Nonassociative Lambek Calculus with Additives and Context-Free Languages

Wojciech Buszkowski[1,2] and Maciej Farulewski[1]

[1]Faculty of Mathematics and Computer Science
Adam Mickiewicz University in Poznań
Chair of Logic and Computation
[2]University of Warmia and Mazury in Olsztyn
buszko@amu.edu.pl, maciejf@amu.edu.pl

Abstract. We study Nonassociative Lambek Calculus with additives $\wedge, \vee$, satisfying the distributive law (Distributive Full Nonassociative Lambek Calculus **DFNL**). We prove that categorial grammars based on **DFNL**, also enriched with assumptions, generate context-free languages. The proof uses proof-theoretic tools (interpolation) and a construction of a finite model, earlier employed in [11] in the proof of Finite Embeddability Property (FEP) of **DFNL**; our paper is self-contained, since we provide a simplified version of the latter proof. We obtain analogous results for different variants of **DFNL**, e.g. **BFNL**, which admits negation $\neg$ such that $\wedge, \vee, \neg$ satisfy the laws of boolean algebra, and **HFNL**, corresponding to Heyting algebras with an additional residuation structure. Our proof also yields Finite Embeddability Property of boolean-ordered and Heyting-ordered residuated groupoids. The paper joins proof-theoretic and model-theoretic techniques of modern logic with standard tools of mathematical linguistics.

1 Introduction

Nonassociative Lambek Calculus **NL** proves the order formulas $\alpha \leq \beta$, valid in *residuated groupoids*, i.e. ordered algebras $(M, \cdot, \backslash, /, \leq)$ such that $(M, \leq)$ is a poset, and $\cdot, \backslash, /$ are binary operations on M, satisfying the residuation law:

$$a \cdot b \leq c \text{ iff } b \leq a\backslash c \text{ iff } a \leq c/b\,, \tag{1}$$

for all $a, b, c \in M$. As an easy consequence of (1), we obtain:

$$a(a\backslash b) \leq b\,,\ (a/b)b \leq a\,, \tag{2}$$

$$\text{if } a \leq b \text{ then } ca \leq cb\,, ac \leq bc\,, c\backslash a \leq c\backslash b\,, a/c \leq b/c\,, b\backslash c \leq a\backslash c\,, c/b \leq c/a\,, \tag{3}$$

for all $a, b, c \in M$. Hence every residuated groupoid is a partially ordered groupoid, if one forgets residuals $\backslash, /$ (we refer to $\cdot$ as product).

NL was introduced by Lambek [19] as a variant of Syntactic Calculus [18], now called Associative Lambek Calculus **L**, which yields the order formulas

O. Grumberg et al. (Eds.): Francez Festschrift, LNCS 5533, pp. 45–58, 2009.

valid in residuated semigroups ($\cdot$ is associative). Both are standard type logics for categorial grammars [3,8,21,22]. While **L** is appropriate for expressions in the form of strings, **NL** corresponds to tree structures. The cut-elimination theorem holds for **NL** and **L**, and it yields the decidability of these systems [18,19].

NL and **L** are examples of *substructural logics*, i.e. non-classical logics whose sequent systems lack some structural rules (Weakening, Contraction, Exchange). Besides multiplicatives $\cdot, \backslash, /$, substructural logics usually admit additives $\wedge, \vee$. The related algebras are residuated lattices $(M, \wedge, \vee, \cdot, \backslash, /, 1)$: here $(M, \wedge, \vee)$ is a lattice, $(M, \cdot, 1)$ is a monoid (i.e. a semigroup with 1), and (1) holds. For the nonassociative case, monoids are replaced by groupoids or unital groupoids (i.e. groupoids with 1); the resulting algebras are called lattice-ordered residuated (unital) groupoids. An algebra of that kind is said to be distributive, if its lattice reduct is distributive. The complete logic for residuated lattices is Full Lambek Calculus **FL**; it admits cut-elimination and is decidable [13]. **FL** amounts to the *-free fragment of Pratt's action logic [10].

Full Nonassociative Lambek Calculus **FNL** is the complete logic of lattice-ordered residuated groupoids. We present it in the form of a sequent system. The cut-elimination theorem holds for this system [13]. It is not useful for our purposes, since we consider the consequence relation of **FNL** which requires the cut rule to be complete with respect to algebraic models. Furthermore, our main issue is **DFNL** in which the distributive law is affixed as a new axiom; the cut rule is necessary in **DFNL**.

