Chapter 2
Some Syntactic Interpretations in Different Systems of Full Lambek Calculus

Wojciech Buszkowski

Abstract Buszkowski (Logical Aspects of Computational Linguistics, 2014 [8]) defines an interpretation of FL without 1 in its version without empty antecedents of sequents (employed in type grammars) and applies this interpretation to prove some general results on the complexity of substructural logics and the generative capacity of type grammars. Here this interpretation is extended for nonassociative logics (also with structural rules), logics with 1, logics with distributive laws for $\wedge$, $\vee$, logics with unary modalities, and multiplicative fragments.

2.1 Introduction and Preliminaries

Full Lambek Calculus is a basic substructural logic [12]. In the present paper Full Lambek Calculus is denoted by FL_1, its 1-free fragment by FL^*, and the subsystem of FL^* not allowing empty antecedents of sequents by FL. This notation differs from a standard one [12], where FL stands for our FL_1. The pure logicians, however, usually ignore logics like FL in our sense, and we need a notation discriminating these different systems.

Type grammars (or: categorial grammars) are formal grammars based on type-theoretic syntax and semantics. The language is described by an assignment of types to lexical items (words), and compound expressions are processed by means of a type logic. Type logics are certain basic substructural logics, usually presented as sequent systems: formulae of these logics are interpreted as types. Type grammars often employ logics not allowing empty antecedents of sequents, e.g., L, NL. NL$\Diamond$.

The present paper studies some relations between the versions allowing empty antecedents (more popular among logicians) and those not allowing them (more popular among linguists). We reduce the provability in the former systems to the provability in the latter, using two translations N and P of formulae in the language of FL (or its extension) into formulae of the same language. N (resp. P) acts on

W. Buszkowski (✉)
Faculty of Mathematics and Computer Science,
Adam Mickiewicz University in Poznań, Poznań, Poland
e-mail: buszko@amu.edu.pl

S. Ju et al. (eds.), *Modality, Semantics and Interpretations*,
Logic in Asia: Studia Logica Library, DOI 10.1007/978-3-662-47197-5_2

negative (resp. positive) occurrences of subformulae in sequents. More details are given at the end of this section.

FL_1 admits no structural rules except *associativity*. Type grammars also employ its nonassociative versions. The basic logics are Full Nonassociative Lambek Calculus (FNL_1), also named Groupoid Logic (GL) in [12, 13], and its subsystem FNL.

FNL_1 can be presented as a sequent system in language $(\cdot, \backslash, /, 1, \wedge, \vee)$. We refer to $\cdot$ as *product*, $\backslash$ as *right implication*, $/$ as *left implication*, $\wedge$ as *and*, $\vee$ as *or*. We reserve metavariables p, q, r, s (possibly with subscripts, primes, etc.) for variables and $\alpha, \beta, \gamma, \delta$ for formulae. *Formula structures* are the elements of the free unital groupoid generated by the set of formulae. All formulae are (atomic) formula structures, λ is the unit (the empty structure), and the compound structures are of the form (Γ, Δ), where Γ and Δ are (nonempty) formula structures. We assume $(\Gamma, \lambda) = (\lambda, \Gamma) = \Gamma$, for any structure Γ. Sequents are of the form $\Gamma \Rightarrow \alpha$. One writes $\Rightarrow \alpha$ for $\lambda \Rightarrow \alpha$. *Contexts* are extended formula structures, containing one occurrence of a special atom x (a place for substitution). If Γ is a context, then $\Gamma[\Delta]$ denotes the substitution of Δ for x in Γ; see [13] for a more precise exposition.

FNL_1 is based on the following axioms and rules:

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

$$\text{(L}\cdot\text{)}\ \frac{\Gamma[(\alpha, \beta)] \Rightarrow \gamma}{\Gamma[\alpha \cdot \beta] \Rightarrow \gamma} \quad \text{(R}\cdot\text{)}\ \frac{\Gamma \Rightarrow \alpha;\ \ \Delta \Rightarrow \beta}{(\Gamma, \Delta) \Rightarrow \alpha \cdot \beta}$$

$$\text{(L}\backslash\text{)}\ \frac{\Gamma[\beta] \Rightarrow \gamma;\ \ \Delta \Rightarrow \alpha}{\Gamma[(\Delta, \alpha\backslash\beta)] \Rightarrow \gamma} \quad \text{(R}\backslash\text{)}\ \frac{(\alpha, \Gamma) \Rightarrow \beta}{\Gamma \Rightarrow \alpha\backslash\beta}$$

$$\text{(L/)}\ \frac{\Gamma[\beta] \Rightarrow \gamma;\ \ \Delta \Rightarrow \alpha}{\Gamma[(\beta/\alpha, \Delta)] \Rightarrow \gamma} \quad \text{(R/)}\ \frac{(\Gamma, \alpha) \Rightarrow \beta}{\Gamma \Rightarrow \beta/\alpha}$$

$$\text{(L}\wedge\text{)}\ \frac{\Gamma[\alpha_i] \Rightarrow \gamma}{\Gamma[\alpha_1 \wedge \alpha_2] \Rightarrow \gamma} \quad \text{(R}\wedge\text{)}\ \frac{\Gamma \Rightarrow \alpha;\ \ \Gamma \Rightarrow \beta}{\Gamma \Rightarrow \alpha \wedge \beta}$$

$$\text{(L}\vee\text{)}\ \frac{\Gamma[\alpha] \Rightarrow \gamma;\ \ \Gamma[\beta] \Rightarrow \gamma}{\Gamma[\alpha \vee \beta] \Rightarrow \gamma} \quad \text{(R}\vee\text{)}\ \frac{\Gamma \Rightarrow \alpha_i}{\Gamma \Rightarrow \alpha_1 \vee \alpha_2}$$

$$\text{(L}1_l\text{)}\ \frac{\Gamma[\Delta] \Rightarrow \alpha}{\Gamma[(1, \Delta)] \Rightarrow \alpha} \quad \text{(L}1_r\text{)}\ \frac{\Gamma[\Delta] \Rightarrow \alpha}{\Gamma[(\Delta, 1)] \Rightarrow \alpha} \quad \text{(R1)}\ \Rightarrow 1$$

$$\text{(CUT)}\ \frac{\Gamma[\alpha] \Rightarrow \beta;\ \ \Delta \Rightarrow \alpha}{\Gamma[\Delta] \Rightarrow \beta}$$

FNL^* denotes the subsystem of FNL_1, restricted to the formulae without 1; so $(L1_l)$, $(L1_r)$ and (R1) are omitted. FNL admits neither 1, nor λ; so the formula structures form the free groupoid generated by the set of 1-free formulae. One says that α is provable, if $\Rightarrow \alpha$ is provable.

We also consider structural rules: *associativity* (a), *exchange* (e), *integrality* (i) (also called: left weakening), and *contraction* (c).

$$\text{(a)}\ \frac{\Gamma[((\Delta_1, \Delta_2), \Delta_3)] \Rightarrow \gamma}{\Gamma[(\Delta_1, (\Delta_2, \Delta_3))] \Rightarrow \gamma} \quad \text{(e)}\ \frac{\Gamma[(\Delta_1, \Delta_2)] \Rightarrow \gamma}{\Gamma[(\Delta_2, \Delta_1)] \Rightarrow \gamma}$$

$$\text{(i)}\ \frac{\Gamma[\Delta_i] \Rightarrow \gamma}{\Gamma[(\Delta_1, \Delta_2)] \Rightarrow \gamma} \quad \text{(c)}\ \frac{\Gamma[(\Delta, \Delta)] \Rightarrow \gamma}{\Gamma[\Delta] \Rightarrow \gamma}$$

Since λ is the unit for the operation (,), rules (a), (e), (c) can be restricted to nonempty structures Δ_i, Δ, and similarly for (i) except the special case: from $\Rightarrow \alpha$ infer $\Delta \Rightarrow \alpha$.

Following [12], by FNL_e we denote FNL with (e), by FNL_{1ei} we denote FNL_1 with (e) and (i), and so on. Logics with (a) are *associative* substructural logics. FNL_{1a} is further denoted by FL_1 and called Full Lambek Calculus with 1 (this is precisely Full Lambek Calculus in the sense of [12]). We also define FL^* as FNL^*_a, FL as FNL_a, FL_e as FNL_{ae}, and so on. If S is a set of structural rules, then FNL^*_S (resp. FNL_S) denotes FNL^* (resp. FNL) enriched with all rules from S, and similarly for FL^*_S and FL_S.

In associative substructural logics the antecedents of sequents can be represented as finite sequences of formulae (then, (a) is implicit). The empty sequence is denoted by ϵ, and one writes $\Rightarrow \alpha$ for $\epsilon \Rightarrow \alpha$. We only recall rules (L $\cdot$) and (R $\cdot$) in this form (see [8] for the full list).

$$\text{(L}\cdot\text{)}\ \frac{\Gamma, \alpha, \beta, \Gamma' \Rightarrow \gamma}{\Gamma, \alpha \cdot \beta, \Gamma' \Rightarrow \gamma}\ , \quad \text{(R}\cdot\text{)}\ \frac{\Gamma \Rightarrow \alpha;\ \ \Delta \Rightarrow \beta}{\Gamma, \Delta \Rightarrow \alpha \cdot \beta}$$

One can also consider the constants $\bot$, $\top$ with the axioms:

$$\text{(a.}\bot\text{)}\ \Gamma[\bot] \Rightarrow \alpha \quad \text{(a.}\top\text{)}\ \Gamma \Rightarrow \top\,.$$

In this paper 0 (used to define negations; see [12]) plays no essential role, like in type grammars. Nonetheless one might add it to all logics studied here, with no axioms, nor rules for 0. We do not consider sequents of the form $\Gamma \Rightarrow$; the rule of *right weakening* can be simulated by the definition $0 = \bot$.

In logics with (e) $\alpha\backslash\beta \Leftrightarrow \beta/\alpha$ is provable; this means: the sequents in both directions are provable. Therefore, one writes $\alpha \to \beta$ for $\alpha\backslash\beta$ and β/α (precisely, instead of two implications $\backslash$, $/$ one employs one implication $\to$).

All logics mentioned above satisfy *the cut-elimination theorem*: every provable sequent can be proved without (CUT). For the $(\cdot, \backslash, /)$-fragments of FL and FNL this theorem was proved by Lambek [20, 21] and for a version of FL_1 by Ono and Komori [32].

The $(\cdot, \backslash, /)$-fragment of FL was proposed by Lambek [20] as the calculus of syntactic types for type grammars. Lambek's name Syntactic Calculus was later replaced by Lambek Calculus. This logic is denoted by L and its nonassociative version by NL. The variants L^*, L_1, L_e, NL^*, NL_1, NL_e, etc., are defined as above.

In the terminology of linear logics, $\cdot$, $\backslash$, $/$, 1, 0 are the *multiplicative* connectives and constants, and $\wedge$, $\vee$, $\bot$, $\top$ are the *additive* ones. In this paper, we assume the higher priority of multiplicatives over additives. For instance, $p \vee q/r$ stands for $p \vee (q/r)$.

The cut-elimination theorem entails *the subformula property*: every provable sequent $\Gamma \Rightarrow \alpha$ has a proof in which all sequents consist of subformulae of the formulae appearing in $\Gamma \Rightarrow \alpha$. As a consequence, the logics allowing cut elimination are conservative extensions of their language-restricted fragments, e.g., FL_1 conservatively extends FL^* and L^*. On the other hand, NL^* is not a conservative extension of NL; for instance, $p/(q/q) \Rightarrow p$ is provable in NL^* but not in NL. Similarly, L^* is not a conservative extension of L, FL^* of FL, and so on. Another consequence is *the decidability* of all logics FNL_{1S}, FNL^*_S, and FNL_S such that S does not contain (c). By various methods it has been shown that all logics FL_{1S} but FL_{1c} are decidable; the undecidability of FL_{1c} and FL_c was announced by K. Chvalovsky and R. Horčik at *Logic, Algebra and Truth Degrees* 2014 (unpublished). Similarly, all logics FL_S are decidable except FL_c. For FNL_{1c}, FNL_{1ec}, FNL_c, FNL_{ec} the problem of (un)decidability is open; the remaining nonassociative logics FNL_{1S} and FNL_S are decidable.

A *type grammar*, based on the logic $\mathcal{L}$ (an $\mathcal{L}$-grammar) can be defined as a triple $G = (\Sigma_G, I_G, \delta_G)$ such that Σ_G is a nonempty finite set, I_G is a mapping which assigns a finite set of formulae of $\mathcal{L}$ to every element of Σ_G, and δ_G is a formula of $\mathcal{L}$. Σ_G is called *the lexicon* (or: alphabet) of G, I_G *the initial type assignment* of G, and δ_G *the designated type* of G. Most often δ_G is a fixed variable, and one denotes it by s_G (or: s).

Variables play the role of *atomic types*, corresponding to basic syntactic categories. Therefore, they are really understood as constants, but we treat them as variables in type logics (in particular, all provable sequents are closed under substitutions).

