}\ \mathrm{R}/ \quad (\Pi \text{ is not empty})$$

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \cdot B, \Delta \vdash C}\ \mathrm{L}\cdot \qquad \frac{\Pi_1 \vdash A \quad \Pi_2 \vdash B}{\Pi_1, \Pi_2 \vdash A \cdot B}\ \mathrm{R}\cdot$$

$$\frac{\Gamma, A, \Delta \vdash C \quad \Gamma, B, \Delta \vdash C}{\Gamma, A \vee B, \Delta \vdash C}\ \mathrm{L}\vee \qquad \frac{\Pi \vdash A}{\Pi \vdash A \vee B} \quad \frac{\Pi \vdash B}{\Pi \vdash A \vee B}\ \mathrm{R}\vee$$

$$\frac{\Gamma, A, \Delta \vdash C}{\Gamma, A \wedge B, \Delta \vdash C} \quad \frac{\Gamma, B, \Delta \vdash C}{\Gamma, A \wedge B, \Delta \vdash C}\ \mathrm{L}\wedge \qquad \frac{\Pi \vdash A \quad \Pi \vdash B}{\Pi \vdash A \wedge B}\ \mathrm{R}\wedge$$

Multiplicative-Additive Lambek Calculus (MALC)

- The cut rule of the following form:

$$\frac{\Pi \vdash A \quad \Gamma, A, \Delta \vdash C}{\Gamma, \Pi, \Delta \vdash C} \ \mathrm{Cut}$$

  is admissible in MALC.

- As said above, algebraic models of MALC are *residuated lattices:* variables and formulae are interpreted as elements of $\mathcal{A}$, and a sequent $A_1, \ldots, A_n \vdash B$ is interpreted as $A_1 \cdot \ldots \cdot A_n \preceq B$.
- Models on algebras of formal languages and models on relational algebras are called *L-models* and *R-models* respectively.
- MALC can be also viewed as a non-commutative intuitionistic version of linear logic (J.-Y. Girard, 1987). This was noticed by V. M. Abrusci (1990).

Lambek Categorial Grammars

- The original motivation for the Lambek calculus is its usage for describing natural language syntax (J. Lambek, 1958).
- This usage is connected to L-models.
- For each letter $a \in \Sigma$ the grammar associates one or more syntactic types, which are formulae of the Lambek calculus: $a \triangleright A$.
- A word $a_1 \dots a_n$ is considered grammatically correct, if the corresponding sequent $A_1, \dots, A_n \vdash s$ is derivable.
- The standard example is "John loves Mary," with the corresponding sequent $np, (np \setminus s) / np, np \vdash s$.

Part I: Distributivity

- Both L-models and R-models are distributive (as lattices): $(A \wedge B) \vee C \equiv (A \vee C) \wedge (B \vee C)$.
- In general, however, residuated lattices can be non-distributive.
- Thus, $(A \vee C) \wedge (B \vee C) \vdash (A \wedge B) \vee C$ is not derivable MALC, which prevents the latter from being L-complete or R-complete.
- Indeed, if this sequent were derivable, then it would be true in **all** residuated lattices, which would make them all distributive (which is not the case).
- There exists a natural, non-distributive modification of L-models which avoids this problem and gains completeness (C. Wurm 2017).

Partial Completeness Results

- L$\wedge$, i.e., MALC without $\vee$, is R-complete (H. Andréka & Sz. Mikulás 1994)
- The Lambek calculus without $\vee$ and $\wedge$, is L-complete (M. Pentus 1995)
- L$(\backslash, /, \wedge)$, that is, MALC with only three connectives: $\backslash, /, \wedge$, is L-complete (W. Buszkowski 1982)
- Open question: L-completeness of L$\wedge$ (i.e., MALC without $\vee$).
- It is also unknown whether adding distributivity as an extra axiom yields completeness.
- *We show that the situation with* L$\vee$ (i.e., MALC without $\wedge$) *is different from the one with* L$\wedge$.

Distributivity without $\wedge$

- **Theorem.** The sequent

$$((x \,/\, y) \vee x) \,/((x \,/\, y) \vee (x \,/\, z) \vee x), (x \,/\, y) \vee x,$$
$$((x \,/\, y) \vee x) \backslash ((x \,/\, z) \vee x) \vdash (x \,/(y \vee z)) \vee x$$

  is not derivable in L$\vee$, but can be derived using the distributivity axiom (and cut).
- Thus, L$\vee$ is neither L-complete nor R-complete (because L-models and R-models are distributive).

How to Guess the Sequent?