Categorial grammars based on **NL** generate precisely the $\epsilon-$free context-free languages [7,17,15]. Pentus [24] proves the same for **L**. Using **FL**, even without $\vee$, one can generate languages which are not context-free, i.e. meets of two context-free languages [16]. This also holds for **FL** with distribution, since it is conservative over its $\vee-$free fragment. The provability problem for **L** is NP-complete [25]; for **FL**, the upper bound is P-SPACE. Categorial grammars with (partial) commutation can generate non-context-free languages [6,12].

We prove that (in opposition to **FL**) categorial grammars based on **DFNL** generate context-free languages, and it remains true if one adds an arbitrary finite set of assumptions to **DFNL**. For **NL**, an analogous result has been proved in [9]. The latter paper also proves the polynomial time decidability of the consequence relation of **NL** (extending a result of [14] for pure **NL**), but it does not hold in the presence of additives. The consequence relation of **DFNL** is decidable; this follows from FEP. It is known that the consequence relation for **L** is undecidable [9].

The construction of a finite model, used in the proof of FEP for **DFNL** in [11], will also be employed here (in a modified form) in order to prove the subformula property (of logics with cut) and, consequently, an interpolation lemma in an unrestricted version (see section 3).

Our methods can be extended to multi-modal variants of **DFNL** which admit several 'product' operations (of arbitrary arity) and the corresponding residual operations. Without additives, this multi-modal framework was presented in [8,9]. It is also naturally related to multi-modal extensions of Lambek Calculus, studied in e.g. [21,22]. This leads us to the proof of FEP for **BFNL**, which is the complete logic of boolean-ordered residuated groupoids, and the context-freeness of the corresponding grammars. Similar results are obtained for **HFNL**, which is the complete logic of Heyting-ordered residuated groupoids. Our results cannot directly be adapted for systems describing non-distributive lattices.

(External) consequence relations for substructural logics have been studied in different contexts; see e.g. [1,5,13]. Put it differently, one studies logics enriched with (finitely many) assumptions. Assumptions are sequents (not closed under substitution) added to axioms of the system (with the cut rule). Categorial grammars are usually required to be lexical in the sense that the logic is common for all languages and all information on the particular language is contained in the type lexicon. But, there are approaches allowing non-lexical assumptions, which results in a more efficient description of the language and an increase of generative power [8,20]. Let us emphasize that our results on context-freeness are new even for pure logics **DFNL**, **BFNL** and **HFNL**, and assumptions do not change anything essential in proofs.

Categorial grammars with additives are not popular in the linguistic literature. There are, nonetheless, good reasons for studying them. As it has been mentioned above, additives are standard operations in linear logics and other substructural logics [13]. The syntactic category of type α is usually understood as the set of all strings (or: trees; this seems more natural for the nonassociative case) which are assigned type α by the grammar. Then, it is natural to consider basic boolean operations on sets and to explicitely represent them in the grammar formalism. In general, they can result in a refinement of language description.

Types with $\wedge$ were used already in Lambek [19] in order to change a finite type assignment $a \mapsto \alpha_i$, $i = 1, \ldots, n$, into a rigid type assignment $a \mapsto \alpha_1 \wedge \cdots \wedge \alpha_n$ in the type lexicon of a categorial grammar (see section 4 for a definition of a categorial grammar). Standard basic categories, e.g. 's' (sentence), 'np' (noun phrase), 'pro' (pronoun), are not sufficient for a really effective description of natural language. One can divide them in subcategories. Lambek [20] considers three types of sentence: s_1 (sentence in the present tense), s_2 (sentence in the past tense), and s (sentence when the tense is not relevant) and four types of personal pronouns: π_1, π_2, π_3 for first, second and third person pronoun, respectively, and π when the person is irrelevant. This naturally leads to non-lexical assumptions, like $s_i \Rightarrow s$, $\pi_i \Rightarrow \pi$, added to the basic logic. An alternative solution is to define $s = s_1 \vee s_2$, $\pi = \pi_1 \vee \pi_2 \vee \pi_3$ which introduces $\vee$ on the scene. Kanazawa [16] proposes feature decomposition of basic categories; 'walks' is assigned type $(\mathrm{np} \wedge \mathrm{sing})\backslash s$, 'walk' type $(\mathrm{np} \wedge \mathrm{pl})\backslash s$, 'walked' type $\mathrm{np} \backslash s$, 'John' type $\mathrm{np} \wedge \mathrm{sing}$, 'the Beatles' type $\mathrm{np} \wedge \mathrm{pl}$, and 'became' type $(\mathrm{np} \backslash s)/(\mathrm{np} \vee \mathrm{ap})$, where 'ap' stands for 'adjective phrase'. Types with negation can be employed to represent a negative information; for instance, John $\mapsto \neg s$ means that 'John' is not a sentence. It is well-known that negative information makes learning algorithms for formal grammars more efficient. It opens an interesting area of application for grammars with negation.