In mathematical linguistics, expressions of the language are often represented as *phrase structures*, i.e. (skeletal) trees whose leaves are labeled by words. To avoid drawing pictures we represent phrase structures as bracketed strings: (XY) represents the tree whose root has two daughters, being the roots of X and Y, respectively. The recursive definition is as follows: (i) all elements of Σ are phrase structures (on Σ), (ii) if X and Y are phrase structures, then (XY) is a phrase structure. Σ^P denotes the set of all phrase structures on Σ, and Σ^{P*} additionally contains the empty structure λ. This resembles the standard notation of the formal language theory: Σ^* (resp. Σ^+) for the set of all (resp. nonempty) finite strings on Σ. By *a language* (resp. *a phrase language*) on Σ one means an arbitrary set $L \subseteq \Sigma^*$ (resp. $L \subseteq \Sigma^{P*}$); it is said to be ϵ-free (resp. λ-free), if $\epsilon \notin L$ (resp. $\lambda \notin L$).

Clearly, the formula structures are precisely the phrase structures on the set of formulae. For better readability, the constituents of a formula structure are separated by a comma, but commas are omitted in phrase structures of language expressions. So we write (John works), but (n,n\s).

Let G be an $\mathcal{L}$-grammar. I_G is extended for phrase structures on Σ_G as follows: $I_G(\lambda) = \lambda$, $I_G((XY)) = \{(\Gamma, \Delta) : \Gamma \in I_G(X), \Delta \in I_G(Y)\}$. One defines: $X :_G \alpha$, for $X \in \Sigma_G^{P*}$, if there exists $\Gamma \in I_G(X)$ such that $\Gamma \Rightarrow \alpha$ is provable in $\mathcal{L}$. $L^P(G, \alpha)$ consists of all $X \in \Sigma_G^{P*}$ such that $X :_G \alpha$. $L^P(G, \delta_G)$ is called *the phrase language* of G and denoted by $L^P(G)$. *The language* of G consists of all yields of the trees from $L^P(G)$, i.e., the strings obtained by dropping all parentheses in the structures from $L^P(G)$.

For natural languages, the elements of Σ_G are interpreted as words (lexical items) and the elements of $L(G)$ as grammatically correct declarative sentences (statements). For formal languages, Σ_G consists of symbols of the language and $L(G)$ of all strings generated by G.

The size limits of this paper do not allow any serious discussion of type grammars applied to natural and formal languages. The reader is referred to the recent textbook [29] for linguistic applications. General overviews can also be found in [4, 27, 30].

We give a few examples. Let us fix the atomic types: s (sentence), n (proper noun), cn (common noun). $\alpha\backslash\beta$ (resp. β/α) is interpreted as the type of *functors* (or: functional expressions) which with any expression of type α on the left (resp. right) form a compound expression of type β. Thus, n\s is a type of (intransitive) verb phrase, e.g., 'works,' 'walks in the garden,' (n\s)/n of transitive verb phrase, e.g., 'likes,' 'desperately loves,' s/(n\s) of noun phrase as subject, e.g., 'he,' 'some student,' 'every teacher,' (s/n)\s of noun phrase as object, e.g., 'him,' 'her,' 'some student,' (s/(n\s))/cn of determiner, e.g. 'some,' 'every,' 'one,' cn/cn of adjective (as noun modifier), (cn/cn)/(cn/cn) of adverb (as adjective modifier), and so on.

This typing is by no way the only possible option. Different authors propose different types. For instance, np (noun phrase) is often counted to atomic types, verb phrases are typed np\s, and transitive verb phrases (np\s)/np. The particular choice is motivated by various reasons, e.g., semantics, analogies with logical formalisms, economy of typing, and others. Above we completely ignore tense, person, number, case. Lambek [22] employs 33 atomic types for a fragment of English, among them: s (statement), s_1 (statement in present tense), s_2 (statement in past tense), π (subject), π_k, for $k = 1, 2, 3$ (subject in kth person). Subtypes are linked with the main type by nonlogical assumptions, e.g., $s_k \Rightarrow$s, $\pi_k \Rightarrow \pi$.

Type logics like L and its extensions are more flexible than the classical type reduction procedure of [1] (going back to ideas of K. Ajdukiewicz (1935) and Y. Bar-Hillel (1953)), which is based on *the reduction laws*: $\alpha, \alpha\backslash\beta \Rightarrow \beta$ and $\beta/\alpha, \alpha \Rightarrow \beta$. The latter can be formalized as the subsystem of L (or NL), restricted to $(\backslash, /)$-types and axiomatized by (Id), (L\), (L/) ((CUT) is admissible); one denotes this poor logic by AB. Since n,(n\s)/n,n $\Rightarrow$ s is provable in AB, then any AB-grammar with the types listed above accepts 'John likes Mary' as a sentence. Since s/(n\s),(n\s)/n,(s/n)\s$\Rightarrow$ s is provable in L, not in AB, then 'he likes her' is accepted by an L-grammar, but not an AB-grammar with these types. One can repair this failure by assigning new types, e.g. (s/n)/((n\s)/n) to 'he.' In general, AB-grammars require many initial types assigned to words, while L-grammars can reduce their number and explain logical relations between types. Here s/(n\s) can be expanded to the new type by the law $\alpha/\beta \Rightarrow (\alpha/\gamma)/(\beta/\gamma)$, provable in L. Proofs in L and its extensions determine semantic transformations, definable in typed lambda calculus. Due to size limits, we cannot discuss this topic here; again we refer to [4, 27, 29]. Logics of semantic types admit (a), (e) and, possibly, other structural rules.

In a sense, L is even too strong for linguistic purposes. By associativity, if G is an L-grammar, then $L^P(G)$ contains all possible phrase structures whose yields belong to $L(G)$. For instance, not only ((every student) (hates (some teacher))) and (((every student) hates) (some teacher)) are accepted, which well reflects the ∀∃ and ∃∀ readings of this sentence, but also (every ((student (hates some)) teacher)), which is linguistically weird (though admits a semantic reading). This and other reasons motivate some linguists to prefer NL as a basic type logic. NL does not accept all possible phrase structures, but it is certainly too weak for a satisfactory description of natural language. For instance, with the types listed above, NL accepts 'he likes Mary,' since (s/(n\s),((n\s)/n,n)) $\Rightarrow$ s is provable, but not 'John likes her.'

The power of nonassociative logics can be strengthened in different ways. One possibility is to employ theories; for instance, we add to NL some assumptions, provable in L (assumptions are not closed under substitutions). Buszkowski [6] shows that the provability from assumptions in NL is decidable in polynomial time, so this approach leads to tractable parsing procedures. Another approach, elaborated by Morrill [30], Moortgat [26, 27] and others (see [29]), extends NL by new operators, e.g., unary modalities $\Diamond, \Box^{\downarrow}$, connected by the unary residuation law: $\Diamond a \leq b$ iff $a \leq \Box^{\downarrow} b$, also several pairs of modalities, new binary products, and other extras. Although associativity is not assumed in general, it is allowed for some modal formulae, and so like with other structural rules. This resembles the role of exponentials in linear logics [14]. Logics with $\Diamond, \Box^{\downarrow}$ will be considered in Sect. 2.4.

Although types with $\wedge, \vee$ were not frequently employed in type grammars, some authors considered them for different reasons. Lambek [21] used $\wedge$ to replace the multivalued type assignment $I_G(a) = \{\alpha_1, \dots, \alpha_n\}$ by the one-valued type assignment $I_G(a) = \alpha_1 \wedge \cdots \wedge \alpha_n$. Kanazawa [17] considered a feature decomposition of types; for instance, singular (resp. plural) noun phrases are typed np $\wedge$ sing (resp. np $\wedge$ pl). Lambek's nonlogical assumptions can be replaced by definitions, e.g., $\pi = \pi_1 \vee \pi_2 \vee \pi_3$. Besides such concrete applications, type logics with $\wedge, \vee$ are interesting for theoretical reasons also from the viewpoint of type grammars. In particular, our translations N, P essentially employ additives.

The methods of this paper are proof-theoretic, not algebraic. Therefore we omit the definitions of algebras, corresponding to logics under consideration. We only note that NL is the strongly complete logic of residuated groupoids, NL_1 of unital residuated groupoids, L of residuated semigroups, L_1 of residuated monoids, FNL of lattice-ordered residuated groupoids, FL of residuated lattices, and so on; see [3, 12, 13]. In algebras, a formula structure Γ is interpreted as the formula $f(\Gamma)$, recursively defined as follows: $f(\lambda) = 1$, $f(\alpha) = \alpha$, $f((\Gamma, \Delta)) = f(\Gamma) \cdot f(\Delta)$. $\Gamma \Rightarrow \gamma$ is true in the algebra A for the valuation μ, if $\mu(f(\Gamma)) \leq \mu(\alpha)$.

In linguistics, the standard models of L are *language models*, i.e., the powerset algebras $\mathcal{P}(\Sigma^+)$. The operations are defined as follows: $L_1 \cdot L_2 = \{uv : u \in L_1, v \in$

$L_2\}$ and \, / are the residual operations. For L*, one replaces Σ^+ by Σ^*, for NL by Σ^P, and for NL* by Σ^{P*}.

In language models the connectives $\wedge$ and $\vee$ are naturally interpreted as intersection and union, respectively, of (phrase) languages. This yields distributive lattices. The distributive laws for $\wedge$, $\vee$ are not provable in logics considered above. One can add them as new axioms. It is sufficient to add:

$$\text{(D) } \alpha \wedge (\beta \vee \gamma) \Rightarrow (\alpha \wedge \beta) \vee (\alpha \wedge \gamma).$$

FNL with (D) is denoted by DFNL, and similarly for other systems, e.g. DFNL*, DFL, etc. The cut-elimination theorem does not hold for these axiomatizations, but can be proved for other, equivalent systems; see Sect. 2.4. DFL_1 is the complete logic of distributive residuated lattices.

Let us note that a linguistic interpretation can be found for logics like FL*, FL, FNL*, FNL, in terms of *syntactic concept lattices* of Clark [11]. Grammatical categories are defined by sets of contexts; the family of all categories is a complete residuated lattice (non-distributive, in general). This construction resembles models obtained by nuclear completion; see [2, 12, 13].

Now we briefly comment on some reasons for the usage of λ-free logics, like NL, FNL, L, FL, in type grammars. In these logics no single formula is provable, hence they cannot be formalized as Hilbert-style systems, nor easily related to other nonclassical logics. Therefore the pure logicians usually ignore them. In type grammars, however, they were extensively studied, starting from Lambek [20]. One reason is semantical: types are to be assigned to meaningful expressions only, and the 'empty expression' has no meaning. In linguistics, expressions are analyzed as syntactic structures, e.g., phrase structures, normally restricted to nonempty structures. Although the theory of production grammars and automata regards languages containing ϵ, type grammars are more natural and elegant, when they are restricted to ϵ-free languages. Since $\Rightarrow s$ is not provable in any consistent substructural logic, $L(G)$ is ϵ-free for any grammar G whose designated type is a variable. Moortgat and Oehrle [28] provide other arguments. Above we assigned type cn/cn to adjectives (noun modifiers) and (cn/cn)/(cn/cn) to adverbs (adjective modifiers). Since $\alpha/\alpha \Leftrightarrow (\alpha/\alpha)/(\alpha/\alpha)$ is provable in L*, then adjectives and adverbs are indistinguishable on the basis of L*, which is unacceptable for linguistics. This problem can be overcome by modifying the initial typing, e.g. assigning a new atomic type to adjectives, but this complicates the grammar and is less natural from the semantical viewpoint. Also the ϵ-free formal languages admit a simpler typing than their companions with ϵ; see [8].

This paper is organized as follows. In Sect. 2.2 we translate every formula of FNL into two formulae $N(\alpha)$ and $P(\alpha)$ of the same language (although N, P depend on the particular logic, their definition is uniform for all logics considered here). Theorem 2.1 states that $\Gamma \Rightarrow \gamma$ is provable in FNL^*_S if and only if $N(\Gamma) \Rightarrow P(\gamma)$ is provable in FNL_S, for any sequent $\Gamma \Rightarrow \gamma$ with $\Gamma \neq \lambda$ and any set of structural rules S. Theorem 2.2 provides an interpretation of FL_{1S} in FL^*_S, extending the interpretation of L_1 in L* from [19].

The functions N, P for FNL^*_S are computable if and only if FNL^*_S is decidable. We show that the decidability of FNL_S implies the decidability of FNL^*_S, and the decidability of FL_S implies the decidability of FL_{1S}.

In Sect. 2.3 the translation maps N, P are adapted for the multiplicative fragments; they obtain the form of multivalued maps which send each formula to a finite set of formulae. The method also works with arbitrary structural rules except (c). Section 2.4 extends the results of Sect. 2.2 for logics with the distributive laws for $\wedge$, $\vee$ and logics with unary modalities. Section 2.5 discusses some consequences of these results for the generative capacity of type grammars and the complexity of substructural logics (the theorem of [16] on the PSPACE-hardness of consistent substructural logics with the disjunction property is extended for logics not allowing empty antecedents and with restricted associativity).

The present paper continues [8], which focuses on FL^* and FL. Some extensions, elaborated here, are announced in [8] without proof. No proof from [8] is rewritten here except some brief outlines in Sect. 2.5.

Since this paper is strongly motivated by linguistic applications of substructural logics, we refer to relatively many works in the tradition of type grammars. Not to overload the list of references we skip several references to relevant logical works. Much more can be found in [12, 13, 18].

