Sequent systems for Lambek calculi and their extensions: A survey of new results

Wojciech Buszkowski

Faculty of Mathematics and Computer Science
Adam Mickiewicz University, Poznań

Lambek's paper "The mathematics of sentence structure" (1958) introduced Syntactic Calculus as an extension of the type reduction procedure for categorial grammars, due to Ajdukiewicz (1935) and Bar-Hillel (1953). This calculus was later called Lambek Calculus and extensively studied in computational linguistics as a basic type logic for these grammars. In 1961 Lambek proposed a non-associative version of this calculus. Here we denote these calculi by L and NL, respectively. Lambek presented both calculi in the form of Gentzen-style sequent systems and proved the cut-elimination theorems for them. A characteristic feature of these systems is the lack of structural rules (exchange, weakening, contraction), which are admitted in sequent systems for classical logic and intuitionistic logic. The connectives are multiplication $\otimes$ and two implications $\backslash, /$; the latter are also called divisions or residuations.

One also considers many extensions of L and NL. In the logical community, in particular in substructural logics, L enriched with 1 (the unit for $\otimes$) and lattice connectives and constants is called Full Lambek Calculus (FL) and regarded as a basic substructural logic. Substructural logics are defined as extensions of FL by new axioms and rules. The name *substructural logics* is justified by the lack of some structural rules in their sequent systems. Propositional Linear Logic (PLL) of Girard (1987) enriches the commutative FL with negation, satisfying the double negation law, and two unary modalities, called exponentials. Its fragment without exponentials is called Multiplicative-Additive Linear Logic (MALL); Lambek connectives, 1 and negation are referred to as multiplicative and lattice connectives and constants as additive. Further extensions admit various modalities, Kleene iteration, the distributive laws for lattice connectives, and others.

In this talk I focus on sequent systems for different logics from this family. More precisely, I consider two properties of them: cut elimination and interpolation. I'm concerned with some consequences of these properties, which are interesting for computational linguistics.

The cut elimination theorems were proved for almost all systems, considered here. This talk, however, discusses generalized versions of these theorems, working for logics extended with non-logical axioms, here called assumptions. As a consequence, these logics are strongly conservative extensions of their language-restricted fragments. Like in foundations of mathematics first-order theories are even more important than the pure first-order logic, I believe that Lambek logics with assumptions can find useful applications in linguistics.

Furthermore, applying cut-free systems for pure logics one constructs syntactic interpretations of stronger logics in weaker logics, admitting Lambek's restriction: the antecedent of any sequent must be nonempty; see (Buszkowski 2014, 2015).

Craig Interpolation Lemma states that for any provable implication $\varphi \to \psi$ there exists an interpolant χ such that χ uses only nonlogical symbols symbols common for φ and ψ and both $\phi \to \chi$ and $\chi \to \psi$ are provable.
For L the interpolation lemma of Roorda (1991), formulated for a sequent system for L, states that for any provable sequent $\Gamma_1, \Delta, \Gamma_2 \Rightarrow A$ there exists an interpolant D such that both $\Gamma_1, D, \Gamma_2 \Rightarrow A$ and $\Delta \Rightarrow D$ are provable, and the number of occurrences of any atom in D is not greater than these numbers for Δ and for the context $\Gamma_1, \_, \Gamma_2 \Rightarrow A$. Roorda's lemma is essential in the proof that Lambek grammars are weakly equivalent to context-free grammars (Pentus 1993).

For multiplicative non-associative logics, if $\Gamma[\Delta] \Rightarrow A$ is provable, then the interpolant D of Δ can be found among subformulas of the formulas in $\Gamma[\Delta] \Rightarrow A$ (for systems with assumptions one must adds subformulas of the formulas in assumptions). For NL with assumptions, the latter lemma was proved in (Buszkowski 2005), and later extended to other logics by several authors. Some interesting consequences are the PTIME complexity of the consequence relation (i.e. provability from assumptions) for these logics and the weak equivalence of categorial grammars based on these logics (also with assumptions) with context-free grammars.