2 Restricted Interpolation

We admit a denumerable set of variables $p, q, r, \ldots$. Formulas are built from variables by means of $\cdot, \backslash, /, \wedge, \vee$. Formula structures (shortly: structures) are built from formulas according to the rules: (1) every formula is a structure, (2) if X, Y are structures then (X, Y) is a structure. We denote arbitrary formulas by $\alpha, \beta, \gamma, \ldots$ and structures by X, Y, Z. *A context* is a structure $X[\circ]$ containing a single occurrence of a special substructure $\circ$ (a place for substitution); $X[Y]$ denotes the result of substitution of Y for $\circ$ in $X[\circ]$.

Sequents are of the form $X \Rightarrow \alpha$. **FNL** assumes the following axioms and inference rules:

$$(\mathrm{Id})\ \alpha \Rightarrow \alpha\,,$$

$$(\cdot\mathrm{L})\ \frac{X[(\alpha,\beta)] \Rightarrow \gamma}{X[\alpha\cdot\beta] \Rightarrow \gamma},\ (\cdot\mathrm{R})\ \frac{X \Rightarrow \alpha;\ Y \Rightarrow \beta}{(X,Y) \Rightarrow \alpha\cdot\beta},$$

$$(\backslash\mathrm{L})\ \frac{X[\beta] \Rightarrow \gamma;\ Y \Rightarrow \alpha}{X[(Y,\alpha\backslash\beta)] \Rightarrow \gamma},\ (\backslash\mathrm{R})\ \frac{(\alpha,X) \Rightarrow \beta}{X \Rightarrow \alpha\backslash\beta},$$

$$(/\mathrm{L})\ \frac{X[\beta] \Rightarrow \gamma;\ Y \Rightarrow \alpha}{X[(\beta/\alpha,Y)] \Rightarrow \gamma},\ (/\mathrm{R})\ \frac{(X,\alpha) \Rightarrow \beta}{X \Rightarrow \beta/\alpha},$$

$$(\wedge\mathrm{L})\ \frac{X[\alpha_i] \Rightarrow \beta}{X[\alpha_1\wedge\alpha_2] \Rightarrow \beta},\ (\wedge\mathrm{R})\ \frac{X \Rightarrow \alpha;\ X \Rightarrow \beta}{X \Rightarrow \alpha\wedge\beta},$$

$$(\vee\mathrm{L})\ \frac{X[\alpha] \Rightarrow \gamma;\ X[\beta] \Rightarrow \gamma}{X[\alpha\vee\beta] \Rightarrow \gamma},\ (\vee\mathrm{R})\ \frac{X \Rightarrow \alpha_i}{X \Rightarrow \alpha_1\vee\alpha_2},$$

$$(\mathrm{CUT})\ \frac{X[\alpha] \Rightarrow \beta;\ Y \Rightarrow \alpha}{X[Y] \Rightarrow \beta}.$$

In $(\wedge\mathrm{L})$ and $(\vee\mathrm{R})$, the subscript i equals 1 or 2. The latter rules and $(\cdot\mathrm{L})$, $(\backslash\mathrm{R})$, $(/\mathrm{R})$ have one premise; the remaining rules have two premises, separated by semicolon. **DFNL** admits the additional axiom scheme:

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

Notice that the converse sequent is provable in **FNL**. (CUT) can be eliminated from **FNL** but not from **DFNL**. For instance, $\alpha\wedge(\beta\vee(\gamma\vee\delta)) \Rightarrow (\alpha\wedge\beta)\vee((\alpha\wedge\gamma)\vee(\alpha\wedge\delta))$ cannot be proved without (CUT).