2.2 An Interpretation of FNL^*_S in FNL_S

The *positive* and *negative* occurrences of subformulae in formulae are recursively defined as follows: (i) α is positive in α, (ii) γ is positive (resp. negative) in $\alpha \circ \beta$, where $\circ \in \{\cdot, \wedge, \vee\}$, if γ is positive (resp. negative) in α or β, (iii) γ is positive (resp. negative) in $\alpha\backslash\beta$, if either γ is positive (resp. negative) in β, or γ is negative (resp. positive) in α, (iv) the same for $\gamma = \beta/\alpha$. Further, γ is positive (resp. negative) in (Γ, Δ), if γ is positive (resp. negative) in Γ or Δ, and γ is positive (resp. negative) in $\Gamma \Rightarrow \alpha$, if γ is either negative (resp. positive) in Γ, or positive (resp. negative) in α.

We fix a set of structural rules S. To every formula γ in language $(\cdot, \backslash, /, \wedge, \vee)$ we assign two formulae $N(\gamma)$ (the negative translation of γ) and $P(\gamma)$ (the positive translation of γ) of the same language; see Table 2.1. By $\vdash^*$ and $\vdash$ we denote the provability in FNL^*_S and FNL_S, respectively. Since the side conditions depend on S, then, actually, the maps N and P also depend on S. It would be more precise to write P_S, N_S instead of P, N. We omit the subscript S, if it is clear from the context or inessential. For example, for FNL^* we obtain $N(p/(q/q)) = p/(q/q) \wedge p$, $P(p\cdot(q/q)) = p\cdot(q/q) \vee p$. P (resp. N) acts on positive (resp. negative) occurrences of formulae in sequents.

N and P can also be defined for formulae containing $0, \bot, \top$. We set $N(\gamma) = P(\gamma) = \gamma$, for $\gamma \in \{0, \bot, \top\}$. We extend N for nonempty formula structures by the recursive clause: $N(\Gamma, \Delta) = (N(\Gamma), N(\Delta))$. The same definition works for contexts; we set $N(x) = x$. Clearly $N(\Gamma[\Delta]) = N(\Gamma)[N(\Delta)]$, for $\Delta \neq \lambda$.

Table 2.1 Translations N and P

γ	$N(\gamma)$	$P(\gamma)$	Condition
p	p	p	p is a variable
$\alpha \circ \beta$	$N(\alpha) \circ N(\beta)$	$P(\alpha) \circ P(\beta)$	$\circ \in \{\wedge, \vee\}$
$\alpha \cdot \beta$	$N(\alpha) \cdot N(\beta)$	$P(\alpha) \cdot P(\beta)$	$\nvdash^* \alpha$ and $\nvdash^* \beta$
$\alpha \cdot \beta$	As above	$P(\alpha) \cdot P(\beta) \vee P(\beta)$	$\vdash^* \alpha$ and $\nvdash^* \beta$
$\alpha \cdot \beta$	As above	$P(\alpha) \cdot P(\beta) \vee P(\alpha)$	$\nvdash^* \alpha$ and $\vdash^* \beta$
$\alpha \cdot \beta$	As above	$P(\alpha) \cdot P(\beta) \vee P(\alpha) \vee P(\beta)$	$\vdash^* \alpha$ and $\vdash^* \beta$
$\alpha\backslash\beta$	$P(\alpha)\backslash N(\beta)$	$N(\alpha)\backslash P(\beta)$	$\nvdash^* \alpha$
$\alpha\backslash\beta$	$P(\alpha)\backslash N(\beta) \wedge N(\beta)$	As above	$\vdash^* \alpha$
β/α	$N(\beta)/P(\alpha)$	$P(\beta)/N(\alpha)$	$\nvdash^* \alpha$
β/α	$N(\beta)/P(\alpha) \wedge N(\beta)$	As above	$\vdash^* \alpha$

Lemmas 2.1 and 2.2 below have been proved in [8] for FL and FL*; the same proofs work for FNL and FNL*. Here we outline different proofs of slightly stronger results.

Lemma 2.1 *For any set S and any formula γ, $N_S(\gamma) \Rightarrow \gamma$ and $\gamma \Rightarrow P_S(\gamma)$ are provable in FNL.*

Proof We prove both claims by simultaneous induction on γ. If γ is an atom, then they hold, by (Id) and the definition of N_S, P_S. For the inductive steps, one uses *the monotonicity rules*, derivable in FNL (in (MON) $\circ \in \{\cdot, \wedge, \vee\}$).

$$\text{(MON)}\ \frac{\alpha \Rightarrow \alpha' ;\ \beta \Rightarrow \beta'}{\alpha \circ \beta \Rightarrow \alpha' \circ \beta'}$$

$$\text{(MON}\backslash\text{)}\frac{\alpha \Rightarrow \alpha' ;\ \beta \Rightarrow \beta'}{\beta'\backslash\alpha \Rightarrow \beta\backslash\alpha'}\ \text{(MON/)}\frac{\alpha \Rightarrow \alpha' ;\ \beta \Rightarrow \beta'}{\alpha/\beta' \Rightarrow \alpha'/\beta}$$

Furthermore, the following sequents are provable in FNL, for any formulae α, β and any set S.

$$P_S(\alpha) \cdot P_S(\beta) \Rightarrow P_S(\alpha \cdot \beta) \tag{2.1}$$

$$N_S(\alpha\backslash\beta) \Rightarrow P_S(\alpha)\backslash N_S(\beta),\ N_S(\beta/\alpha) \Rightarrow N_S(\beta)/P_S(\alpha). \tag{2.2}$$

Let $\gamma = \alpha\backslash\beta$. By the induction hypothesis, $N_S(\beta) \Rightarrow \beta$ and $\alpha \Rightarrow P_S(\alpha)$ are provable in FNL. Then, $P_S(\alpha)\backslash N_S(\beta) \Rightarrow \alpha\backslash\beta$ is provable, by (MON\), and consequently, $N_S(\gamma) \Rightarrow \gamma$ is provable, by (2.2) and (CUT). By the induction hypothesis, $N_S(\alpha) \Rightarrow \alpha$ and $\beta \Rightarrow P_S(\beta)$ are provable in FNL. Then, $\alpha\backslash\beta \Rightarrow N_S(\alpha)\backslash P_S(\beta)$ is provable, by (MON\), and we have $P_S(\alpha\backslash\beta) = N_S(\alpha)\backslash P_S(\beta)$.

The remaining cases are left to the reader.

In the proof of Lemma 2.2, $N = N_S$ and $P = P_S$.

Lemma 2.2 *Let S be fixed. For any formula γ, $N_S(\gamma) \Leftrightarrow \gamma$ and $P_S(\gamma) \Leftrightarrow \gamma$ are provable in FNL$^*_S$.*

Proof Although $N(\gamma) \Rightarrow \gamma$ and $\gamma \Rightarrow P(\gamma)$ are provable, by Lemma 2.1, it is convenient to prove both claims with $\Leftrightarrow$ by simultaneous induction on γ. For atoms, they are obvious. For the inductive steps, we use the fact that $\vdash^* \alpha \Leftrightarrow \beta$ is a congruence on the formula algebra. Furthermore, the following are provable in FNL$^*_S$, for any formulae α, β.

$$P(\alpha) \cdot P(\beta) \Leftrightarrow P(\alpha \cdot \beta) \tag{2.3}$$

$$P(\alpha)\backslash N(\beta) \Leftrightarrow N(\alpha\backslash\beta),\ \ N(\beta)/P(\alpha) \Leftrightarrow N(\beta/\alpha) \tag{2.4}$$

We show (2.3). $P(\alpha) \cdot P(\beta) \Rightarrow P(\alpha \cdot \beta)$ is provable, by (2.1), since FNL is a subsystem of FNL$^*_S$. We prove $\Leftarrow$. We consider four cases. (1°) $\nvdash^* \alpha$ and $\nvdash^* \beta$. The claim is obvious. (2°) $\vdash^* \alpha$ and $\nvdash^* \beta$. Then, $P(\alpha \cdot \beta) = P(\alpha) \cdot P(\beta) \vee P(\beta)$. By Lemma 2.1 and (CUT), $\vdash^* P(\alpha)$, hence $\vdash^* P(\beta) \Rightarrow P(\alpha) \cdot P(\beta)$, by (R $\cdot$), and $\vdash^* P(\alpha \cdot \beta) \Rightarrow P(\alpha) \cdot P(\beta)$, by (L$\vee$). For (3°) $\nvdash^* \alpha$, $\vdash^* \beta$, and (4°) $\vdash^* \alpha$, $\vdash^* \beta$, the arguments are similar. (2.4) is left to the reader.

Let $\gamma = \alpha \cdot \beta$. By the induction hypothesis, $N(\alpha) \Leftrightarrow \alpha$ and $N(\beta) \Leftrightarrow \beta$ are provable in FNL$^*_S$. Then, $N(\alpha) \cdot N(\beta) \Leftrightarrow \alpha \cdot \beta$ is provable. We have $N(\alpha \cdot \beta) = N(\alpha) \cdot N(\beta)$, which yields $N(\gamma) \Leftrightarrow \gamma$. By the induction hypothesis $P(\alpha) \Leftrightarrow \alpha$ and $P(\beta) \Leftrightarrow \beta$ are provable, hence $P(\alpha) \cdot P(\beta) \Leftrightarrow \alpha \cdot \beta$ is provable. Consequently, $P(\gamma) \Leftrightarrow \gamma$ is provable, by (2.3) and (CUT).

The remaining cases are left to the reader.

We are ready to prove the main result of this section.

Theorem 2.1 Let S be fixed. (I) For any sequent $\Gamma \Rightarrow \gamma$, where $\Gamma \neq \lambda$, $\Gamma \Rightarrow \gamma$ is provable in FNL$^*_S$ if and only if $N_S(\Gamma) \Rightarrow P_S(\gamma)$ is provable in FNL$_S$. (II) Let (i) belong to S. For any formula γ, provable in FNL$^*_S$, and any $\Gamma \neq \lambda$, the sequent $\Gamma \Rightarrow P_S(\gamma)$ is provable in FNL$_S$.

Proof We prove the 'if' part of (I). Assume $\vdash N(\Gamma) \Rightarrow P(\gamma)$. Then, $\vdash^* N(\Gamma) \Rightarrow P(\gamma)$, hence $\vdash^* \Gamma \Rightarrow \gamma$, by Lemma 2.2 and (CUT).

The 'only if' part of (I), for S not containing (i), is proved by induction on cut-free proofs in FNL$^*_S$. For (Id), $\vdash N(\alpha) \Rightarrow P(\alpha)$ holds, by (Id), Lemma 2.1 and (CUT). If $\Gamma \Rightarrow \gamma$ is an axiom (a.$\bot$) or (a.$\top$), then $N(\Gamma) \Rightarrow P(\gamma)$ is also an axiom from this group.

The rules for $\wedge$, $\vee$ and (L $\cdot$), (R$\backslash$), (R$/$) are treated easily. Let us consider (L$\vee$). By the induction hypothesis, $\vdash N(\Gamma[\alpha]) \Rightarrow P(\gamma)$ and $\vdash N(\Gamma[\beta]) \Rightarrow P(\gamma)$. Then, $\vdash N(\Gamma)[N(\alpha)] \Rightarrow P(\gamma)$ and $\vdash N(\Gamma)[N(\beta)] \Rightarrow P(\gamma)$, and we apply (L$\vee$) in FNL$_S$.

Structural rules (a), (e), (c) cause no problem: the induction hypothesis applied to the premise directly yields our claim for the conclusion.

We consider (R $\cdot$). There are three subcases. (1°) $\Gamma \neq \lambda$ and $\Delta \neq \lambda$. By the induction hypothesis, $\vdash N(\Gamma) \Rightarrow P(\alpha)$ and $\vdash N(\Delta) \Rightarrow P(\beta)$. Then, $\vdash N((\Gamma, \Delta)) \Rightarrow P(\alpha) \cdot P(\beta)$, by (R $\cdot$). Using (R$\vee$), if necessary, we obtain $\vdash N((\Gamma, \Delta)) \Rightarrow P(\alpha \cdot \beta)$. (2°) $\Gamma = \lambda$ and $\Delta \neq \lambda$. Then, $\vdash^* \alpha$. Also $\vdash N(\Delta) \Rightarrow P(\beta)$, by the induction hypothesis. Consequently, $\vdash N(\Delta) \Rightarrow P(\alpha \cdot \beta)$, by (R$\vee$), possibly applied twice. (3°) $\Gamma \neq \lambda$ and $\Delta = \lambda$. One argues as for (2°).

We consider (L$\backslash$). There are two subcases. (1°) $\Delta \neq \lambda$. By the induction hypothesis, $\vdash N(\Gamma[\beta]) \Rightarrow P(\gamma)$ and $\vdash N(\Delta) \Rightarrow P(\alpha)$. By (L$\backslash$), we obtain $\vdash N(\Gamma)[(N(\Delta), P(\alpha)\backslash N(\beta))] \Rightarrow P(\gamma)$. Using (L$\wedge$), if necessary, we get $\vdash N(\Gamma)[(N(\Delta), N(\alpha\backslash\beta))] \Rightarrow P(\gamma)$, hence $\vdash N(\Gamma[(\Delta, \alpha\backslash\beta)]) \Rightarrow P(\gamma)$. (2°) $\Delta = \lambda$. Then, $\vdash^* \alpha$ and $N(\alpha\backslash\beta) = P(\alpha)\backslash N(\beta) \wedge N(\beta)$. As for (1°), we get $\vdash N(\Gamma[\beta]) \Rightarrow P(\gamma)$, hence $\vdash N(\Gamma[\alpha\backslash\beta]) \Rightarrow P(\gamma)$, by (L$\wedge$). The argument for (L/) is similar.