- **Lemma.** If $A \vdash D$ and $B \vdash D$ are derivable (join), then for $C = (A \,/\, D) \cdot A \cdot (A \setminus B)$ we have $C \vdash A$ and $C \vdash B$ (meet). (see Lambek 1958, Pentus 1994)
- In particular, $C = (A \,/ (A \vee B)) \cdot A \cdot (A \setminus B)$ is a meet for A and B.
- Take $A = (x \,/\, y) \vee x$ and $B = (x \,/\, z) \vee x$.
- By distributivity,

$$((x \,/\, y) \vee x) \wedge ((x \,/\, z) \vee x) \vdash ((x \,/\, y) \wedge (x \,/\, z)) \vee x$$

- The succedent is equivalently replaced by $(x \,/ (y \vee z)) \vee x$.
- The antecedent is replaced by a stronger meet $C = (A \,/ (A \vee B)) \cdot A \cdot (A \setminus B)$ (it is stronger, since $C \vdash A$, $C \vdash B$, thus $C \vdash A \wedge B$).
- This yields, using cut, derivability of our sequent in the presence of distributivity.

Proving Non-Derivability in L$\vee$

- Non-derivability of our sequent in L$\vee$ does *not* come automatically from non-derivability of the distributivity law, since our new meet C is stronger than $A \wedge B$.
- However, the derivability problem is decidable, so we can just use derivability-checking software (developed by P. Jipsen, available online), which gives the answer in several seconds.
- In our WoLLIC 2019 paper, we also do manual proof search.
- One can also construct an algebraic countermodel (shorter, but requires some creativity).

Commutative and Affine Generalizations

- Adding the permutation rule of the following form

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}\ \mathrm{P}$$

  to MALC (that is, making things commutative) gives the multiplicative-additive fragment of intuitionistic linear logic (ILL).
- If one additionally adds weakening

$$\frac{\Gamma, \Delta \vdash C}{\Gamma, A, \Delta \vdash C}\ \mathrm{W}$$

  this will give the multiplicative-additive fragment of intuitionistic affine logic (IAL).

Commutative and Affine Generalizations

- The sequent

$$((x \,/\, y) \vee x) \,/((x \,/\, y) \vee (x \,/\, z) \vee x), (x \,/\, y) \vee x, \\ ((x \,/\, y) \vee x) \backslash ((x \,/\, z) \vee x) \vdash (x \,/(y \vee z)) \vee x$$

  is still not derivable if we add commutativity (permutation rule), that is, in ILL.
- For the affine case (IAL, with weakening rule), the sequent should be slightly modified

$$((x \,/\, y) \vee w) \,/((x \,/\, y) \vee (x \,/\, z) \vee w), (x \,/\, y) \vee w, \\ ((x \,/\, y) \vee w) \backslash ((x \,/\, z) \vee w) \vdash (x \,/(y \vee z)) \vee w$$

Part II: Systems with the Unit

- In intuitionistic linear logic, the unit constant (multiplicative truth) is axiomatized as follows:

$$\frac{\Gamma, \Delta \vdash C}{\Gamma, 1, \Delta \vdash C}\ \mathrm{L}1 \qquad \frac{}{\vdash 1}\ \mathrm{R}1$$

- Thus, adding 1 requires abolishing antecedent non-emptiness restriction.
- In residuated lattices, this corresponds to moving from arbitrary semigroups (recall that, in any residuated lattice, $(\mathcal{A}, \cdot)$ is a semigroup) to monoids: $(\mathcal{A}, \cdot, 1)$.
- In particular, we modify the definition of L-models by allowing the empty word in languages.

Undecidability with the Unit

- The multiplicative unit constant, 1, is necessarily interpreted in L-models as $\{\varepsilon\}$ (due to $A \cdot 1 \vdash A$).
- Axiomatising the unit as multiplicative truth in linear logic yields incomplete systems: for example, $(1 \wedge G) \cdot F \equiv F \cdot (1 \wedge G)$ is true in L-models, but not derivable in non-commutative linear logic.
- We present a minimal system $\mathrm{L}^{+\varepsilon}(\backslash, \wedge, 1)$, which captures the following L-correct principles: $A \cdot \{\varepsilon\} = \{\varepsilon\} \cdot A$ ("commuting") and $\{\varepsilon\} \cdot \{\varepsilon\} = \{\varepsilon\}$ ("doubling").
- Notice that it is in the language of $\backslash, \wedge, 1$ only.