Let Φ be a set of sequents. We write $\Phi \vdash X \Rightarrow \alpha$ if $X \Rightarrow \alpha$ is deducible from Φ in **DFNL**. By $F(X)$ we denote the formula arising from X after one has replaced each comma by $\cdot$. By $(\cdot\mathrm{L})$ and (Id), $(\cdot\mathrm{R})$, (CUT), $X \Rightarrow \alpha$ and $F(X) \Rightarrow \alpha$ are mutually deducible. Consequently, without loss of generality we can assume that Φ consists of sequents of the form $\alpha \Rightarrow \beta$ (simple sequents). In models, $\Rightarrow$ is interpreted as $\leq$ and, by definition, an assignment f satisfies $X \Rightarrow \alpha$ iff $f(F(X)) \leq f(\alpha)$.

In what follows, we always assume that Φ is a finite set of simple sequents. T denotes a set of formulas. By a $T-$sequent we mean a sequent such that all formulas occurring in it belong to T. We write $X \Rightarrow_{\Phi,T} \alpha$ if $X \Rightarrow \alpha$ has a deduction from Φ in **DFNL** which consists of $T-$sequents only (then, $X \Rightarrow \alpha$ must be a $T-$sequent). In proofs we write $\Rightarrow_T$ for $\Rightarrow_{\Phi,T}$. The following lemma is proved for **DFNL** but the same proof works for **FNL**.

Lemma 1. *Let T be closed under $\wedge, \vee$. Let $X[Y] \Rightarrow_{\Phi,T} \gamma$. Then, there exists $\delta \in T$ such that $X[\delta] \Rightarrow_{\Phi,T} \gamma$ and $Y \Rightarrow_{\Phi,T} \delta$.*

Proof. δ is called an interpolant of Y in $X[Y] \Rightarrow \gamma$. The proof proceeds by induction on $T-$deductions of $X[Y] \Rightarrow \gamma$ from Φ.

The case of axioms and assumptions is easy; they are simple sequents $\alpha \Rightarrow \gamma$, so $Y = \alpha$ and $\delta = \alpha$.

Let $X[Y] \Rightarrow \gamma$ be the conclusion of a rule. (CUT) is easy. If Y comes from one premise of (CUT), then we take an interpolant from this premise. Otherwise Y must contain Z, where the premises are $X[\alpha] \Rightarrow \gamma$, $Z \Rightarrow \alpha$. So, $Y = U[Z]$, and it comes from $U[\alpha]$ in the first premise. Then, an interpolant δ of $U[\alpha]$ in this premise is also an interpolant of Y in the conclusion, by (CUT).

Let us consider other rules. First, we assume that Y does not contain the formula, introduced by the rule (the active formula). If Y comes from exactly one premise of the rule, then one takes an interpolant from this premise. Let us consider ($\wedge$R). The premises are $X[Y] \Rightarrow \alpha$, $X[Y] \Rightarrow \beta$, and the conclusion is $X[Y] \Rightarrow \alpha \wedge \beta$. By the induction hypothesis, there are interpolants δ of Y in the first premise and δ' of Y in the second one. We have $X[\delta] \Rightarrow_T \alpha$, $X[\delta'] \Rightarrow_T \beta$, $Y \Rightarrow_T \delta$, $Y \Rightarrow_T \delta'$. Then, $\delta \wedge \delta'$ is an interpolant of Y in the conclusion, by ($\wedge$L), ($\wedge$R). Let us consider ($\vee$L). The premises are $X[\alpha][Y] \Rightarrow \gamma$, $X[\beta][Y] \Rightarrow \gamma$, and the conclusion is $X[\alpha \vee \beta][Y] \Rightarrow \gamma$, where Y does not contain $\alpha \vee \beta$. As above, there are interpolants δ, δ' of Y in the premises. Again $\delta \wedge \delta'$ is an interpolant of Y in the conclusion, by ($\wedge$L), ($\vee$L) and ($\wedge$R). For ($\cdot$R) with premises $U \Rightarrow \alpha$, $V \Rightarrow \beta$ and conclusion $(U, V) \Rightarrow \alpha \cdot \beta$, if $Y = (U, V)$, then we take $\delta = \alpha \cdot \beta$.