Let S contain (i). The ‘only if’ part of (I) and (II) are proved by simultaneous induction on cut-free proofs in FNL_S^* (precisely: on the number of sequents appearing in the cut-free proof). For (I), the argument copies the above, but we need one new case: rule (i).

(1°) $\Gamma[(\Delta_1, \Delta_2)] \Rightarrow \gamma$ results from $\Gamma[\Delta_2] \Rightarrow \gamma$, by (i), with $\Delta_1 \neq \lambda$, $\Delta_2 \neq \lambda$. By the induction hypothesis, $\vdash N(\Gamma[\Delta_2]) \Rightarrow P(\gamma)$, hence $\vdash N(\Gamma[(\Delta_1, \Delta_2)]) \Rightarrow P(\gamma)$, by (i) in FNL_S. (2°) As above with the premise $\Gamma[\Delta_1] \Rightarrow \gamma$. The argument is similar. (3°) $\Gamma \Rightarrow \gamma$ results from $\Rightarrow \gamma$, by (i). By the induction hypothesis for (II), $\vdash N(\Gamma) \Rightarrow P(\gamma)$.

We prove (II). Let γ be provable in FNL_S^*. Then, $\Rightarrow \gamma$ must be either an instance of (a.$\top$), or the conclusion of one of the rules: (R $\cdot$), (R$\backslash$), (R/), (R$\wedge$), (R$\vee$). We fix $\Gamma \neq \lambda$. For (a.$\top$), $\Gamma \Rightarrow P(\gamma)$ is also an instance of (a.$\top$). Let us consider the rules.

(R $\cdot$). Then $\gamma = \alpha \cdot \beta$, and the premises are $\Rightarrow \alpha$, $\Rightarrow \beta$. By the induction hypothesis, $\vdash \Gamma \Rightarrow P(\alpha)$, hence $\vdash \Gamma \Rightarrow P(\alpha \cdot \beta)$, by (R$\vee$) (applied twice).

(R$\backslash$). Then $\gamma = \alpha\backslash\beta$, and the premise is $\alpha \Rightarrow \beta$. By the induction hypothesis for (I), $\vdash N(\alpha) \Rightarrow P(\beta)$. Using (i) in FNL_S, we obtain $\vdash (N(\alpha), \Gamma) \Rightarrow P(\beta)$, hence $\vdash \Gamma \Rightarrow P(\gamma)$, by (R$\backslash$). For (R/), the argument is similar.

(R$\wedge$). Then, $\gamma = \alpha \wedge \beta$, and the premises are $\Rightarrow \alpha$, $\Rightarrow \beta$. By the induction hypothesis, $\vdash \Gamma \Rightarrow P(\alpha)$ and $\vdash \Gamma \Rightarrow P(\beta)$, hence $\vdash \Gamma \Rightarrow P(\gamma)$, by (R$\wedge$).

(R$\vee$). Then $\gamma = \alpha_1 \vee \alpha_2$, and the premise is $\Rightarrow \alpha_i$. By the induction hypothesis, $\vdash \Gamma \Rightarrow P(\alpha_i)$, hence $\vdash \Gamma \Rightarrow P(\gamma)$, by (R$\vee$).

Notice that the proof of Theorem 2.1 would not work, if (CUT) were not eliminated.

To better understand the relation between FNL_S^* and FNL_S we need an additional lemma. We define $\equiv$ as the smallest congruence in the formula algebra such that:

$$(\alpha \circ \beta) \circ \gamma \equiv \alpha \circ (\beta \circ \gamma), \;\; \alpha \circ \beta \equiv \beta \circ \alpha\,, \;\; \alpha \circ \alpha \equiv \alpha\,, \tag{2.5}$$

for all formulae α, β, γ and $\circ \in \{\wedge, \vee\}$. Clearly $\alpha \equiv \beta$ entails: $\alpha \Leftrightarrow \beta$ is provable in FNL.

Lemma 2.3 *Let S be fixed. For any formula γ, $N_S(N_S(\gamma)) \equiv N_S(\gamma)$ and $P_S(P_S(\gamma)) \equiv P_S(\gamma)$.*

Proof Again we proceed by induction on γ, using the fact: $\vdash^* \alpha$ iff $\vdash^* P(\alpha)$, which holds, by Lemma 2.2. Here we only consider the case $\gamma = \alpha \cdot \beta$. There are four subcases.

(1°) $\nvdash^* \alpha$, $\nvdash^* \beta$. We compute:

$$P(P(\gamma)) = P(P(\alpha) \cdot P(\beta)) = P(P(\alpha)) \cdot P(P(\beta)) \equiv P(\alpha) \cdot P(\beta) = P(\gamma).$$

(2°) $\vdash^* \alpha$, $\nvdash^* \beta$. We compute:

$$P(P(\gamma)) = P(P(\alpha) \cdot P(\beta) \vee P(\beta)) = P(P(\alpha) \cdot P(\beta)) \vee P(P(\beta)) =$$

$$P(P(\alpha)) \cdot P(P(\beta)) \vee P(P(\beta)) \vee P(P(\beta)) \equiv P(\alpha) \cdot P(\beta) \vee P(\beta) \equiv P(\gamma).$$

The remaining subcases are treated in a similar way. If $\vdash^* \alpha$, $\vdash^* \beta$, then all three patterns of (2.5) are needed.

A sequent of the form $N_S(\Gamma) \Rightarrow P_S(\gamma)$, where $\Gamma \neq \lambda$, is said to be *stable* in FNL_S. From Lemma 2.3 it follows that in FNL_S^* every sequent whose antecedent is nonempty is deductively equivalent to a stable sequent; $\Gamma \Rightarrow \gamma$ and $N(\Gamma) \Rightarrow P(\gamma)$ are derivable from each other (using (CUT)).

Corollary 2.1 *For any stable sequent in FNL_S, the sequent is provable in FNL_S if and only if it is provable in FNL_S^*.*

Proof Fix a sequent $N(\Gamma) \Rightarrow P(\gamma)$ with $\Gamma \neq \lambda$. The 'only if' part is obvious. For the 'if' part, assume that $N(\Gamma) \Rightarrow P(\gamma)$ is provable in FNL_S^*. By Theorem 1, $N(N(\Gamma)) \Rightarrow P(P(\gamma))$ is provable in FNL_S. Consequently, $N(\Gamma) \Rightarrow P(\gamma)$ is provable in FNL_S, by Lemma 2.3.

Notice that P, N are not extensional (hence not monotone) in FNL_S: $\vdash \alpha \Rightarrow \beta$ need not imply $\vdash P(\alpha) \Rightarrow P(\beta)$, nor $\vdash N(\alpha) \Rightarrow N(\beta)$, and similarly for $\Leftrightarrow$ instead of $\Rightarrow$. Let $\alpha = ((p/(q/q)) \cdot (q/q))/(q/q)$, $\beta = p/(q/q)$. Then, $\alpha \Leftrightarrow \beta$ in FNL, $P(\alpha) = ((p/(q/q)) \cdot (q/q) \vee p/(q/q))/(q/q)$, $P(\beta) = \beta$, and $\nvdash P(\alpha) \Rightarrow P(\beta)$.

Theorem 2.1 does not allow to interpret in FNL_S the provability of $\Rightarrow \gamma$ in FNL_S^*. An indirect reduction, however, is possible with the aid of the following properties of provability in FNL_S^*.

(Pr.1) $\nvdash^* p$, $\nvdash^* 0$, $\nvdash^* \bot$, $\vdash^* \top$,
(Pr.2) $\vdash^* \alpha \circ \beta$ iff $\vdash^* \alpha$ and $\vdash^* \beta$, for $\circ \in \{\cdot, \wedge\}$,
(Pr.3) $\vdash^* \alpha\backslash\beta$ iff $\vdash N(\alpha) \Rightarrow P(\beta)$, $\vdash^* \beta/\alpha$ iff $\vdash N(\alpha) \Rightarrow P(\beta)$,
(Pr.4) $\vdash^* \alpha \vee \beta$ iff $\vdash^* \alpha$ or $\vdash^* \beta$.

In (Pr.3) N, P are to be computed for the direct subformulae of $\alpha\backslash\beta$ and β/α, and this computation requires checking the provability of some proper subformulae of α or β. One easily obtains an algorithm which computes $N(\alpha)$, $P(\alpha)$ and checks $\vdash^* \alpha$, for any formula α; this algorithm is based on the algorithm for checking the provability in FNL_S. Accordingly, the decidability of FNL_S implies the decidability of FNL^*_S.

Theorem 2.1 remains true for some language-restricted fragments, e.g. the languages $(\cdot, \backslash, \wedge, \vee)$, $(\backslash, /, \wedge)$ and $(\backslash, /, \wedge, \vee)$ (also with $0, \bot, \top$), since these languages are closed under N, P.

The fragment $\mathrm{FL}^*_e[\rightarrow, \wedge, 0]$ is interesting, since it has the same expressive power as Multiplicative-Additive Linear Logic (MALL), denoted by InFL_e in [12]. On the one hand, FL^*_e is a conservative fragment of MALL. On the other hand, MALL can be faithfully interpreted in $\mathrm{FL}^*_e[\rightarrow, \wedge, 0]$; see [5]; also MALL with $\bot, \top$ can be interpreted in $\mathrm{FL}^*_e[\rightarrow, \wedge, 0, \top]$. By Theorem 2.1 and (Pr.1)–(Pr.4), one can reduce MALL to $\mathrm{FL}_e[\rightarrow, \wedge, 0]$. This also works for Cyclic MALL (CyInFL) and $\mathrm{FL}[\backslash, /, \wedge, 0]$ enriched with *the cyclic rule*:

$$\text{(C)}\ \frac{\Gamma, \Delta \Rightarrow 0}{\Delta, \Gamma \Rightarrow 0}.$$

For associative logics with 1, our interpretation can be composed with the one of Kuznetsov [19] who interprets L_1 in L^*; his interpretation remains correct for FL_{1S} and FL^*_S. Let us briefly describe this method for FL_1 and FL^* (structural rules cause no difficulty). Here the antecedents of sequents are finite sequences of formulae.

First, axioms (Id) of FL_1 are restricted to $p \Rightarrow p$, for any variable p. Then, all sequents $\alpha \Rightarrow \alpha$ are provable (without (CUT)).

Second, rule (L1) (of the form: from $\Gamma, \Gamma' \Rightarrow \alpha$ infer $\Gamma, 1, \Gamma' \Rightarrow \alpha$) is replaced by the new axioms: (a.1) $1^n \Rightarrow 1$, (a.2) $1^m, p, 1^n \Rightarrow p$, for $m, n \geq 0$ (here 1^n denotes the sequence of n copies of 1). Axiom (R1) equals (a.1) for $n = 0$, and $1 \Rightarrow 1$ is (a.1) for $n = 1$. It is obvious that every sequent provable in the new system is also provable in FL_1. The converse follows from the admissibility of (L1) in the new system (show it by induction on proofs). Let $\mathcal{L}$ denote the new system. One can show that $\mathcal{L}$ allows cut elimination, but it is not essential for this argument.

Third, define a substitution σ by: $\sigma(p) = (1 \cdot p) \cdot 1$, for any variable p. Let $\mathcal{L}^-$ denote $\mathcal{L}$ without (a.2). One shows: $\Gamma \Rightarrow \gamma$ is provable in FL_1 if and only if $\sigma(\Gamma \Rightarrow \gamma)$ is provable in $\mathcal{L}^-$. The 'if' part is obvious. The 'only if' part is proved by induction on proofs in $\mathcal{L}$.

Fourth, a pseudo-substitution η is defined by: $\eta(1) = q \backslash q$ (here q is a new variable, not occurring in sequents under consideration). One shows: $\Gamma \Rightarrow \gamma$ is provable in FL_1 if and only if $\eta(\sigma(\Gamma \Rightarrow \gamma))$ is provable in FL^*. The 'if' part is easy: substitute 1 for q in $\eta(\sigma(\Gamma \Rightarrow \gamma))$. The 'only if' part is a consequence of the following: if $\Gamma \Rightarrow \gamma$ is provable in $\mathcal{L}^-$, then $\eta(\Gamma \Rightarrow \gamma)$ is provable in FL^* (use induction on proofs in $\mathcal{L}^-$). Associativity is essential for proving $(q\backslash q)^n \Rightarrow q\backslash q$ in FL^*.

Theorem 2.2 Let S be a set of structural rules (e), (i), (c). For any sequent $\Gamma \Rightarrow \gamma$ with $\Gamma \neq \epsilon$, $\Gamma \Rightarrow \gamma$ is provable in FL_{1S} if and only if $N(\eta(\sigma(\Gamma))) \Rightarrow P(\eta(\sigma(\gamma)))$ is provable in FL_S.

Table 2.2 $P(\alpha \cdot \beta)$ for logics with (i)

γ	$P(\gamma)$	Condition
$\alpha \cdot \beta$	$P(\alpha) \cdot P(\beta)$	$\nvdash^* \alpha$ and $\nvdash^* \beta$
$\alpha \cdot \beta$	$P(\beta)$	$\vdash^* \alpha$
$\alpha \cdot \beta$	$P(\alpha)$	$\nvdash^* \alpha$ and $\vdash^* \beta$

Proof This theorem immediately follows from Theorem 2.1 and the equivalence proved above. Precisely, Theorem 2.1 should be rewritten in the form appropriate for antecedents represented as sequences of formulae.