Lambek calculi

The calculus NL in algebraic form. A, B, C denote formulas (types) formed out of atoms $p, q, r, \ldots$ and connectives $\otimes, \backslash, /$. The axioms are

$$\text{(Id) } A \Rightarrow A$$

and the inference rules are

$$\text{(RES-R1) } \frac{A \otimes B \Rightarrow C}{B \Rightarrow A \backslash C} \quad \text{(RES-R2) } \frac{A \otimes B \Rightarrow C}{A \Rightarrow C / B}$$

$$\text{(CUT) } \frac{A \Rightarrow B \,;\; B \Rightarrow C}{A \Rightarrow C}$$

The algebraic form of L is obtained by affixing two new axioms, which express the associative law

$$\text{(As1) } (A \otimes B) \otimes C \Rightarrow A \otimes (B \otimes C) \quad \text{(As2) } A \otimes (B \otimes C) \Rightarrow (A \otimes B) \otimes C$$

$\Rightarrow$ is interpreted as the partial order in the model.

The following rules are derivable in both NL and L

$$\text{(MON-R1)} \; \frac{A \Rightarrow B}{C \otimes A \Rightarrow C \otimes B} \; \text{(MON-R1')} \; \frac{A \Rightarrow B}{A \otimes C \Rightarrow B \otimes C}$$

$$\text{(MON-R2)} \; \frac{A \Rightarrow B}{C \backslash A \Rightarrow C \backslash B} \; \text{(MON-R2')} \; \frac{A \Rightarrow B}{B \backslash C \Rightarrow A \backslash C}$$

$$\text{(MON-R3)} \; \frac{A \Rightarrow B}{A / C \Rightarrow B / C} \; \text{(MON-R3')} \; \frac{A \Rightarrow B}{C / B \Rightarrow C / A}$$

We list some laws provable in NL
(L1) $A \otimes (A \backslash B) \Rightarrow B$, $(B/A) \otimes A \Rightarrow B$ (application)
(L2) $A \Rightarrow (B/A) \backslash B$, $A \Rightarrow B/(A \backslash B)$ (type-raising)
(L3) $A \Rightarrow B \backslash (B \otimes A)$, $A \Rightarrow (A \otimes B)/B$ (expansion)
and in L
(L4) $(A \backslash B) \otimes (B \backslash C) \Rightarrow A \backslash C$, $(A/B) \otimes (B/C) \Rightarrow A/C$ (composition)
(L5) $(A \backslash B)/C \Rightarrow A \backslash (B/C)$, $A \backslash (B/C) \Rightarrow (A \backslash B)/C$ (switching)

Let us briefly discuss the linguistic meaning of this formalism. The basic interpretation refers to syntax. Σ is the lexicon of some language. Sentences and other phrases are represented as strings of words from Σ. Types represent certain sets of strings, corresponding to syntactic categories. The order is inclusion.
Atomic types represent some basic categories, e.g. s - the category of declarative sentences (statements), n - the category of proper nouns, N - the category of common nouns.
Compound types represent functor categories. A string x is of type $A\backslash B$ (resp. B/A), if and only if for any y of type A the concatenation yx (resp. xy) is of type B; one says that x is a left-looking (resp. right-looking) functor from category A to category B. Also z is of type $A \otimes B$ if and only if $z = xy$, for some x of type A and y of type B.
In non-associative frameworks, strings are replaced by bracketed strings (phrase structures). One writes (x, y) for xy.

So $n\backslash s$ represents the category of verb phrases (e.g 'works'), $(n\backslash s)/n$ the category of transitive verb phrases (e.g. 'likes'), $s/(n\backslash s)$ the category of (full) noun phrases (e.g. 'every boy'), $(s/(n\backslash s))/N$ the category of determiners (e.g. 'every'), and so on.