$\mathsf{L}^{+\varepsilon}(\backslash, \wedge, 1)$

$$\frac{}{A \vdash A}\ \mathrm{Id} \qquad \frac{}{A, 1 \vdash A}\ 1$$

$$\frac{\Pi \vdash A \quad \Gamma, B, \Delta \vdash C}{\Gamma, \Pi, A \backslash B, \Delta \vdash C}\ \mathrm{L}\backslash \qquad \frac{A, \Pi \vdash B}{\Pi \vdash A \backslash B}\ \mathrm{R}\backslash$$

$$\frac{\Gamma, A, \Delta \vdash C}{\Gamma, A \wedge B, \Delta \vdash C} \quad \frac{\Gamma, B, \Delta \vdash C}{\Gamma, A \wedge B, \Delta \vdash C}\ \mathrm{L}\wedge \qquad \frac{\Pi \vdash A \quad \Pi \vdash B}{\Pi \vdash A \wedge B}\ \mathrm{R}\wedge$$

$$\frac{\Gamma, A, (1 \wedge G), \Delta \vdash C}{\Gamma, (1 \wedge G), A, \Delta \vdash C}\ \mathrm{L}\varepsilon \qquad \frac{\Gamma, (1 \wedge G), A, \Delta \vdash C}{\Gamma, A, (1 \wedge G), \Delta \vdash C}\ \mathrm{R}\varepsilon$$

$$\frac{\Gamma, (1 \wedge G), (1 \wedge G), \Delta \vdash C}{\Gamma, (1 \wedge G), \Delta \vdash C}\ \mathrm{D}\varepsilon$$

- **Theorem.** Any system which includes $\mathsf{L}^{+\varepsilon}(\backslash, \wedge, 1)$ and is L-sound is undecidable.
- In particular, so is the set of all L-true sequents, but for this set we do not even know whether it is r.e.

Undecidability Proof Sketch

- We encode 2-counter Minsky machines.
- The direction from computations to derivations is established by constructing the corresponding proofs in $\mathrm{L}^{+\varepsilon}(\backslash, \wedge, 1)$.
- The backwards direction is performed via L-models.

Encoding Minsky Machines

- Atoms (propositional variables): e_1, e_2 (start/end markers); p_1, p_2 (the number of p_i's is the value of counter c_i); $\ell_0, \ell_1, \ldots$ (states of the machine); b.
- If the machine is in state L_i, with $c_1 = k_1$ and $c_2 = k_2$, then it is encoded as follows:

$$e_1, \underbrace{p_1, \ldots, p_1}_{k_1 \text{ times}}, \ell_i, \underbrace{p_2, \ldots, p_2}_{k_2 \text{ times}}, e_2$$

Encoding Minsky Machines

- Each instruction I of the machine is encoded by the corresponding formula A_I
  $(F^{bb} = (F \setminus b) \setminus b$ is the pseudo-double-negation):

I	A_I
$L_i : inc(c_1);\ goto\ L_j;$	$\ell_i \backslash (p_1 \cdot \ell_j)^{bb}$
$L_i : inc(c_2);\ goto\ L_j;$	$\ell_i \backslash (\ell_j \cdot p_2)^{bb}$
$L_i : dec(c_1);\ goto\ L_j;$	$(p_1 \cdot \ell_i) \setminus \ell_j^{bb}$
$L_i : dec(c_2);\ goto\ L_j;$	$(\ell_i \cdot p_2) \setminus \ell_j^{bb}$
$L_i : if(c_1 = 0)\ goto\ L_j;$	$(e_1 \cdot \ell_i) \backslash (e_1 \cdot \ell_j)^{bb}$
$L_i : if(c_2 = 0)\ goto\ L_j;$	$(\ell_i \cdot e_2) \backslash (\ell_j \cdot e_2)^{bb}$

Encoding Minsky Machines

- All operations are encoded into a leading $1 \wedge G$, where G is a big conjunction.
- G includes the following formulae:
  - A_I for each instruction I of our Minsky machine. Each A_I is of the form $g_{\alpha,\beta} = \beta \setminus \alpha^{bb}$.
  - $g_{\xi,\xi} = \xi \setminus \xi^{bb}$ for each atom ξ.
  - $(e_1 \cdot \ell_0 \cdot e_2) \setminus b$, for terminating computation. (L_0 is the final state, and the counters are required to be zero.)

Lemma

The sequent $1 \wedge G, \Delta \vdash b$, where Δ encodes the initial configuration of the machine, is derivable in $\mathrm{L}^{+\varepsilon}(\setminus, \wedge, 1)$ if and only if the machine reaches state L_0 with zero counters, starting from this initial configuration.