For logics with (i), Kuznetsov's interpretation can be simplified. Since (L1) is an instance of (i), we omit (L1) in the axiomatization. Then, there is no need for auxiliary logics $\mathcal{L}$, $\mathcal{L}^-$. One directly proves: $\Gamma \Rightarrow \gamma$ is provable in FL_{1S} if and only if $\eta(\Gamma \Rightarrow \gamma)$ is provable in FL^*_S, and drops σ in Theorem 2.2. This also works for nonassociative logics with (i).

For S containing (i), $P(\alpha) \cdot P(\beta) \Rightarrow P(\alpha)$ and $P(\alpha) \cdot P(\beta) \Rightarrow P(\beta)$ are provable in FNL_S; also $P(\alpha) \Leftrightarrow P(\beta)$ is provable in FNL_S, if α and β are provable in FNL^*_S (use Theorem 2.1. (II)). Therefore, the clauses defining $P_S(\alpha \cdot \beta)$ can be simplified as in Table 2.2.

Notice that Kunetsov's interpretation is polynomial, while our is exponential: the size of $N(\alpha)$ and $P(\alpha)$ can be exponential in the size of α. We define $\alpha_0 = p$, $\alpha_{n+1} = (q \backslash q) \backslash \alpha_n$. The size of α_n is linear in n, but $N(\alpha_n)$ contains 2^n occurrences of p.

Since the provability in FL_{1S} is reducible to the provability in FL^*_S, and the latter to the provability in FL_S, then the decidability of FL_S implies the decidability of FL_{1S}. FL_{1c} is undecidable, hence FL^*_c and FL_c are undecidable (see Sect. 2.1).

2.3 Multiplicative Fragments

$N(\gamma)$ and $P(\gamma)$ may contain additives, if even γ is a multiplicative formula. Therefore, the results of Sect. 2.2 cannot directly be applied to multiplicative logics NL^* and NL, L^* and L, etc. We, however, show that N, P can be replaced by multivalued maps which send a formula to a finite set of formulae. Buszkowski [8] announces this solution for L^* and L (without proof). Here we also regard nonassociative logics, possibly with structural rules (a), (e), (i), and provide a proof, employing a multivalued interpretation of FNL_S in NL_S, working for sequents with limited occurrences of $\wedge$, $\vee$.

If U, V are sets of formulae and $\circ$ is a binary connective, then we define: $U \circ V = \{\alpha \circ \beta : \alpha \in U, \beta \in V\}$. Every formula γ in language $(\cdot, \backslash, /, \wedge, \vee)$ is translated into a set $I(\gamma)$, of formulae in language $(\cdot, \backslash, /)$. This also works with 1, 0, $\bot$, $\top$ added to both languages.

(I.1) $I(\alpha) = \{\alpha\}$, for any atomic formula α,
(I.2) $I(\alpha \circ \beta) = I(\alpha) \circ I(\beta)$, for $\circ \in \{\cdot, \backslash, /\}$,
(I.3) $I(\alpha \circ \beta) = I(\alpha) \cup I(\beta)$, for $\circ \in \{\wedge, \vee\}$.

The occurrence of a connective $\circ$ in a formula (resp. sequent) is called *positive*, if $\circ$ is the main connective of a subformula which occurs positively in this formula (resp. sequent), and similarly for negative occurrences.

Lemma 2.4 *Let γ contain no positive (resp. negative) occurrence of $\wedge$ and no negative (resp. positive) occurrence of $\vee$. Then, for any $\delta \in I(\gamma)$, $\delta \Rightarrow \gamma$ (resp. $\gamma \Rightarrow \delta$) is provable in FNL.*

Proof Both claims are proved by simultaneous induction on γ. If γ is atomic, they are obvious.

$\gamma = \alpha \cdot \beta$. So $I(\gamma) = I(\alpha) \cdot I(\beta)$. Let $\delta \in I(\gamma)$. Then, $\delta = \delta_1 \cdot \delta_2$, for some $\delta_1 \in I(\alpha)$, $\delta_2 \in I(\beta)$. By the induction hypothesis, $\delta_1 \Rightarrow \alpha$ and $\delta_2 \Rightarrow \beta$ (resp. $\alpha \Rightarrow \delta_1$, $\beta \Rightarrow \delta_2$) are provable in FNL, hence $\delta \Rightarrow \gamma$ (resp. $\gamma \Rightarrow \delta$) is provable, by (MON).

$\gamma = \alpha\backslash\beta$. So $I(\gamma) = I(\alpha)\backslash I(\beta)$, α contains no negative (resp. positive) occurrence of $\wedge$ and no positive (resp. negative) occurrence of $\vee$, and β contains no positive (resp. negative) occurrence of $\wedge$ and no negative (resp. positive) occurrence of $\vee$. Let $\delta \in I(\gamma)$. So $\delta = \delta_1\backslash\delta_2$, for some $\delta_1 \in I(\alpha)$, $\delta_2 \in I(\beta)$. By the induction hypothesis, $\alpha \Rightarrow \delta_1$ and $\delta_2 \Rightarrow \beta$ (resp. $\delta_1 \Rightarrow \alpha$ and $\beta \Rightarrow \delta_2$) are provable, hence $\delta \Rightarrow \gamma$ (resp. $\gamma \Rightarrow \delta$) is provable, by (MON\). For $\gamma = \beta/\alpha$, the reasoning is similar.

$\gamma = \alpha \wedge \beta$. Only the second claim is applicable. Let $\delta \in I(\gamma)$. Then, $\delta \in I(\alpha)$ or $\delta \in I(\beta)$. Assume $\delta \in I(\alpha)$. Clearly α satisfies the assumptions of the second claim. By the induction hypothesis, $\alpha \Rightarrow \delta$ is provable, hence $\gamma \Rightarrow \delta$ is provable, by (L$\wedge$). For $\delta \in I(\beta)$, the reasoning is similar.

$\gamma = \alpha \vee \beta$. Only the first claim is applicable. Let $\delta \in I(\gamma)$. Then, $\delta \in I(\alpha)$ or $\delta \in I(\beta)$. We only consider the first case. Again, α satisfies the assumptions of the first claim. By the induction hypothesis, $\delta \Rightarrow \alpha$ is provable, hence $\delta \Rightarrow \gamma$ is provable, by (R$\vee$).

We extend I to be defined for formula structures: $I(\lambda) = \{\lambda\}$, $I((\Gamma, \Delta)) = \{(\Gamma', \Delta') : \Gamma' \in I(\Gamma), \Delta' \in I(\Delta)\}$, and for sequents: $I(\Gamma \Rightarrow \gamma) = \{\Gamma' \Rightarrow \gamma' : \Gamma' \in I(\Gamma), \gamma' \in I(\gamma)\}$. In words, $I(\Gamma \Rightarrow \gamma)$ consists of all sequents which are obtained from $\Gamma \Rightarrow \gamma$ by replacing every formula α occurring in Γ (as an atomic structure) by some formula $\beta \in I(\alpha)$ and γ by some formula $\delta \in I(\gamma)$.

Lemma 2.5 *Let S be a set of structural rules (a), (e), (i). Let $\Gamma \Rightarrow \gamma$ be a sequent containing no positive occurrence of $\wedge$ and no negative occurrence of $\vee$. Then, $\Gamma \Rightarrow \gamma$ is provable FNL$_S$ (resp. FNL$^*_S$, FNL1_S) if and only if there exists a sequent $\Gamma' \Rightarrow \gamma' \in I(\Gamma \Rightarrow \gamma)$ such that $\Gamma' \Rightarrow \gamma'$ is provable in NL$_S$ (resp. NL$^*_S$, NL1_S). This remains true for logics with* $0, \bot, \top$.

Proof We only prove the lemma for FNL$_S$ and NL$_S$; for variants the argument is almost the same.

We prove the 'if' part. Let $\Gamma' \Rightarrow \gamma'$ be provable in NL_S and $\Gamma' \Rightarrow \gamma' \in I(\Gamma \Rightarrow \gamma)$. Every formula α occurring in Γ (as an atomic structure) contains no negative occurrence of $\wedge$ and no positive occurrence of $\vee$, and γ contains no positive occurrence of $\wedge$ and no negative occurrence of $\vee$. Each α in Γ is replaced in Γ' by some $\beta \in I(\alpha)$, but $\alpha \Rightarrow \beta$ is provable in FNL, by Lemma 2.4. Also $\gamma' \Rightarrow \gamma$ is provable in FNL, by Lemma 2.4. Consequently, $\Gamma \Rightarrow \gamma$ is provable in FNL_S, by (CUT).

The 'only if' part is proved by induction on cut-free proofs in FNL_S. For (Id) $\alpha \Rightarrow \alpha$, there exists $\beta \in I(\alpha)$, and $\beta \Rightarrow \beta$ is again (Id). For $\Gamma \Rightarrow \gamma$ being (a.$\top$) or (a.$\bot$), every sequent from $I(\Gamma \Rightarrow \gamma)$ is an axiom of the same kind.

Structural rules (a), (e), (i) cause no problem: we apply the induction hypothesis to the premise, then apply the same rule.

The remaining rules to be considered are all rules for multiplicative connectives and (L$\wedge$), (R$\vee$). For the former rules, the arguments are easy. We only consider (L$\backslash$). The premises are: $\Gamma[\beta] \Rightarrow \gamma$ and $\Delta \Rightarrow \alpha$, and the conclusion is: $\Gamma[(\Delta, \alpha\backslash\beta)] \Rightarrow \gamma$. By the induction hypothesis, there exist sequents $\Gamma'[\beta'] \Rightarrow \gamma' \in I(\Gamma[\beta] \Rightarrow \gamma)$ and $\Delta' \Rightarrow \alpha' \in I(\Delta \Rightarrow \alpha)$, provable in NL_S. By (L$\backslash$), $\Gamma'[(\Delta', \alpha'\backslash\beta')] \Rightarrow \gamma'$ is provable in NL_S, and the latter sequent belongs to $I(\Gamma[(\Delta, \alpha\backslash\beta)] \Rightarrow \gamma)$.

(L$\wedge$). The premise is $\Gamma[\alpha_i] \Rightarrow \gamma$, where $i = 1$ or $i = 2$, and the conclusion is $\Gamma[\alpha_1 \wedge \alpha_2] \Rightarrow \gamma$. By the induction hypothesis, there exists a sequent $\Gamma'[\alpha_i'] \Rightarrow \gamma' \in I(\Gamma[\alpha_i] \Rightarrow \gamma)$, provable in NL_S. Since $\alpha_i' \in I(\alpha_i)$, then $\alpha_i' \in I(\alpha_1 \wedge \alpha_2)$. Consequently, the latter sequent belongs to $I(\Gamma[\alpha_1 \wedge \alpha_2] \Rightarrow \gamma)$.

(R$\vee$). The premise is $\Gamma \Rightarrow \alpha_i$, where $i = 1$ or $i = 2$, and the conclusion is $\Gamma \Rightarrow \alpha_1 \vee \alpha_2$. By the induction hypothesis, there exists a sequent $\Gamma' \Rightarrow \alpha_i' \in I(\Gamma \Rightarrow \alpha_i)$, provable in NL_S. As above, this sequent also belongs to $I(\Gamma \Rightarrow \alpha_1 \vee \alpha_2)$.

Lemma 2.5 does not hold for logics with (c). For instance, $p \wedge q \Rightarrow p \cdot q$ is provable in FNL_c, but neither $p \Rightarrow p \cdot q$, nor $q \Rightarrow p \cdot q$ is provable. The limitation of occurrences of $\wedge$, $\vee$ is essential: $(p, r) \Rightarrow p \cdot r$ is provable, $(p \vee q, r) \Rightarrow p \cdot r$ is not provable, and $(p, r) \Rightarrow p \cdot r \in I((p \vee q, r) \Rightarrow p \cdot r)$, but the occurrence of $\vee$ is negative.

Using Lemma 2.4, one easily proves that for any sequent $\Gamma \Rightarrow \gamma$, containing no negative occurrence of $\wedge$ and no positive occurrence of $\vee$, if $\Gamma \Rightarrow \gamma$ is provable in FNL_S, then every sequent $\Gamma' \Rightarrow \gamma' \in I(\Gamma \Rightarrow \gamma)$ is provable in NL_S (similarly for variants). We leave it as an open problem whether the converse implication holds.

We define multivalued maps N_S' and P_S', which send each formula in language $(\cdot, \backslash, /)$ (possibly with 0, $\bot$, $\top$) into a finite set of such formulae.

$$N_S'(\gamma) = I(N_S(\gamma)), \;\; P_S'(\gamma) = I(P_S(\gamma))$$

$N_S'(\Gamma)$ is defined in a similar way as $I(\Gamma)$ above.

Theorem 2.3 Let S be a set of structural rules (a), (e), (i). Let $\Gamma \Rightarrow \gamma$ be a sequent in language $(\cdot, \backslash, /)$, possibly with 0, $\bot$, $\top$, such that $\Gamma \neq \lambda$. Then, $\Gamma \Rightarrow \gamma$ is provable in NL_S^* if and only if there exist $\Gamma' \in N_S'(\Gamma)$, $\gamma' \in P_S'(\gamma)$, such that $\Gamma' \Rightarrow \gamma'$ is provable in NL_S.