$n \Rightarrow s/(n\backslash s)$ (an instance of (L2)) says that every proper noun is a full noun phrase.
Here we ignore agreement, flection etc. To regard them a finer typing is necessary. For instance, s_1 for statements in present tense, s_2 for statements in past tense, n_i, $i = 1, 2, 3$, for subjects in the $i-$th person. Then, 'likes' is of type $n_3\backslash s_1$, 'like' is of types $n_1\backslash s_1$ and $n_2\backslash s_1$, and 'liked' is of types $n_i\backslash s_2$, for $i = 1, 2, 3$.

One may use n as a general type of subjects, assuming $n_i \Rightarrow n$, for $i = 1, 2, 3$, and similarly s as a general type of statements, assuming $s_i \Rightarrow s$, for $i = 1, 2$. (This naturally leads to Lambek calculi with assumptions.) Then, 'liked' can be assigned type $n\backslash s_2$, and $n\backslash s_2 \Rightarrow n\backslash s$ is provable, by (MON-R2).

Categorial grammars

A categorial grammar provides the whole information on the particular language by types assigned to words (the principle of lexicalism). Other types are derived by the logic underlying the grammar (common for all languages). One may assign many types to one word.

We assign 'John': n,'smiles': $n\backslash s$. This yields 'John smiles': s, since $n \otimes (n\backslash s) \Rightarrow s$ is provable both in NL and L.

We assign 'often': $(n\backslash s)/(n\backslash s)$. This yields 'John often smiles': s, since $n \otimes ((n\backslash s)/(n\backslash s)) \otimes (n\backslash s) \Rightarrow s$ is provable in L.

In NL, $n \otimes (((n\backslash s)/(n\backslash s)) \otimes (n\backslash s)) \Rightarrow s$ is provable, which yields (John, (often, smiles)): s.

We assign 'boy': N, 'some': $(s/(n\backslash s))/N$. This yields 'some boy smiles': s in grammars based on L, and ((some, boy), smiles): s in grammars based on NL.

Sequent systems for L and NL

For L, a sequent is of the form $\Gamma \Rightarrow A$, where Γ is a nonempty sequence of types. Commas in Γ are interpreted as $\otimes$. The axioms are (Id) $A \Rightarrow A$. The rules are as follows.

$$(\otimes\Rightarrow)\ \frac{\Gamma, A, B, \Delta \Rightarrow C}{\Gamma, A\otimes B, \Delta \Rightarrow C} \quad (\Rightarrow\otimes)\ \frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A\otimes B}$$

$$(\backslash\Rightarrow)\ \frac{\Gamma, B, \Delta \Rightarrow C \quad \Phi \Rightarrow A}{\Gamma, \Phi, A\backslash B, \Delta \Rightarrow C} \quad (\Rightarrow\backslash)\ \frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A\backslash B}$$

$$(/\Rightarrow)\ \frac{\Gamma, A, \Delta \Rightarrow C \quad \Phi \Rightarrow B}{\Gamma, A/B, \Phi, \Delta \Rightarrow C} \quad (\Rightarrow /)\ \frac{\Gamma, B \Rightarrow A}{\Gamma \Rightarrow A/B}$$

$$(\mathrm{CUT})\ \frac{\Gamma, A, \Delta \Rightarrow B \quad \Phi \Rightarrow A}{\Gamma, \Phi, \Delta \Rightarrow B}$$

In $(\Rightarrow\backslash)$ and $(\Rightarrow /)$ the sequence Γ must be nonempty. Removing this constraint yields a stronger system L^{ϵ}, admitting empty antecedents of sequents.

In NL the antecedent of a sequent is a bunch. A bunch is a single formula or (Γ, Δ), where Γ and Δ are simpler bunches. A context is a bunch $\Gamma[\_]$ with one occurrence of the special atom $\_$. $\Gamma[\Delta]$ denotes the substitution of Δ for $\_$ in this context.
The axioms of NL are (Id). The inference rules are as follows.