From Computations to Derivations

- Since G includes $g_{\alpha,\beta} = (\beta \setminus \alpha^{bb})$, then derivability of $1 \wedge G, \alpha, \Delta \vdash b$ yields derivability of $1 \wedge G, \Delta, \beta \vdash b$.
  - This enables Minsky commands, but only on the left side of the configuration.
  - This derivation essentially uses "doubling."
- Cyclic transpositions. If G includes $g_{\xi,\xi} = \xi \setminus \xi^{bb}$ for any atom ξ (which is the case), and Δ_1, Δ_2 are all atomic, then derivability of $1 \wedge G, \Delta_1, \Delta_2 \vdash b$ yields derivability of $1 \wedge G, \Delta_2, \Delta_1 \vdash b$.
  - This allows locating $1 \wedge G$ near the necessary place in the configuration.
- Finally, we have $(e_1 \cdot \ell_0 \cdot e_2) \setminus b$ in G.
  - This encodes the finish of computation, $(L_0, 0, 0)$.

From Derivations to Computations

- Let Σ (alphabet) include all atoms.
- Let B_M be the set of "terminating strings," that is, codes of configurations of the Minsky machine M, such that the machine, starting from this configuration, reaches the terminating one $(L_0, 0, 0)$.
- Consider the following L-interpretation:

$$w(a) = \begin{cases} \{a\}, & \text{if } a \neq b \\ \{xy \mid yx \in B_M\}, & \text{for } a = b \end{cases}$$

- **Lemma.** For any instruction I of M, $w(A_I) \ni \varepsilon$. Hence, $w(1 \wedge G) = \{\varepsilon\}$.
- If $1 \wedge G, e_1, \underbrace{p_1, \ldots, p_1}_{k_1 \text{ times}}, \ell_i, \underbrace{p_2, \ldots, p_2}_{k_2 \text{ times}} \vdash b$ is derivable, then interpretation of the antecedent is in $w(b)$, whence the configuration (L_i, k_1, k_2) terminates to $(L_0, 0, 0)$.

Models on Regular Languages

- Recall that the class of regular languages is the minimal class of languages including $\varnothing$, $\{\varepsilon\}$, singletons $\{a\}$ for any $a \in \Sigma$, and closed under language multiplication, union, and iteration (Kleene star): $A^* = \{\varepsilon\} \cup A \cup (A \cdot A) \cup (A \cdot A \cdot A) \cup \ldots$
- A specific class of L-models includes only models in which all languages are regular.
- We shall call such models $\mathrm{L}\textsc{reg}$-models.
- This definition is consistent, since the class of regular languages is closed under Lambek operations.
- Without the unit constant 1, the calculus $\mathrm{L}(\backslash, /, \wedge)$ is complete w.r.t. $\mathrm{L}\textsc{reg}$-models (this follows from Buszkowski's and Sorokin's work).

Models on Regular Languages

- The situation changes if we add the unit.
- We still consider theories in the language of MALC with the unit constant.
- As shown by the encoding of Minsky machines above, the theory of **all** L-models in the language of $\backslash, \wedge, 1$ is undecidable; more precisely—Σ_1^0-hard.
- **NB:** we do not claim that it belongs to Σ_1^0, it could be harder!
- On the other hand, the theory of the **subclass** of LREG-models belongs to the Π_1^0 class.
- Indeed, we now have to quantify over regular languages, that is, over regular expressions. This yields an *arithmetical* universal quantifier, thus Π_1^0.

Models on Regular Languages

- Since no Σ_1^0-hard language can belong to Π_1^0, we get the following

Theorem

The theories of L-models and of $\textsc{Lreg}$*-models, in the language of* $\backslash, \wedge, 1$*, are different.*

○ NO regular model property

Theorem 1.1 We can find a sequent of the specific form

$$\mathbb{1} \wedge G,\ \ \mathbf{\Delta} \vdash b$$

so that

(a) We can construct an L-model such that the sequent is not valid in the model,

(b) But the sequent is valid in any LREG-models.

∘ The minimalistic propositional systems that are still PSPACE-complete

Main Complexity Results:

Commutative (Linear logic)	Non-commutative (Lambek, circular)
$\mathcal{L}^1(\backslash)$ is NP-complete (Kanovich)	$\mathcal{L}^1(\backslash)$ is polytime (Savateev)
$\mathcal{L}^1(\backslash, \wedge)$ is PSPACE-complete	$\mathcal{L}^1(\backslash, \wedge)$ is PSPACE-complete
$\mathcal{L}^1(\backslash, \vee)$ is PSPACE-complete	$\mathcal{L}^1(\backslash, \vee)$ is PSPACE-complete

One implication, one conjunction or one disjunction. Here $\mathcal{L}^1(\backslash)$, $\mathcal{L}^1(\backslash, \wedge)$, etc., denote fragments with **only one** variable.