Proof Since FNL^*_S is conservative over NL^*_S, then, by Theorem 1, $\Gamma \Rightarrow \gamma$ is provable in NL^*_S if and only if $N_S(\Gamma) \Rightarrow P_S(\gamma)$ is provable FNL_S. The latter sequent satisfies the assumptions of Lemma 2.5: $\wedge$ (resp. $\vee$) can only be introduced by N_S (resp. P_S), and N_S (resp. P_S) acts on negative (resp. positive) occurrences of subformulae in $\Gamma \Rightarrow \gamma$, hence all occurrences of $\wedge$ (resp. $\vee$) in $N_S(\Gamma) \Rightarrow P_S(\gamma)$ are negative (resp. positive). Therefore, $N_S(\Gamma) \Rightarrow P_S(\gamma)$ is provable in FNL_S if and only if there exists $\Gamma' \Rightarrow \gamma' \in I(N_S(\Gamma) \Rightarrow P_S(\gamma))$, provable in NL_S. Clearly $\Gamma' \in N'_S(\Gamma)$, $\gamma' \in P'_S(\gamma)$.

2.4 Distributive and Modal Logics

We extend the results of Sect. 2.2 for logics corresponding to algebras based on distributive lattices, i.e., DFNL^*_S and DFNL_S, and logics with unary modalities $\Diamond, \Box$.

As we have noted in Sect. 2.1, DFNL can be axiomatized as FNL with (D), but this system does not allow cut elimination. Another axiomatization (of DFL_1) was proposed by Kozak [18], following similar solutions for relevant logics, due to J.M. Dunn and G. Mints. We recall it (for DFNL and its variants) with minor modifications.

The new axiomatization admits two structural operators: besides $(\,,\,)$, corresponding to product, also $(\,,\,)_\wedge$, corresponding to $\wedge$. Precisely, the formula structures are recursively defined as follows: (i) all formulae are (atomic) formula structures, (ii) if Γ, Δ are formula structures, then (Γ, Δ) and $(\Gamma, \Delta)_\wedge$ are formula structures. In models, $(\Gamma, \Delta)_\wedge$ is interpreted as $f(\Gamma) \wedge f(\Delta)$ (see the definition of f in Sect. 2.1).

The axioms and rules of DFNL are those of FNL except that $(\mathrm{L}\wedge)$ is replaced by:

$$(\mathrm{L}\wedge\mathrm{D})\ \frac{\Gamma[(\alpha, \beta)_\wedge] \Rightarrow \gamma}{\Gamma[\alpha \wedge \beta] \Rightarrow \gamma},$$

and one adds structural rules (a), (e), (i), (c) for $(\,,\,)_\wedge$.

$$(\mathrm{a}\wedge)\ \frac{\Gamma[((\Delta_1, \Delta_2)_\wedge, \Delta_3)_\wedge] \Rightarrow \gamma}{\Gamma[(\Delta_1, (\Delta_2, \Delta_3)_\wedge)_\wedge] \Rightarrow \gamma} \quad (\mathrm{e}\wedge)\ \frac{\Gamma[(\Delta_1, \Delta_2)_\wedge] \Rightarrow \gamma}{\Gamma[(\Delta_2, \Delta_1)_\wedge] \Rightarrow \gamma}$$

$$(\mathrm{i}\wedge)\ \frac{\Gamma[\Delta_i] \Rightarrow \alpha}{\Gamma[(\Delta_1, \Delta_2)_\wedge] \Rightarrow \gamma} \quad (\mathrm{c}\wedge)\ \frac{\Gamma[(\Delta, \Delta)_\wedge] \Rightarrow \alpha}{\Gamma[\Delta] \Rightarrow \gamma}$$

DFNL^* admits the empty formula structure λ. Notice that λ is the unit for $(\,,\,)$ but not for $(\,,\,)_\wedge$. By adding the constant 1 and $(\mathrm{L}1_l)$, $(\mathrm{L}1_r)$, $(\mathrm{R}1)$, we obtain DFNL_1. As above, we can also add any set S, of structural rules (a), (e), (i), (c) (for $(\,,\,)$), which yields logics DFNL_S, DFNL^*_S and DFNL_{1S}. Again, logics with (a) are denoted by DFL_S, DFL^*_S and DFL_{1S}.

Formula structures $(\lambda, \Delta)_\wedge$ and $(\Delta, \lambda)_\wedge$ can be introduced by $(\mathrm{i}\wedge)$; in general, they are not equal to Δ. Nonetheless, one may ignore them in any system DFNL^*_S. If S does not contain (i), then these structures cannot appear in any proof tree (also involving (CUT)) of a sequent not containing them. If S contains (i), then they can appear in such proofs (λ can be eliminated, using (i)), but we assume that λ is the unit for both (,) and $(\,,\,)_\wedge$; so $(\lambda, \Delta)_\wedge = (\Delta, \lambda)_\wedge = \Delta$.

Kozak [18] proves that DFL_1 without (CUT) is complete with respect to (finite) residuated distributive lattices, which yields an algebraic proof of the cut-elimination theorem ([18] employs quasi-embeddings, used earlier for other substructural logics in [2, 12, 31]). In a similar way (or by standard, syntactic arguments), one can prove the cut-elimination theorem for all systems DFNL_{1S}, DFNL^*_S, DFNL_S. All nonassociative logics DFNL_{1S}, DFNL^*_S, DFNL_S are decidable; each of them possesses the finite embeddability property, which yields the decidability of the corresponding consequence relation (with finitely many assumptions) [7, 9, 15]. Kozak [18] shows the finite model property of DFL_1, DFL_{1e}, DFL_{1i}, DFL_{1ei}, hence these logics are decidable. This yields the decidability of their 1-free fragments; similar methods bring the decidability of DFL, DFL_e, DFL_i, DFL_{ei}. With (i) all DFL_S are decidable; this problem is open for versions with (c) (but without (i)).

Theorem 2.1 remains true for DFNL^*_S versus DFNL_S with no essential changes in definitions and proofs (the new structural rules and (L$\wedge$D) cause no problems). N_S and P_S are defined by Table 2.1, and $N_S((\Gamma, \Delta)_\wedge) = (N_S(\Gamma), N_S(\Delta))_\wedge$. Warning: $\vdash^*$ and $\vdash$ denote here the provability in DFNL^*_S and DFNL_S, respectively. In the analogue of Theorem 2.1, we assume that no substructure of Γ is represented as $(\lambda, \Delta)_\wedge$ or $(\Delta, \lambda)_\wedge$.

Theorem 2.2 holds for associative systems. Since we do not represent formula structures as sequences, auxiliary axioms (a.1), (a.2), used in the proof, must be appropriately modified; the details are left to the reader.

The results of Sect. 2.3 remain true, but they bring nothing new, since the multiplicative fragment of DFNL^*_S (resp. DFNL_S) equals NL^*_S (resp. NL_S). This also holds for the fragments restricted to only negative occurrences of $\wedge$ and only positive occurrences of $\vee$.

Now we consider logics with unary modalities $\Diamond$, $\Box^\downarrow$, treated as multiplicative operators. The corresponding sequent systems admit one new unary structural operation, traditionally symbolized by $\langle\,\rangle$. The introduction rules for $\Diamond$, $\Box^\downarrow$ are as follows.

$$(\mathrm{L}\Diamond)\ \frac{\Gamma[\langle\alpha\rangle] \Rightarrow \beta}{\Gamma[\Diamond\alpha] \Rightarrow \beta}\quad (\mathrm{R}\Diamond)\ \frac{\Gamma \Rightarrow \alpha}{\langle\Gamma\rangle \Rightarrow \Diamond\alpha}$$

$$(\mathrm{L}\Box^\downarrow)\ \frac{\Gamma[\alpha] \Rightarrow \beta}{\Gamma[\langle\Box^\downarrow\alpha\rangle] \Rightarrow \beta}\quad (\mathrm{R}\Box^\downarrow)\ \frac{\langle\Gamma\rangle \Rightarrow \beta}{\Gamma \Rightarrow \Box^\downarrow\beta}$$

NL with $\Diamond$, $\Box^\downarrow$ and the above rules is denoted by NL$\Diamond$, and a similar notation is used for other systems, e.g. FNL$\Diamond^*$ is an analogous extension of FNL^*, $\mathrm{FNL}\Diamond_1$ of FNL_1, $\mathrm{FL}\Diamond_1$ of FL_1, and so on. These logics are special instances of systems of Full Generalized Lambek Calculus (FGL); see [7].

The cut-elimination theorem holds for systems of this kind, also with structural rules (a), (e), (i), (c); [26] provides some proofs. This remains true for these logics enriched with special modal rules:

$$(\mathrm{r.K}_s)\ \frac{\Gamma[(\langle\Delta_1\rangle, \langle\Delta_2\rangle)] \Rightarrow \gamma}{\Gamma[\langle(\Delta_1, \Delta_2)\rangle] \Rightarrow \gamma}\,,$$

$$(\mathrm{r.T})\ \frac{\Gamma[\langle\Delta\rangle] \Rightarrow \gamma}{\Gamma[\Delta] \Rightarrow \gamma}\quad (\mathrm{r.4})\ \frac{\Gamma[\langle\Delta\rangle] \Rightarrow \gamma}{\Gamma[\langle\langle\Delta\rangle\rangle] \Rightarrow \gamma}\,.$$

In systems with (CUT), these rules are equivalent to modal axioms:

$$(\mathrm{K}_s)\ \Diamond(\alpha \cdot \beta) \Rightarrow (\Diamond\alpha) \cdot (\Diamond\beta),\ \ (\mathrm{T.})\ \alpha \Rightarrow \Diamond\alpha,\ \ (4.)\ \Diamond\Diamond\alpha \Rightarrow \Diamond\alpha.$$

(K_s) is deductively equivalent to $\Box^{\downarrow}(\alpha\backslash\beta) \Rightarrow \Box^{\downarrow}\alpha\backslash\Box^{\downarrow}\beta$, the latter resembling the axiom (K) of classical modal logics. Without (c) all modal substructural logics of this kind are decidable. $\mathrm{FL}\Diamond_{1c}$ and $\mathrm{FL}\Diamond_c$ are undecidable, as conservative extensions of FL_{1c} and FL_c, respectively.

NL$\Diamond$ is complete with respect to residuated groupoids with unary operations $\Diamond, \Box^{\downarrow}$, satisfying *the unary residuation law*: $\Diamond a \leq b$ iff $a \leq \Box^{\downarrow} b$, for all elements a, b. Accordingly, $\Box^{\downarrow}$ is the right residual of $\Diamond$. Analogous completeness theorems hold for other logics of this kind with respect to the appropriate classes of algebras with $\Diamond, \Box^{\downarrow}$.

Substructural logics with $\Diamond, \Box^{\downarrow}$ are interesting for many reasons. Unary residuated pairs represent, in a sense, the basic species of residuation; up to the direction of order they amount to Galois connections. They were applied in type grammars in order to refine typing of natural language expressions; see [26, 29, 30]. Let us give one simple example. The initial types 'he': np, 'her': np, 'likes': (np\s)/np yield 'he likes her': s on the basis of NL. Proper nouns inhabit a subtype of np. Since $\Diamond\Box^{\downarrow}\alpha \Rightarrow \alpha$ is provable in NL$\Diamond$, one can define pn $= \Diamond\Box^{\downarrow}$np. Then, adjectives can be typed pn/pn, which yields 'he likes poor Jane': s, but 'he likes poor her': s cannot be derived, in accordance with the English grammar.

We extend the translations N_S, P_S for the language with $\Diamond, \Box^{\downarrow}$, by setting:

$$P_S(O\alpha) = OP_S(\alpha),\ N_S(O\alpha) = ON_S(\alpha),\ \text{for } O \in \{\Diamond, \Box^{\downarrow}\}.$$

Observe that sequents in $\mathrm{FNL}\Diamond^*_S$ can contain substructures $\langle\lambda\rangle$. All explicit occurrences of λ in nonempty formula structures are of this form. In opposition to the situation for DFNL^*_S, these substructures can appear in proof trees of sequents not containing them. For instance, (R$\Box^{\downarrow}$) infers $\Rightarrow \Box^{\downarrow}\alpha$ from $\langle\lambda\rangle \Rightarrow \alpha$. $\langle\lambda\rangle$ can be introduced by (R$\Diamond$) and (i).

A formula structure Γ is said to be λ-*free*, if $\Gamma \neq \lambda$ and Γ contains no substructure $\langle\lambda\rangle$. For any structure Γ, $N_S(\Gamma)$ is obtained by replacing each formula α occurring in Γ (as an atomic substructure) by $N_S(\alpha)$.

The results of Sect. 2.2 can be extended for logics with $\Diamond, \Box^{\downarrow}$. In the proofs of Lemmas 2.1 and 2.2, we use the monotonicity (hence also extensionality) of $\Diamond$ and $\Box^{\downarrow}$ in these logics.

For S not containing (i), the claim (I) of Theorem 1 holds for all sequents $\Gamma \Rightarrow \gamma$ such that Γ is λ-free. It is essential that in proof trees of these sequents, only λ-free structures appear in antecedents except for the empty antecedents (handled as in Sect. 2.1) and the subproofs of sequents $\Rightarrow \alpha$ (not essential in the proof of (I)). The proof follows the one in Sect. 2.1; the 'only if' part involves new cases, corresponding to the introduction rules for $\Diamond$, $\Box^{\downarrow}$.

We consider (R$\Box^{\downarrow}$). The premise is $\langle\Gamma\rangle \Rightarrow \beta$, and the conclusion is $\Gamma \Rightarrow \Box^{\downarrow}\beta$. By the induction hypothesis, $\langle N(\Gamma)\rangle \Rightarrow P(\beta)$ is provable in FNL$\Diamond_S$, hence $N(\Gamma) \Rightarrow \Box^{\downarrow}P(\beta)$ is provable in FNL$\Diamond_S$ and $\Box^{\downarrow}P(\beta) = P(\Box^{\downarrow}\beta)$. The remaining rules are left to the reader.