$$(\otimes \Rightarrow)\ \frac{\Gamma[(A, B)] \Rightarrow C}{\Gamma[A \otimes B] \Rightarrow C} \quad (\Rightarrow \otimes)\ \frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{(\Gamma, \Delta) \Rightarrow A \otimes B}$$

$$(\backslash \Rightarrow)\ \frac{\Gamma[B] \Rightarrow C \quad \Delta \Rightarrow A}{\Gamma[(\Delta, A \backslash B)] \Rightarrow C} \quad (\Rightarrow \backslash)\ \frac{(A, \Gamma) \Rightarrow B}{\Gamma \Rightarrow A \backslash B}$$

$$(/ \Rightarrow)\ \frac{\Gamma[A] \Rightarrow C \quad \Delta \Rightarrow B}{\Gamma[(A/B, \Delta)] \Rightarrow C} \quad (\Rightarrow /)\ \frac{(\Gamma, B) \Rightarrow A}{\Gamma \Rightarrow A/B}$$

$$(\mathrm{CUT})\ \frac{\Gamma[A] \Rightarrow B \quad \Delta \Rightarrow A}{\Gamma[\Delta] \Rightarrow B}$$

NL^{ϵ} admits the empty bunch ϵ as the antecedent of a sequent. One assumes $(\epsilon, \Gamma) = \Gamma = (\Gamma, \epsilon)$ and writes $\Rightarrow A$ for $\epsilon \Rightarrow A$.

Cut elimination

The cut-elimination theorem Every sequent provable in the logic has a proof with no application of (CUT).

This theorem was proved for the sequent systems for L and NL by Lambek (1958, 1961) by proof-theoretic methods. The same proofs work for L^{ϵ} and NL^{ϵ}.

Main consequences:

(1) Every provable sequent has a proof such that every formula appearing in the proof is a subformula of some formula occurring in this sequent (the subformula property).

(2) The logic is a conservative extension of its all language restricted fragments. For instance, L is a conservative extension of L restricted to $\backslash, /$.

(3) If the size of the conclusion of every rule is not less than the size of each premise of this rule, than the logic is decidable. This holds for the four logics considered above.

We also consider the consequence relation $S \vdash \Gamma \Rightarrow A$, where S is a set of sequents., called assumptions. In opposition to axioms, the set of assumptions need not be closed under substitution.

The extended subformula property If a sequent is provable in the logic from a set of assumptions S, then it has a proof from S such that every formula appearing in the proof is a subformula of some formula occurring in this sequent or in assumptions.
For NL and L, this was proved by W.B. (2005) by a model-theoretic argument: if a sequent has no proof satisfying this property, than there exists a model such that all assumptions are true but the sequent is not true.
My PhD student Zhe Lin (2010) found a proof-theoretic argument. For NL and L, one can only consider assumptions is of the form $A \Rightarrow B$. For NL, each assumption $A \Rightarrow B$ is replaced by the following rule.

$$(S{-}\mathrm{CUT})\ \frac{\Gamma[B] \Rightarrow C \quad \Delta \Rightarrow A}{\Gamma[\Delta] \Rightarrow C}$$

$$(S\text{−CUT}) \text{ for L } \frac{\Gamma, B, \Delta \Rightarrow C \quad \Phi \Rightarrow A}{\Gamma, \Phi, \Delta \Rightarrow C}$$

In NL^{ϵ} and L^{ϵ} one can only consider assumptions of the form $\Rightarrow A$. Each assumption $\Rightarrow A$ is replaced by the following $(S\text{−CUT})$-rule.

$$\text{for } \mathsf{NL}^{\epsilon} \ \frac{\Gamma[A] \Rightarrow B}{\Gamma[\epsilon] \Rightarrow B} \ \text{for } \mathsf{L}^{\epsilon} \ \frac{\Gamma, A, \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow B}$$

One must reduce $\Gamma[\epsilon]$ to a bumch without explicit occurrences of ϵ.
One proves the cut-elimination theorem in the form: Every sequent provable from S has a proof not applying (CUT), but it can use some rules $(S\text{−CUT})$.
This yields the extended subformula property. Accordingly, each logic preserves the consequence relation of its language restricted fragments.
This does not yield decidability. The consequence relations for L and L^{ϵ} (with finite S) are undecidable.