Soft Subexponentials and Multiplexing

Max Kanovich, Stepan Kuznetsov , **Vivek Nigam** and Andre Scedrov

Logical Frameworks

Logical Specifications allow for the specification of deductive systems, logics, and operational semantics.

- **Linear Logical Frameworks**: Specify state conscious systems;

$$\Theta; \Gamma \longrightarrow G$$

Unbounded Formulas: Interpreted as a **set** of formulas.

Linear Formulas: Interpreted as a **multiset** of formulas.

Logical Frameworks

Two extensions of **Linear Logical Frameworks**:

Subexponentials

[Nigam,Olarte,Pimentel, Reis]

$$\Theta_1; \ldots ; \Theta_n ; \Gamma_1 ; \ldots ; \Gamma_m \longrightarrow G$$

Allows for **many unbounded and linear contexts**.

- **Extended expressiveness:** specificaiton of systems with several contexts: logics, concurrent programming, etc.

Ordered Logics

[Pfenning,Simmons,Polakow]

$$\Theta; \Gamma; L \longrightarrow G$$

L - **Ordered Formulas:** Interpreted as a **list** of formulas.

- **Extended expressiveness:** specificaiton of systems with some order (PL evaluation strategies, systems with lists, etc.)

Contribution 1: A logical framework with commutative and non-commutative subexponentials.

Proof System with Subexponentials

Subexponential Signature

$$\Sigma = \langle \mathcal{I}, \preceq, \mathcal{W}, \mathcal{C}, \mathcal{E} \rangle$$

SNILL_Σ proof system.

- $\mathcal{I}$ is a set of lables, $\mathcal{W}, \mathcal{C}, \mathcal{E} \subseteq \mathcal{I}$
- $\preceq$ is a pre-order relation over $\mathcal{I}$ upwardly closed w.r.t. $\mathcal{W}, \mathcal{C}, \mathcal{E}$.

For each $\mathsf{s} \in \mathcal{I}$:

$$\frac{\Gamma_1, F, \Gamma_2 \to G}{\Gamma_1, !^{\mathsf{s}} F, \Gamma_2 \to G}\ Der \qquad \frac{!^{\mathsf{s}_1} F_1, \ldots, !^{\mathsf{s}_n} F_n \longrightarrow F}{!^{\mathsf{s}_1} F_1, \ldots, !^{\mathsf{s}_n} F_n \longrightarrow !^{\mathsf{s}} F}\ !^{\mathsf{s}}{}_R, \text{provided, } \mathsf{s} \preceq \mathsf{s}_i, 1 \leq i \leq n$$

For each $\mathsf{w} \in \mathcal{W}$ and $\mathsf{c} \in \mathcal{C}$:

$$\frac{\Gamma, \Delta \longrightarrow G}{\Gamma, !^{\mathsf{w}} F, \Delta \longrightarrow G}\ W \qquad \frac{\Gamma_1, !^{\mathsf{c}} F, \Delta, !^{\mathsf{c}} F, \Gamma_2 \to G}{\Gamma_1, !^{\mathsf{c}} F, \Delta, \Gamma_2 \to G}\ C_1 \qquad \frac{\Gamma_1, !^{\mathsf{c}} F, \Delta, !^{\mathsf{c}} F, \Gamma_2 \to G}{\Gamma_1, \Delta, !^{\mathsf{c}} F, \Gamma_2 \to G}\ C_2$$

For each $\mathsf{e} \in \mathcal{E}$:

$$\frac{\Gamma_1, \Delta, !^{\mathsf{e}} F, \Gamma_2 \to C}{\Gamma_1, !^{\mathsf{e}} F, \Delta, \Gamma_2 \to C}\ E_1 \qquad \frac{\Gamma_1, !^{\mathsf{e}} F, \Delta, \Gamma_2 \to C}{\Gamma_1, \Delta, !^{\mathsf{e}} F, \Gamma_2 \to C}\ E_2$$

Application: Type-Logical Grammar

Assign logical formulas (or types) to sentences.

$$\begin{array}{ccc} & N \backslash S / N & \\ \text{"John} & \text{loves} & \text{Mary."} \\ N & & N \end{array}$$

$$\dfrac{N \to N \qquad \dfrac{N \to N \quad S \to S}{N, N \backslash S \to S}}{N, N \backslash S / N, N \to S}$$

The proof of formulas for sentences **may have contraction**: parasitic extraction.