For S containing (i), we need an auxiliary notion. A pseudo-substitution θ is defined as follows: $\theta(\Gamma)$ replaces each explicit occurrence of λ in Γ by $q\backslash q$, where q is a fixed variable (q may occur in the sequents under consideration). As above, Theorem 2.1 remains true for λ-free Γ (in both (I) and (II)). It is easier to prove the following, stronger version.

Theorem 2.4 Let S contain (i). (I) For any sequent $\Gamma \Rightarrow \gamma$ such that $\Gamma \neq \lambda$, $\Gamma \Rightarrow \gamma$ is provable in FNL$\Diamond^*_S$ if and only if $N_S(\theta(\Gamma)) \Rightarrow P_S(\gamma)$ is provable in FNL$\Diamond_S$. (II) For any formula γ, if γ is provable in FNL$\Diamond^*_S$, then for any λ-free Γ, $\Gamma \Rightarrow P_S(\gamma)$ is provable in FNL$\Diamond_S$.

Proof The proof is similar to that of Theorem 2.1. The 'if' part of (I) follows from the analogue of Lemma 2.2, the equality $N_S(q\backslash q) = q\backslash q$, and the provability of $\lambda \Rightarrow q\backslash q$ in FNL$\Diamond^*_S$. (II) and the 'only if' part of (I) are proved by simultaneous induction on cut-free proofs in FNL$\Diamond^*_S$ (precisely: on the number of sequents appearing in the proof tree). Most arguments are entirely similar, and we do not repeat them. We only consider two new cases.

The proof of (I): (R$\Diamond$) with the premise $\Rightarrow \alpha$ and the conclusion $\langle\lambda\rangle \Rightarrow \Diamond\alpha$. By the induction hypothesis, applied to (II), $q\backslash q \Rightarrow P(\alpha)$ is provable in FNL$\Diamond_S$, hence $\langle q\backslash q\rangle \Rightarrow \Diamond P(\alpha)$ is provable in FNL$\Diamond_S$. So $N(\theta(\langle\lambda\rangle)) \Rightarrow P(\Diamond\alpha)$ is provable in FNL$\Diamond_S$.

The proof of (II): (R$\Box^{\downarrow}$) with the premise $\langle\lambda\rangle \Rightarrow \beta$ and the conclusion $\Rightarrow \Box^{\downarrow}\beta$. By the induction hypothesis, applied to (I), $\langle q\backslash q\rangle \Rightarrow P(\beta)$ is provable in FNL$\Diamond_S$, hence $q\backslash q \Rightarrow P(\Box^{\downarrow}\beta)$ is provable in FNL$\Diamond_S$. Let Γ be λ-free. FNL$\Diamond_S$ proves $\Gamma \Rightarrow q\backslash q$, by (Id), (i), (L\), hence also $\Gamma \Rightarrow P(\Box^{\downarrow}\beta)$, by (CUT).

These results can also be proved for DFNL$\Diamond^*_S$ versus DFNL$\Diamond_S$; λ-free formula structures are required to contain neither $\langle\lambda\rangle$, nor structures of the form $(\lambda, \Delta)_\wedge$, $(\Delta, \lambda)_\wedge$. All nonassociative logics DFNL$\Diamond_{1S}$, DFNL$\Diamond^*_S$. DFNL$\Diamond_S$ are decidable [7]. We leave it for further research how to formulate and to prove them for logics with special modal rules (axioms). One faces new difficulties, especially for logics without (i). For instance, (r.K$_S$) and (r.T) can eliminate $\langle\lambda\rangle$, hence $\Gamma \Rightarrow \gamma$, where Γ is λ-free, can be inferred from sequents with $\langle\lambda\rangle$.

Without (i), the provability in FNL$\Diamond^*_S$ cannot be fully reduced to the provability in FNL$\Diamond_S$. $\Box^{\downarrow}\alpha$ is provable if and only if $\langle\lambda\rangle \Rightarrow \alpha$ is provable, but the latter is not expressible in FNL$\Diamond_S$. With (i), using Theorem 2.4.(I), we obtain the conditions:

(Pr.5) $\vdash^* \Box^{\downarrow}\alpha$ iff $\vdash \langle q\backslash q\rangle \Rightarrow P(\alpha)$,
(Pr.6) $\nvdash^* \Diamond\alpha$.

Kuznetsov's reduction does not work for $\mathrm{FL}\Diamond_{1S}$, if (i) is not in S. $(1, \langle p\rangle) \Rightarrow \Diamond p$ is provable, by (Id), (R$\Diamond$), (L1_l), but $(q\backslash q, \langle((q\backslash q)\cdot p)\cdot(q\backslash q)\rangle) \Rightarrow \Diamond(((q\backslash q)\cdot p)\cdot(q\backslash q))$ is not provable.

The results of Sect. 2.3 can be extended for multiplicative fragments of $\mathrm{FNL}\Diamond^*_S$, $\mathrm{FNL}\Diamond_S$, if S does not contain (c). For unary operators O, one defines $I(O\alpha) = O(I(\alpha))$, where $O(U) = \{O\alpha : \alpha \in U\}$.

2.5 Applications

We discuss two applications of the translations N, P. First, we show that, essentially, the (phrase) languages generated by type grammars based on the logic $\mathcal{L}^*$, allowing empty antecedents, are also generated by type grammars based on the subsystem $\mathcal{L}$, not allowing empty antecedents.

We assume that $\mathcal{L}^*$ and $\mathcal{L}$ are formalized in the same language and satisfy (N-P): for any sequent $\Gamma \Rightarrow \gamma$ such that $\Gamma \neq \lambda$ and all atomic substructures of Γ are formulae of the language, $\Gamma \Rightarrow \gamma$ is provable in $\mathcal{L}^*$ if and only if $N(\Gamma) \Rightarrow P(\gamma)$ is provable in $\mathcal{L}$. N, P are some fixed translations within this formal language. We assume $N(p) = p$, for any variable p.

Proposition 2.1 *For any $\mathcal{L}^*$-grammar G, there exists an $\mathcal{L}$-grammar G' such that $L^P(G') = L^P(G) - \{\lambda\}$ and $L(G') = L(G) - \{\epsilon\}$.*

Proof Let $G = (\Sigma_G, I_G, \delta_G)$. We define G' by: $\Sigma_{G'} = \Sigma_G$, $\delta_{G'} = P(\delta_G)$, $I_{G'}(a) = \{N(\alpha) : \alpha \in I_G(a)\}$. Since $\mathcal{L}$ does not admit λ, then $\lambda \notin L^P(G')$. By induction on the size of X, one proves: for any $X \in \Sigma^P_G$, $I_{G'}(X) = \{N(\Gamma) : \Gamma \in I_G(X)\}$. This yields: $X \in L^P(G')$ iff there exists $\Delta \in I_{G'}(X)$ such that $\Delta \Rightarrow \delta_{G'}$ is provable in $\mathcal{L}$ iff there exists $\Gamma \in I_G(X)$ such that $N(\Gamma) \Rightarrow P(\delta_G)$ is provable in $\mathcal{L}$ iff there exists $\Gamma \in I_G(X)$ such that $\Gamma \Rightarrow \delta_G$ is provable in $\mathcal{L}^*$ iff $X \in L^P(G)$. So $L^P(G') = L^P(G) - \{\lambda\}$, and consequently, $L(G') = L(G) - \{\epsilon\}$.

We additionally assume that no variable in provable in $\mathcal{L}^*$. For extensions of FNL^*, this requirement is equivalent to the consistency of $\mathcal{L}^*$. Recall that a sequent system is *consistent*, if not all sequents are provable.

Corollary 2.2 *For any $\mathcal{L}^*$-grammar G such that δ_G is a variable, there exists an $\mathcal{L}$-grammar G' such that $L^P(G') = L^P(G)$ and $L(G') = L(G)$.*

Proof This follows from Proposition 2.1 and the unprovability of $\Rightarrow s_G$ (hence $\lambda \notin L^P(G)$).

These results (together with those from Sects. 2.2, 2.3 and 2.4) show that the generative capacity of type logics admitting λ is not essentially greater than that of type logics without λ, both for phrase languages and string languages.

Theorem 2.3 implies analogous consequences for type grammars without additives. Now $I_{G'}(a) = \bigcup\{N'(\alpha) : \alpha \in I_G(a)\}$. If δ_G is a variable, then $P'(\delta_G) = \{\delta_G\}$, which yields Corollary 2, for $\delta_{G'} = \delta_G$. If δ_G is compound, then $P'(\delta_G)$ may contain several types, say, $\delta_1, \ldots, \delta_n$. Let G'_i be defined as G' except that δ_i is the designated type. Clearly $L^P(G) - \{\lambda\}$ is the union of all $L^P(G'_i)$, for $i = 1, \ldots, n$, and similarly for string languages. All basic families of formal languages, e.g. regular languages, context-free languages, context-sensitive languages, r.e. languages, are closed under finite unions.

We omit a detailed discussion of the generative capacity of the particular classes of type grammars; the reader is referred to [4, 9, 29]. Let us only note that AB-grammars, L-grammars, NL-grammars generate the ϵ-free context-free languages, FL-grammars generate a proper superclass of ϵ-free context-free languages [17], but DFNL-grammars remain context-free [9]. Lin [23, 24] prove the context-freeness of type grammars based on some modal extensions of NL and DFNL.

The second application concerns the computational complexity of substructural logics. Horčik and Terui [16] prove the following, general theorem.

(HT) Every consistent substructural logic, possessing the disjunction property (DP), is PSPACE-hard.

By a substructural logic one means here an extension of FL_1 by additional axioms and rules (possibly in a richer language, e.g. with 0, $\bot$, $\top$; one can also add unary operators and others). (DP) means: if $\alpha \vee \beta$ is provable, then α is provable or β is provable.

Buszkowski [8] proves the following, stronger theorem: every logic $\mathcal{L}$ such that FL$\subseteq$ $\mathcal{L} \subseteq \mathcal{L}'$, for some consistent substructural logic $\mathcal{L}'$, possessing (DP), is PSPACE-hard. Here $\subseteq$ denotes the inclusion between the sets of provable sequents of the form $\alpha \Rightarrow \beta$ or $\Rightarrow \alpha$. The reader is referred to [8, 16] for a discussion of the significance of these results. For some particular logics, the PSPACE-hardness (PSPACE-completeness) was proved earlier, e.g. IL (intuitionistic logic), FL_1, MALL ((DP) holds for each of them). By the theorem from [8], FL_S and DFL_S are PSPACE-hard: they extend FL and are contained in IL (IL can be identified with $\mathrm{FL}_{1\bot eic}$ with $0 = \bot$). This cannot be inferred from (HT). See [8] for other examples.

Here we further generalize this theorem towards nonassociative logics. By $(a\downarrow)$ we denote the top-down direction of (a) (in algebras, this corresponds to right-associativity: $a \cdot (b \cdot c) \leq (a \cdot b) \cdot c$).

Proposition 2.2 *Every logic $\mathcal{L}$ such that $FNL_{(a\downarrow)} \subseteq \mathcal{L} \subseteq \mathcal{L}'$, for some consistent substructural logic $\mathcal{L}'$, possessing (DP), is PSPACE-hard.*

By symmetry, (a↓) can be replaced by (a↑), i.e., the bottom-up direction of (a). With (e) each of them yields (a).

To outline the proof we must recall some points of the proofs of (HT) and the version from [8].

Horčik and Terui [16] reduces the validity problem for closed quantified boolean formulae (QBFs) to the provability problem in the logic $\mathcal{L}$, fulfilling the assumptions of (HT). A closed QBF has the form $Q_n x_n \ldots Q_1 x_1 \varphi_0$, where $Q_i \in \{\forall, \exists\}$

and φ_0 is a boolean formula in DNF, whose variables are $x_1, \dots, x_n$. Let $\varphi_k = Q_k x_k \dots Q_1 x_1 \varphi_0$, for $k = 0, \dots, n$. The formula φ_k is encoded by a formula α_k of $\mathcal{L}$.

We recall the encoding with minor changes of notation. One fixes different variables $p_k, \bar{p}_k, q_k$, for $k = 1, \dots, n$. Let $\varphi_0 = \psi_1 \vee \dots \vee \psi_m$, where each ψ_j is a finite conjunction of literals. One encodes ψ_j by $\beta_j = \delta_1 \cdot \delta_2 \cdots \delta_n$, where: (1) $\delta_i = p_i$ if ψ_j contains the literal x_i, (2) $\delta_i = \bar{p}_i$ if ψ_j contains the literal $\neg x_i$, (3) $\delta_i = p_i \vee \bar{p}_i$ otherwise. One defines $\alpha_0 = \beta_1 \vee \dots \vee \beta_m$. For $k = 1, \dots, n$, one defines: (1) $\alpha_k = (p_k \vee \bar{p}_k) \backslash \alpha_{k-1}$ if $Q_k = \forall$, (2) $\alpha_k = (p_k \backslash q_k \vee \bar{p}_k \backslash q_k)/(\alpha_{k-1} \backslash q_k)$ if $Q_k = \exists$.