Second, we assume that Y contains the active formula (so, the rule must be an L-rule). If Y is a single formula, then we take $\delta = Y$. Assume that Y is not a formula. For ($\cdot$L), ($\wedge$L), we take an interpolant of Y' in the premise, where Y' is the natural source of Y. For ($\backslash$L) with premises $X[\beta] \Rightarrow \gamma$, $Z \Rightarrow \alpha$ and conclusion $X[(Z, \alpha\backslash\beta)] \Rightarrow \gamma$, we consider the source Y' of Y (Y' occurs in $X[\beta]$ and contains β). Then, Y arises from Y' by substituting $(Z, \alpha\backslash\beta)$ for β. Hence, an interpolant of Y' in the first premise is also an interpolant of Y in the conclusion, by ($\backslash$L). The case of (/L) is similar. The final case is ($\vee$L) with premises $Z[U[\alpha]] \Rightarrow \gamma$, $Z[U[\beta]] \Rightarrow \gamma$ and conclusion $Z[U[\alpha \vee \beta]] \Rightarrow \gamma$, where $Y = U[\alpha \vee \beta]$. Let δ be an interpolant of $U[\alpha]$ in the first premise and δ' be an interpolant of $U[\beta]$ in the second premise. Then, $\delta \vee \delta'$ is an interpolant of Y in the conclusion, by ($\vee$L),($\vee$R). $\square$

3 Finite Models and Interpolation

We prove an (extended) subformula property and an interpolation lemma for the deducibility relation $\vdash$ in **DFNL**. We need some constructions of lattice-ordered residuated groupoids.

Let $(M, \cdot)$ be a groupoid. On the powerset $P(M)$ one defines operations: $U \cdot V = \{ab : a \in U, b \in V\}$, $U\backslash V = \{c \in M : U \cdot \{c\} \subseteq V\}$, $U/V = \{c \in M : \{c\} \cdot V \subseteq U\}$, $U \vee V = U \cup V$, $U \wedge V = U \cap V$. $P(M)$ with these operations is a distributive lattice-ordered groupoid (it is a complete lattice). The order is $\subseteq$.

An operator $C : P(M) \mapsto P(M)$ is called *a closure operator* (or: a nucleus) on $(M, \cdot)$, if it satisfies the following conditions: (C1) $U \subseteq C(U)$, (C2) if $U \subseteq V$ then $C(U) \subseteq C(V)$, (C3) $C(C(U)) \subseteq C(U)$, (C4) $C(U) \cdot C(V) \subseteq C(U \cdot V)$, for all $U, V \subseteq M$ [13]. A set $U \subseteq M$ is said to be closed, if $C(U) = U$. By $C(M, \cdot)$ we denote the family of all closed subsets of M. Operations on $C(M, \cdot)$ are defined as follows: $U \otimes V = C(U \cdot V)$, $U\backslash V$, U/V and $U \wedge V$ as above, $U \vee V = C(U \cup V)$. (The product operation in $C(M, \cdot)$ is denoted by $\otimes$ to avoid collision with $\cdot$ in $P(M)$.) It is known that $C(M, \cdot)$ with these operations is a complete lattice-ordered residuated groupoid [13]; it need not be distributive. The order is $\subseteq$.

(C4) is essential in the proof that $C(M, \cdot)$ is closed under $\backslash, /$. Actually, if U is closed, then $V\backslash U$ and U/V are closed, for any $V \subseteq M$. Let us consider U/V. Since (2) hold in $P(M)$, then $(U/V) \cdot V \subseteq U$. We get $C(U/V) \cdot V \subseteq C(U/V) \cdot C(V) \subseteq C((U/V) \cdot V) \subseteq C(U) = U$, and consequently, $C(U/V) \subseteq U/V$, by (1) for $P(M)$. The reader is invited to prove that (1) holds in $C(M, \cdot)$ and $C(U \cup V)$ is the join of U, V in $C(M, \cdot)$.

Let T be a nonempty set of formulas. By T^* we denote the set of all structures formed out of formulas from T. $T^*[\circ]$ denotes the set of all contexts $X[\circ]$ whose all atomic substructures different from $\circ$ belong to T.