"John signed the paper without reading it"

"The paper that John signed without reading."

"It" has been omitted twice.

Application: Type-Logical Grammar

"The paper that John signed without reading."

Contraction to fill the gap.

$$
\dfrac{\dfrac{\dfrac{\dfrac{N, N \backslash S / N, N, (N \backslash S) \backslash (N \backslash S) / GC, GC / N, N \to S}{N, N \backslash S / N, !^{\mathsf{s}}N, (N \backslash S) \backslash (N \backslash S) / GC, GC / N, !^{\mathsf{s}}N \to S}\,Der}{N, N \backslash S / N, (N \backslash S) \backslash (N \backslash S) / GC, GC / N, !^{\mathsf{s}}N \to S}\,C_L}{N, N \backslash S / N, (N \backslash S) \backslash (N \backslash S) / GC, GC / N \to S / !^{\mathsf{s}}N} \qquad N / CN, CN, CN \backslash CN \to N}{N / CN, CN, (CN \backslash CN) / (S / !^{\mathsf{s}}N), N, N \backslash S / N, (N \backslash S) \backslash (N \backslash S) / GC, GC / N \to N}
$$

Type-Logical Grammars

Our previous work [**IJCAR18**] proposed a Subexponential Non-Commutative Linear Logical Framework for Type-Logical Grammars (and distributed systems):

"The paper that John signed without reading."

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{N, N \setminus S \,/\, N, N, (N \setminus S) \setminus (N \setminus S) \,/\, GC, GC \,/\, N, N \rightarrow S}
{N, N \setminus S \,/\, N, !^{\mathsf{s}}N, (N \setminus S) \setminus (N \setminus S) \,/\, GC, GC \,/\, N, !^{\mathsf{s}}N \rightarrow S}\ \mathit{Der}
}
{N, N \setminus S \,/\, N, (N \setminus S) \setminus (N \setminus S) \,/\, GC, GC \,/\, N, !^{\mathsf{s}}N \rightarrow S}\ C_L
}
{N, N \setminus S \,/\, N, (N \setminus S) \setminus (N \setminus S) \,/\, GC, GC \,/\, N \rightarrow S \,/\, !^{\mathsf{s}}N}
\qquad
N \,/\, CN, CN, CN \setminus CN \rightarrow N
}
{N \,/\, CN, CN, (CN \setminus CN) \,/\, (S \,/\, !^{\mathsf{s}}N), N, N \setminus S \,/\, N, (N \setminus S) \setminus (N \setminus S) \,/\, GC, GC \,/\, N \rightarrow N}
$$

Type-Logical Grammars

However, as shown recently [**JLC 2020**], our proposed logical framework does not satisfy **Lambek's Non Emptiness Property**.

"All sequent antecendents shall not be empty."

This means that our previous logical framework may type sentences that are not grammatically correct.

very interesting book $\xrightarrow{\text{corresponds}}$ $(N \,/\, N) \,/ (N \,/\, N), N \,/\, N, N \rightarrow N$

very book $\xrightarrow{\text{corresponds}}$ $(N \,/\, N) \,/ (N \,/\, N), N \rightarrow N$

Does not satisfy Lambek's Non-Emptiness Property

Key Inspiration

Subexponentials

$!$ is non-commutative

Two types of subexponentials $!$ and ∇

from Soft Linear Logic

- Multiplexing rule instead of contraction:

$$\frac{\Gamma, F, \ldots, F, \Delta \to G}{\Gamma, !F, \Delta \to G}\ !_L$$

from Light Linear Logic

$$\frac{F \to G}{!F \to !G}\ !_R \qquad \frac{F \to G}{\nabla F \to \nabla G}\ \nabla_R$$

Exactly one formula in the antecedent.

Our proposed non-commutative logical framework SLLM contains subexponentials with behavior from soft and light linear logics..

Lambek System with Nonemptiness [1958]

$$\frac{}{F \to F}\; I$$ Initial

$$\frac{\Pi \to G \quad \Gamma_1, F, \Gamma_2 \to C}{\Gamma_1, F\,/\,G, \Pi, \Gamma_2 \to C}\; /_L \qquad \frac{\Pi, F \to G}{\Pi \to G\,/\,F}\; /_R, \text{ where } \Pi \neq \emptyset$$ Right Division

$$\frac{\Pi \to F \quad \Gamma_1, G, \Gamma_2 \to C}{\Gamma_1, \Pi, F \setminus G, \Gamma_2 \to C}\; \setminus_L \qquad \frac{F, \Pi \to G}{\Pi \to F \setminus G}\; \setminus_R, \text{ where } \Pi \neq \emptyset$$ Left Division