Extensions

The same can be shown for different extensions of Lambek calculi. Full Lambek Calculus (FL) enriches L with additive connectives $\wedge, \vee$ with the corresponding rules.

$$(\wedge \Rightarrow) \frac{\Gamma, A, \Delta \Rightarrow C}{\Gamma, A \wedge B, \Delta \Rightarrow C} \; \frac{\Gamma, B, \Delta \Rightarrow C}{\Gamma, A \wedge B, \Delta \Rightarrow C} \; (\Rightarrow \wedge) \frac{\Gamma \Rightarrow A \; \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \wedge B}$$

$$(\vee \Rightarrow) \frac{\Gamma, A, \Delta \Rightarrow C \; \Gamma, B, \Delta \Rightarrow C}{\Gamma, A \vee B, \Delta \Rightarrow C} \; (\Rightarrow \vee) \frac{\Gamma \Rightarrow A}{\Gamma \Rightarrow A \vee B} \; \frac{\Gamma \Rightarrow B}{\Gamma \Rightarrow A \vee B}$$

For FNL the left-introduction rules are as follows.

$$(\wedge \Rightarrow) \frac{\Gamma[A] \Rightarrow C}{\Gamma[A \wedge B] \Rightarrow C} \; \frac{\Gamma[B] \Rightarrow C}{\Gamma[A \wedge B] \Rightarrow C} \; (\vee \Rightarrow) \frac{\Gamma[A] \Rightarrow C \; \Gamma[B] \Rightarrow C}{\Gamma[A \vee B] \Rightarrow C}$$

In the same way one obtains FL^{ϵ} and FNL^{ϵ} from L^{ϵ} and NL^{ϵ}, respectively. In the literature on substructural logics, Full Lambek Calculus is defined as FL^{ϵ} with constant 1, admitting:

$$(\Rightarrow 1) \;\; \Rightarrow 1 \; (1 \Rightarrow) \frac{\Gamma, \Delta \Rightarrow A}{\Gamma, 1, \Delta \Rightarrow A}$$

Optionally one adds the constant 0 with no new axioms or rules.
One defines substructural negations $A^{\sim} = A\backslash 0$, $A^{-} = 0/A$.

One can also add some structural rules: exchange (e), weakening (w), contraction (c).

$$(e)\,\frac{\Gamma, A, B, \Delta \Rightarrow C}{\Gamma, B, A, \Delta \Rightarrow C}\;(w)\,\frac{\Gamma, \Delta \Rightarrow A}{\Gamma, \Phi, \Delta \Rightarrow A}$$

$$(c)\,\frac{\Gamma, \Phi, \Phi, \Delta \Rightarrow A}{\Gamma, \Phi, \Delta \Rightarrow A}$$

We omit non-associative versions of these rules.
In logics with (e) the formulas $A\backslash B$ and B/A are equivalent. Both are replaced by $A \rightarrow B$. In logic with (w) and 1 one proves $A \Rightarrow 1$, hence 1 is interpreted as the greatest element in models.
Each logic is a strongly conservative extension of its language restricted fragments.

A syntactic interpretation

Systems admitting empty antecedents are preferred by logicians: they yield provable formulas.
Systems not admitting empty antecedents are preferred by some linguists. For instance, in L^{ϵ} the types N/N and $((N/N)/(N/N)$ are equivalent, while linguists want to distinguish them: N/N is assigned to adjectives and $(N/N)/(N/N)$ to adverbs.
In W.B. (2014, 2015) the former systems (admitting empty antecedents) are interpreted in the latter (not admitting empty antecedents). One defines two interpretation maps P and N: P is applied to positive occurrences and N to negative occurrences of subformulas in the sequent. This is more elegant for logics with $\wedge, \vee$.
$P(p) = N(p) = p$ for atoms p
$P(A \wedge B) = P(A) \wedge P(B)$, $N(A \wedge B) = N(A) \wedge N(B)$, similarly for $\vee$
$N(A \otimes B) = N(A) \otimes N(B)$