For $k = 0, \dots, n$, e_k denotes a partial valuation which assigns truth values to $x_{k+1}, \dots, x_n$; so e_n is the empty valuation. One represents e_k by the sequence $\varepsilon_k = (r_{k+1}, \dots, r_n)$, where: (1) $r_i = p_i$ if $e_k(x_i) = 1$, (2) $r_i = \bar{p}_i$ if $e_k(x_i) = 0$; so ε_n is the empty sequence. By set(ε_k) we denote the set of variables occurring in ε_k. Two crucial lemmas in [16] can be summarized as follows (in (iii) $\vdash$ denotes the provability from assumptions).

(T1) For any $k = 0, 1, \dots, n$, the following conditions are equivalent: (i) e_k satisfies φ_k, (ii) $\varepsilon_k \Rightarrow \alpha_k$ is provable in $\mathcal{L}$, (iii) set$(\varepsilon_k) \vdash \alpha_k$ in $\mathcal{L}$.

As a consequence, φ is valid if and only if α_n is provable in $\mathcal{L}$, which yields the desired reduction. (T1) is proved by simultaneous induction on k. The implication (ii) $\Rightarrow$(iii) is obvious, so the nontrivial steps are (i) $\Rightarrow$(ii) and (iii) $\Rightarrow$(i). For $k = 0$, (iii) $\Rightarrow$(i) employs Lemma A: in any nontrivial residuated lattice, 1 is not a minimal element. (DP) is used to prove the inductive step (iii) $\Rightarrow$(i) for $Q_k = \exists$. In fact, one uses Lemma B: if U is a set of variables and $U \vdash \alpha \vee \beta$ in $\mathcal{L}$, then $U \vdash \alpha$ or $U \vdash \beta$ in $\mathcal{L}$ (this follows from (DP)).

We have no space to recall more details of the proof. Already from the above it can be seen that empty antecedents and models with 1 play an essential role. Also the proof of the inductive step (iii)$\Rightarrow$(i) employs the law $\beta/(\gamma\backslash\gamma) \Rightarrow \beta$, provable in FL*, not in FL. Furthermore, (DP) lacks sense for logics not allowing empty antecedents. This proof cannot directly be adapted for logics like FL, DFL, etc.

Buszkowski [8] observes that the proof from [16] yields a stronger result: every logic $\mathcal{L}$ such that $\mathrm{FL}_1 \subseteq \mathcal{L} \subseteq \mathcal{L}'$, for some consistent substructural logic $\mathcal{L}'$, possessing (DP), is PSPACE-hard. For, if φ is valid, then α_n is provable in FL_1, hence in $\mathcal{L}$; if α_n is provable in $\mathcal{L}$, then α_n is provable in $\mathcal{L}'$, hence φ is valid. Therefore, φ is valid if and only if α_n is provable in $\mathcal{L}$.

Buszkowski [8] replaces FL_1 by FL. We recall some main points. First, φ is replaced by χ, which is obtained from φ by adding the literal x_{n+1} to every ψ_j. All valuations e_k are supposed to assign 1 to x_{n+1}; so e_n is defined for x_{n+1} only and assigns 1 to it. Clearly φ is valid if and only if e_n satisfies χ. We encode χ_k by α_k, as above (but now we have one new variable p_{n+1}; we do not use $\bar{p}_{n+1}$).

(T1) entails (T2): φ is valid if and only if $p_{n+1} \Rightarrow \alpha_n$ is provable in $\mathcal{L}'$. In particular, (T2) holds for $\mathcal{L}' = \mathrm{FL}_1$, and FL_1 can be replaced by FL*, since α_n does not contain 1. One also shows: $P(\alpha_k) = \alpha_k$, for $k = 0, \dots, n$; so $p_{n+1} \Rightarrow \alpha_n$ is stable in FL. The proof of this equality uses the unprovability of α_k, for any $k \leq n$.

By Corollary 2.1 and the above, φ is valid if and only if $p_{n+1} \Rightarrow \alpha_n$ is provable in FL. This equivalence holds for both FL and $\mathcal{L}'$, so it holds for $\mathcal{L}$ as well.

The proof does not work for nonassociative logics. A closer examination shows that (a) is essential precisely in the inductive step for (i)⇒(ii), with $Q_k = \exists$, in the proof of (T1). Let us recall the argument. Assume that e_k satisfies φ_k. Then, e_k can be extended to e_{k-1}, satisfying φ_{k-1}. Assume $e_{k-1}(x_k) = 1$. By the induction hypothesis, $p_k, \varepsilon_k \Rightarrow \alpha_{k-1}$ is provable. Hence $p_k, \varepsilon_k, \alpha_{k-1}\backslash q_k \Rightarrow q_k$ is provable, by (Id), (L\), which yields $\varepsilon_k, \alpha_{k-1}\backslash q_k \Rightarrow p_k\backslash q_k$, by (R\) (associativity). By (R∨), we get $\varepsilon_k, \alpha_{k-1}\backslash q_k \Rightarrow p_k\backslash q_k \vee \bar{p}_k\backslash q_k$, hence $\varepsilon_k \Rightarrow \alpha_k$, by (R/). For $e_{k-1}(x_k) = 0$, the argument is similar.

We can reconstruct the encoding and carry out all other steps of the proof in nonassociative logics: we replace sequences of formulae by formula structures with parentheses associated to the right, e.g. (p_1, p_2, p_3) by $(p_1, (p_2, p_3))$, and similarly for products of several formulae (in β_j). Lemma A holds for unital lattice-ordered residuated groupoids, and Lemma B for extensions of FNL_1. The argument in the preceding paragraph really employs (a↓). Indeed, from $((p_k, \varepsilon_k), \alpha_{k-1}\backslash q_k) \Rightarrow q_k$ we infer $(p_k, (\varepsilon_k, \alpha_{k-1}\backslash q_k)) \Rightarrow q_k$, then apply (R\), (R∨), (R/). This yields Proposition 2.2. As shown in [25], (a↓) is derivable in Residuated Basic Logic (RBL), being a conservative extension of Basic Propositional Logic, hence RBL is PSPACE-hard.

We can replace right-associativity by mixed associativity: $a \cdot (b \odot c) \leq (a \cdot b) \odot c$. Here $\cdot$ and $\odot$ are two products; \, / are the residual operations for the former, and $\backslash_{\odot}, /^{\odot}$ for the latter. FNL^2 denotes the variant of FNL admitting new connectives $\odot, \backslash_{\odot}, /^{\odot}$ and the new structural operator $(\ ,\ )_{\odot}$. The rule of *mixed associativity* is the following:

$$\text{(ma)}\ \frac{\Gamma[((\Delta_1, \Delta_2), \Delta_3)_{\odot}] \Rightarrow \gamma}{\Gamma[(\Delta_1, (\Delta_2, \Delta_3)_{\odot})] \Rightarrow \gamma}\,.$$

Algebraic models are lattice-ordered *double residuated groupoids*: the binary residuation law:

$$a \cdot b \leq c \text{ iff } b \leq a\backslash c \text{ iff } a \leq c/b$$

holds for both $\cdot, \backslash, /$ and $\odot, \backslash_{\odot}, /^{\odot}$. In unital algebras, 1 is the unit for $\cdot$ and $\odot$. The encoding is modified for $Q_k = \exists$: $\alpha_k = (p_k\backslash q_k \vee \bar{p}_k\backslash q_k)/^{\odot}(\alpha_{k-1}\backslash_{\odot} q_k)$. The proof, presented above, can be transformed into a proof of Proposition 2 with $\mathrm{FNL}^2_{(ma)}$ in the place of $\mathrm{FNL}_{(a\downarrow)}$.

Associativity can be entirely avoided in a different way. One may encode $Q_k = \exists$ by: $\alpha_k = p_k\backslash\alpha_{k-1} \vee \bar{p}_k\backslash\alpha_{k-1}$. Then, all arguments above remain correct for nonassociative logics. Unfortunately, the encoding is not polynomial. It becomes polynomial, if formulae are represented as directed acyclic graphs: each (free) subformula occupies only one node of the graph (like in circuits). Therefore, Proposition 2.2 is true for FNL in the place of $\mathrm{FNL}_{(a\downarrow)}$, with formulae represented as dags. With the standard representation, we can only prove the coNP-hardness (the validity of QBFs without ∃ can be reduced to the provability in FNL*, hence in FNL, as above).

References

1. Y. Bar-Hillel, Ch. Gaifman, and E. Shamir. On categorial and phrase structure grammars. *Bulletin Res. Council Israel*, F(9):155–166, 1960.
2. F. Belardinelli, P. Jipsen, and H. Ono. Algebraic aspects of cut elimination. *Studia Logica*, 77(2):209–240.
3. W. Buszkowski. Completeness results for Lambek syntactic calculus. *Zeitschrift f. mathematische Logik und Grundlagen der Mathematik*, 32(1):13–28, 1986.
4. W. Buszkowski. Mathematical linguistics and proof theory. In van Benthem and ter Meulen [33], pages 683–736.
5. W. Buszkowski. Finite models of some substructural logics. *Mathematical Logic Quarterly*, 48(1):63–72, 2002.
6. W. Buszkowski. Lambek calculus with nonlogical axioms. In Casadio et al. [10], pages 77–93.
7. W. Buszkowski. Interpolation and FEP for logics of residuated algebras. *Logic Journal of the IGPL*, 19(3):437–454, 2011.
8. W. Buszkowski. An interpretation of Full Lambek Calculus in its variant without empty antecedents of sequents. In N. Asher and S.Soloviev, editors, *Logical Aspects of Computational Linguistics. LACL 2014*, volume 8535 of *LNCS*, pages 30–43. 2014.
9. W. Buszkowski and M. Farulewski. Nonassociative Lambek calculus with additives and context-free languages. In *Languages: From Formal to Natural*, volume 5533 of *LNCS*, pages 45–58. 2009.
10. C. Casadio, P. J. Scott, and R. A. Seely, editors. *Language and Grammar. Studies in Mathematical Linguistics and Natural Language*. CSLI Publications, 2005.
11. A. Clark. A learnable representation for syntax using residuated lattices. In P. de Groote, M. Egg, and L. Kallmeyer, editors, *Formal Grammar. FG 2009*, volume 5591 of *Lecture Notes in Artificial Intelligence*, pages 183–198. Springer, 2011.
12. N. Galatos, P. Jipsen, T. Kowalski, and H. Ono. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*. Studies in Logic and The Foundations of Mathematics. Elsevier, 2007.
13. N. Galatos and H. Ono. Cut elimination and strong separation for substructural logics: an algebraic approach. *Annals of Pure and Applied Logic*, 161(9):1097–1133, 2010.
14. J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
15. Z. Haniková and R. Horčik. The finite embeddability property for residuated groupoids. *Algebra Universalis*, 72(1):1–13, 2014.
16. R. Horčik and K. Terui. Disjunction property and complexity of substructural logics. *Theoretical Computer Science*, 412:3992–4006, 2011.
17. M. Kanazawa. The Lambek calculus enriched with additional connectives. *Journal of Logic, Language and Information*, 1(2), 1992.
18. M. Kozak. Distributive Full Lambek Calculus has the finite model property. *Studia Logica*, 91(2):201–216, 2009.
19. S. Kuznetsov. Lambek grammars with the unit. In P. de Groote and M.-J. Nederhof, editors, *Formal Grammar 2010/2011*, volume 7395 of *LNCS*, pages 262–266. Springer, 2012.
20. J. Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.
21. J. Lambek. On the calculus of syntactic types. In R. Jakobson, editor, *Structure of Language and Its Mathematical Aspects*, pages 166–178. American Mathematical Society, Providence, 1961.
22. J. Lambek. *From Word to Sentence: a computational algebraic approach to grammar*. Polimetrica, 2008.
23. Z. Lin. Modal nonassociative Lambek calculus with assumptions: complexity and context-freeness. In *Language and Automata Theory and Applications. LATA 2010*, volume 6031 of *LNCS*, pages 414–425. Springer, 2010.
24. Z. Lin. Distributive Full Nonassociative Lambek Calculus with S4-modalities is context-free. In *Logical Aspects of Computational Linguistics. LACL 2012*, volume 7351 of *LNCS*, pages 161–172. Springer, 2012.

25. M. Ma and Z. Lin. Residuated basic logic. Paper presented at AWPL 2, Guangzhou, 2014.
26. M. Moortgat. Multimodal linguistic inference. *Journal of Logic, Language and Information*, 5(3–4):349–385, 1996.
27. M. Moortgat. Categorial Type Logics. In van Benthem and ter Meulen [33], pages 93–177.
28. M. Moortgat and R.T. Oehrle. Pregroups and type-logical grammar: Searching for convergence. In Casadio et al. [10], pages 141–160.
29. R. Moot and C. Retoré. *The Logic of Categorial Grammars. A Deductive Account of Natural Language Syntax and Semantics*, volume 6850 of *LNCS*. Springer, 2012.
30. G. Morrill. *Type Logical Grammar*. Kluwer, Dordrecht, 1994.
31. M. Okada and K. Terui. The finite model property for various fragments of intuitionistic linear logic. *Journal of Symbolic Logic*, 64:790–802, 1999.
32. H. Ono and Y. Komori. Logics without the contraction rule. *Journal of Symbolic Logic*, 50:169–201, 1985.
33. J. van Benthem and A. ter Meulen, editors. *Handbook of Logic and Language*. Elsevier, 1997.

http://www.springer.com/978-3-662-47196-8

Modality, Semantics and Interpretations
The Second Asian Workshop on Philosophical Logic
Ju, S.; Liu, H.; Ono, H. (Eds.)
2015, VI, 188 p. 8 illus., Hardcover
ISBN: 978-3-662-47196-8