$$\frac{\Gamma_1, F, G, \Gamma_2 \to C}{\Gamma_1, F \cdot G, \Gamma_2 \to C}\; \cdot_L \qquad \frac{\Gamma_1 \to F \quad \Gamma_2 \to G}{\Gamma_1, \Gamma_2 \to F \cdot G}\; \cdot_R$$ Product

$$\frac{\Pi \to F\{e/x\}}{\Pi \to \forall x.F}\; \forall_R \qquad \frac{\Gamma_1, F\{t/x\}, \Gamma_2 \to C}{\Gamma_1, \forall x.F, \Gamma_2 \to C}\; \forall_L$$ Quantifier

The order of formulas is important.

Subexponentials

Two subexponentials: $!$ and ∇.

$k > 0$

$$\frac{\Gamma_1, \overbrace{A, A, \ldots, A}^{k \text{ times}}, \Gamma_2 \to C}{\Gamma_1, !A, \Gamma_2 \to C}\ !_L\ \underline{(k \geq 1)} \qquad \frac{A \to C}{!A \to !C}\ !_R$$

No weakening, no contraction and no exchange

$$\frac{\Gamma_1, A, \Gamma_2 \to C}{\Gamma_1, \nabla A, \Gamma_2 \to C}\ \nabla_L \qquad \frac{A \to C}{\nabla A \to \nabla C}\ \nabla_R$$

$$\frac{\Gamma_1, \Gamma_2, \nabla A, \Gamma_3 \to C}{\Gamma_1, \nabla A, \Gamma_2, \Gamma_3 \to C} \qquad \frac{\Gamma_1, \nabla A, \Gamma_2, \Gamma_3 \to C}{\Gamma_1, \Gamma_2, \nabla A, \Gamma_3 \to C}\ \nabla_E$$

No weakening and no contraction.

Basic Properties

Theorem

- The calculus SLLM enjoys admissibility of the Cut Rule.
- Given an atomic A and sequent $\Gamma(A) \longrightarrow C(A)$ derivable in SLLM, then for any formula B, $\Gamma(B) \longrightarrow C(B)$ is also derivable in SLLM.

What if we take a more general rule: $$\frac{\Gamma \to C}{!\Gamma \to !C}\ !_R$$

$$\dfrac{\dfrac{B, B \setminus C \to C}{!B, !(B \setminus C) \to !C} \qquad !C \to C \cdot C}{!B, !(B \setminus C) \to C \cdot C}\ Cut$$

No cut-free proof.

Examples

The letter that Bob sent yesterday.

"Bob sent the letter yesterday"

$$\dfrac{\dfrac{\dfrac{N, V\,/\,N, N, V\,\backslash\,V \to S}{\nabla N, N, V\,/\,N, V\,\backslash\,V \to S}\;\nabla_L, \nabla_E}{N, V\,/\,N, V\,\backslash\,V \to S'} \qquad N, N\,\backslash\,N \to N}{N, (N\,\backslash\,N)\,/\,S', N, V\,/\,N, V\,\backslash\,V \to N}$$

$$S' = (\nabla N\,\backslash\,S)$$

Examples

The letter that Bob sent without reading.

"Bob sent the letter without reading it"

$$\dfrac{\dfrac{\dfrac{\dfrac{\Delta_1, N, \Delta_2, N \longrightarrow S}{\nabla N, \nabla N, \Delta_1, \Delta_2 \longrightarrow S}\ \nabla_L, \nabla_E}{!\nabla N, \Delta_1, \Delta_2 \longrightarrow S}\ !_L}{\Delta_1, \Delta_2 \longrightarrow S''} \qquad N, N \setminus N \longrightarrow N}{N, (N \setminus N) \,/\, S'', \Delta_1, \Delta_2 \longrightarrow N}$$

$$S' = (!\nabla N \setminus S)$$

Lambek's non-emptiness restriction

Theorem

- The calculus SLLM provides Lambek's non-emptiness restriction: If a sequent $\Gamma \longrightarrow C$ is provable, then list Γ is not empty.

- No weakening.
- The introduction of $!$ or ∇ never produces the empty list.

Focused Proof System

In the paper, we also propose a **focused proof system** for SLLM, thus enabling proof search.

Our previous work **[IJCAR 2018]** left open how to design focused proof system for subexponentials that do not allow both weakening and exchange.

Key Challenge

$$\frac{\Gamma, F, \ldots, F, \Delta \to G}{\Gamma, !F, \Delta \to G}$$

This rule has a great deal of non-determinism as one has to decide how many copies of F appears in the premise.