$P(A \otimes B) = P(A) \otimes P(B)$ if neither $\Rightarrow A$, nor $\Rightarrow B$ is provable
$P(A \otimes B) = (P(A) \otimes P(B)) \vee P(B)$ if $\Rightarrow A$ but not $\Rightarrow B$ is provable
$P(A \otimes B) = (P(A) \otimes P(B)) \vee P(A)$ if $\Rightarrow B$ but not $\Rightarrow A$ is provable
$P(A \otimes B) = (P(A) \otimes P(B)) \vee P(A) \vee P(B)$ if both $\Rightarrow A$ and $\Rightarrow B$ are provable
$P(A \backslash B) = N(A) \backslash P(B)$, $P(B/A) = P(B)/N(A)$
$N(A \backslash B) = P(A) \backslash N(B)$, $N(B/A) = N(B)/P(A)$ if $\Rightarrow A$ is not provable
$N(A \backslash B) = (P(A) \backslash N(B)) \wedge N(B)$,
$N(B/A) = (N(B)/P(A)) \wedge N(B)$ if $\Rightarrow A$ is provable

Theorem For every sequent $\Gamma \Rightarrow A$ with $\Gamma \neq \epsilon$, this sequent is provable in FL^{ϵ} if and only if $N(\Gamma) \Rightarrow P(A)$ is provable in FL. This also holds for logics with structural rules and non-associative logics. One essentially uses cut-elimination for the pure logics.

Interpolation

Interpolation in our sense reverses cut elimination. Every provable sequent (also from assumptions) can be presented as the conclusion of (CUT), where the cut-formula is a subformula of some formula appearing in this sequent (or in some assumption).

The interpolation theorem Let $\Gamma[\Delta] \Rightarrow A$ be provable from S. There exists a formula D, being a subformula of some formula in this sequent or in sequents from S, such that both $\Gamma[D] \Rightarrow A$ and $\Delta \Rightarrow D$ are provable from S.
D is called an interpolant of Δ in $\Gamma[\Delta] \Rightarrow A$.
W.B. (2005) proves this theorem for NL. The proof works for NL^{ϵ} (also with 1). One proceeds by induction on proofs, satisfying the extended subformula property.
Let T be a set of formulas, closed under subformulas.
A $T-$sequent consists of formulas from T only.
A sequent of the form $A \Rightarrow B$ or $(A, B) \Rightarrow C$ is said to be basic.

We fix a finite set of assumptions S. Every assumption is of the form $A \Rightarrow B$ with $A, B \in T$.
We construct a set $C(T)$ which consists of all basic $T-$sequents provable from S in NL. The construction can be performed in polynomial time. We omit details.
Therefore, the consequence relation for NL is PTIME.

Furthermore, every provable $T-$sequent $\Gamma \Rightarrow A$ can be proved from $C(T)$ using (CUT) only. This is essentially a derivation in a context-free grammar.
Therefore, the categorial grammars, based on NL with finitely many assumptions, generate context-free languages.

Bulińska (2009) proved the same results for NL^{ϵ} with 1 and Zhe Lin (2010) for some modal extensions of NL.

For logics with $\wedge, \vee$, the interpolant can be a $\wedge, \vee-$combination of formulas from a finite set. In general, this yields an infinite set of possible interpolants, It is known that the consequence relations for FNL and FNL$^{\epsilon}$ are undecidable (Chvalovsky 2015).
They are decidable for FNL and FNL$^{\epsilon}$ enriched with the distributive laws. It suffices to add one new axiom.

$$\text{(Dist) } A \wedge (B \vee C) \Rightarrow (A \wedge B) \vee (A \wedge C)$$

The interpolation theorem can be proved with a finite set of possible interpolants. Using the distributive laws, every $\wedge, \vee-$combination of formulas from T can be transformed into its CNF. Omitting repetitions, one obtains a finite set of these forms. This yields the decidability and the context-freeness of languages generated by categorial grammars based on these logics (W.B. 2011). The consequence relation for these logics with $\top, \bot$ is EXPTIME-complete (Shkatov, van Alten 2019).

Non-associative linear logics

In the last years some related results were obtained for the multiplicative fragments of non-associative linear logics. Recently non-associative linear logics have been studied by authors from the community of computational logic; see e.g. (Blaisdell et al. 2022). The first logic of this kind was introduced by de Groote and Lamarche (2002) under the name Classical Non-associative Lambek Calculus. The logic can be presented as the extension of NL by negation $^{\sim}$, satisfying the double negation and the contraposition laws: $A^{\sim\sim} \Leftrightarrow A$ and $A^{\sim}/B \Leftrightarrow A\backslash B^{\sim}$. This system does not admit cut elimination. In the cited paper a cut-free one-sided sequent system is given and its PTIME-complexity is proved. The system is shown to be a conservative extension of NL.

W.B. (2016) studies a dual one-directed sequent system, proves the cut-elimination theorem and the interpolation theorem. The logic is shown to be a strongly conservative extension of NL. Categorial grammars based on this logic (also with assumptions) generate context-free languages.

One also considers the extension of NL with two negations $^{\sim}, ^{-}$, satisfying: $A^{\sim -} \Leftrightarrow A$, $A^{-\sim} \Leftrightarrow A$ and $A^{\sim}/B \Leftrightarrow A \backslash B^{-}$, like in Non-commutative MALL of Abrusci (1991).
W.B (2017) presents a one-sided sequent system for this logic, called Involutive Non-associative Lambek Calculus, proves the cut-elimination theorem and the PTIME-complexity of the pure logic. The proof uses an interpolation theorem. The methods are proof-theoretic. The complexity (even the decidability) of the consequence relation remains an open problem.
W.B. (2019) studies an extension of this logic by a unary modality and obtains similar results by model-theoretic tools. Categorial grammars based on this logic are shown to generate context-free languages.
Some results have been adapted for logics with $\wedge, \vee$ and 1 by Płaczek (2021).

Selected references

E. Blaisdell, M. Kanovich, S. Kuznetsov, E. Pimentel, A. Scedrov: Non-commutative, non-associative multi-modal linear logic. In: LNCS 13385 (2022).

W. Buszkowski: Lambek calculus with nonlogical axioms. In: C. Casadio et al. (eds.), Language and Grammar, CSLI Pubnlications (2005)

W. B.: Interpolation and FEP for logics of residuated algebras. Logic Journal of The IGPL 19 (2011), 437–454.

W.B.: Some syntactic interpretations in different systems of Full Lambek Calculus. In: S. Ju et al. (eds.), Modality, Semantics and Interpretations, Studia Logica Library, Springer (2015).

W.B.: On Classical Nonassociative Lambek Calculus. In: M. Amblard, Ph. de Groote et al. (eds.), Logical Aspects of Computational Linguistics, LNCS 10054 (2016).

W.B.: On Involutive Nonassociative Lambek Calculus. Journal of Logic, Language and Information 28 (2019), 157-181.

K. Chvalovsky: Undecidability of consequence relation in full nonassociative Lambek calculus. Journal of Symbolic Logic 80 (2015), 524–540.
J.-Y. Girard: Linear logic. Theoretical Computer Science 50 (1987), 1–102.
P. de Groote, F. Lamarche: Classical non-associative Lambek calculus. Studia Logica 71 (2002), 355–388.
J. Lambek: The mathematics of sentence structure. The American Mathematical Monthly 65 (1958), 154–170.
J. Lambek: On the calculus of syntactic types. In: R. Jakobson (ed.), Structure of Language and Its Mathematical Aspects, American Mathematical Society (1961).
Z. Lin: Modal nonassociative Lambek calculus with assumptions: Complexity and Context-Freeness. In: LNCS 6031 (2010).
P. Płaczek: Extensions of Lambek calculi: Sequent systems, conservativeness and computational complexity. PhD Thesis, Adam Mickiewicz University (2021).

D. Shkatov, C.J. Van Alten: Complexity of the universal theory of bounded residuated lattice-ordered groupoids. Algebra Universalis 80.3 (2019).

Thank you