Solution Idea

Introduce two new modalities:

Formula is used once or more times

Formula is used zero or more times

$$\frac{\Gamma, \sharp^+ F, \Delta}{\Gamma, !F, \Delta \to G} \qquad \frac{\Gamma, F, \sharp^* F, \Delta \to G}{\Gamma, \sharp^+ F, \Delta \to G} \qquad \frac{\Gamma, F, \Delta \to G}{\Gamma, \sharp^* F, \Delta \to G} \qquad \frac{\Gamma, \Delta \to G}{\Gamma, \sharp^* F, \Delta \to G}$$

Structural rules are incorporated into the introduction rules:

$$\frac{\sharp^* C, \Gamma_2 \to F \quad \Gamma_1, \sharp^+ C, G, \Gamma_3 \to H}{\Gamma_1, \sharp^+ C, \Gamma_2, F \setminus G, \Gamma_3 \to H} \qquad \frac{F \to G}{\Gamma_1^*, !F, \Gamma_2^* \to !G} \qquad \frac{F \to G}{\Gamma_1^*, \nabla F, \Gamma_2^* \to \nabla G}$$

Theorem

- Let Γ, G be formulas not containing $\#^+, \#^*$. A sequent $\Gamma \longrightarrow G$ is provable in SLLM# if and only if it is provable in SLLM.

Complexity

Provability in SLLM is undecidable in general.

Encoding of Turing Machines (TMs):

- A Turing Machine configuration is encoded in the sequent context:

  $[B_1, q_1, \xi, B_2]$ $\longrightarrow$ TM state q_1, tape with B_1, ξ, B_2, and head looking at ξ.

- An instruction $I : q\xi \rightarrow q'\eta R$, for example, is encoded as the formula $!\nabla[(q \cdot \xi) \setminus (\eta \cdot q')]$:

  $\longrightarrow$ The prefix enables the instruction to be used multiple times at any place of the tape.

- Strong correspondence (level of proofs): A deterministic TM M leads to a final configuration using instructions $I_1, \ldots, I_m$ only iff the following sequent is derivable in SLLM:

$$!\nabla A_{I_1},\ !\nabla A_{I_2},\ \ldots,\ !\nabla A_{I_m},\ B_1 \cdot q_1 \cdot \xi \cdot B_2 \ \longrightarrow\ q_0$$

- Focused proof system helps to prove this result.

Complexity

Some decidable fragments:

Theorems

- If we bound k in the multiplexing rule in the calculus SLLM with a fixed constant k_0, such a fragment becomes decidable.

- In the case where we bound k in the multiplexing rule in the calculus SLLM with a fixed constant k_0, and, in addition, we bound the depth of nesting of $!A$, we get NP-completeness.

This result provides NP-procedures for parsing complex and compound sentences in many practically important cases.

Conclusions and Future Work

We proposed SLLM, a proof system for type-logical grammars that:

- admits cut-elimination;
- admits substitution;
- satisfies Lambek's non-emptiness restriction.

We proposed a sound and complete focused proof system for SLLM

We investigated the complexity for SLLM provability.

For future work:

- Classical logic versions of our logical framework;
- Extending systems with additives;
- Implementation of lazy forms of proof search.

Related Work

- R. J. Simmons and F. Pfenning. Weak Focusing for Ordered Linear Logic. Technical Report CMU-CS-10-147 2011.
- J. Polakow. Linear logic programming with an ordered context. In PPDP 2000.
- F. Pfenning and R. J. Simmons. Substructural operational semantics as ordered logic programming. In LICS, pages 101–110, 2009.
- M. Kanovich, S. Kuznetsov, V. Nigam,and A. Scedrov. Subexponentials in non-commutative linear logic. In Mathematical Structures in Computer Science 2018. Dale Miller's Festschrift.
- M. Kanovich, S. Kuznetsov, V. Nigam,and A. Scedrov. A Logical Framework with Commutative and Non-commutative Subexponentials. In IJCAR 2018.
- G. Morrill and O. Valentin. Multiplicative-additive focusing for parsing as deduction. In First International Workshop on Focusing, 2015.
- C. Olarte, E. Pimentel, and V. Nigam. Subexponential concurrent constraint programming. Theor. Comput. Sci., 606:98–120, 2015.
- V. Nigam, E. Pimentel, and G. Reis. An extended framework for specifying and reasoning about proof systems. J. Log. Comput., 26(2):539–576, 2016.
- V. Nigam. A framework for linear authorization logics. TCS, 536:21–41, 2014.