T^* is a (free) groupoid with the operation $X \cdot Y = (X, Y)$. Hence $P(T^*)$ is a lattice-ordered residuated groupoid with operations defined as above. For $Z[\circ] \in T^*[\circ]$ and $\alpha \in T$, we define a set:

$$[Z[\circ], \alpha] = \{X \in T^* : Z[X] \Rightarrow_{\Phi,T} \alpha\}. \tag{4}$$

The family of all sets $[Z[\circ], \alpha]$, defined in this way, is denoted $B(T)$. An operator $C_T : P(T^*) \mapsto P(T^*)$ is defined as follows:

$$C_T(U) = \bigcap \{[Z[\circ], \alpha] \in B(T) : U \subseteq [Z[\circ], \alpha]\}\,, \tag{5}$$

for $U \subseteq T^*$. It is easy to see that C_T satisfies (C1), (C2), (C3). We prove (C4). Let $U, V \subseteq T^*$ and $X \in C_T(U)$, $Y \in C_T(V)$. We show $(X, Y) \in C_T(U \cdot V)$. Let $[Z[\circ], \alpha] \in B(T)$ be such that $U \cdot V \subseteq [Z[\circ], \alpha]$. For any $X' \in U$, $Y' \in V$, $(X', Y') \in [Z[\circ], \alpha]$, whence $Z[(X', Y')] \Rightarrow_T \alpha$. So, $U \subseteq [Z[(\circ, Y')], \alpha]$, whence $C_T(U) \subseteq [Z[(\circ, Y')], \alpha]$, by (5), and the latter holds for any $Y' \in V$. Then, $Z[(X, Y')] \Rightarrow_T \alpha$, for any $Y' \in V$. We get $V \subseteq [Z[(X, \circ)], \alpha]$, which yields $C_T(V) \subseteq [Z[(X, \circ)], \alpha]$, by (5). Consequently, $Z[(X, Y)] \Rightarrow_T \alpha$, whence $(X, Y) \in [Z[\circ], \alpha]$ (see [23,2,13,11] for similar arguments).

We have shown that C_T is a closure operator on $(T^*, \cdot)$. We consider the algebra $C_T(T^*, \cdot)$, further denoted by $M(T, \Phi)$. Clearly, all sets in $B(T)$ are closed under C_T. We define:

$$[\alpha] = [\circ, \alpha] = \{X \in T^*; X \Rightarrow_{\Phi,T} \alpha\}\,. \tag{6}$$

For $\alpha \in T$, $[\alpha] \in B(T)$. The following equations are true in $M(T,\Phi)$ provided that all formulas appearing in them belong to T.

$$[\alpha] \otimes [\beta] = [\alpha \cdot \beta],\ [\alpha]\backslash[\beta] = [\alpha\backslash\beta],\ [\alpha]/[\beta] = [\alpha/\beta]\,, \tag{7}$$

$$[\alpha] \vee [\beta] = [\alpha \vee \beta],\ [\alpha] \wedge [\beta] = [\alpha \wedge \beta]\,. \tag{8}$$

We prove the first equation (7). If $X \Rightarrow_T \alpha$ and $Y \Rightarrow_T \beta$ then $(X,Y) \Rightarrow_T \alpha\cdot\beta$, by ($\cdot$R). Consequently, $[\alpha]\cdot[\beta] \subseteq [\alpha\cdot\beta]$. Then $[\alpha]\otimes[\beta] = C_T([\alpha]\cdot[\beta]) \subseteq [\alpha\cdot\beta]$, by (C2), (C3). We prove the converse inclusion. Let $[Z[\circ],\gamma] \in B(T)$ be such that $[\alpha]\cdot[\beta] \subseteq [Z[\circ],\gamma]$. By (Id), $\alpha \in [\alpha]$, $\beta \in [\beta]$, whence $Z[(\alpha,\beta)] \Rightarrow_T \gamma$. Then, $Z[\alpha\cdot\beta] \Rightarrow_T \gamma$, by ($\cdot$R). Hence, if $X \in [\alpha\cdot\beta]$ then $Z[X] \Rightarrow_T \gamma$, by (CUT), which yields $X \in [Z[\circ],\gamma]$. We have shown $[\alpha\cdot\beta] \subseteq C_T([\alpha]\cdot[\beta])$.

We prove the second equation (7). Let $X \in [\alpha]\backslash[\beta]$. Since $\alpha \in [\alpha]$, then $(\alpha,X) \in [\beta]$. Hence $(\alpha,X) \Rightarrow_T \beta$, which yields $X \Rightarrow_T \alpha\backslash\beta$, by ($\backslash$R). We have shown $\subseteq$. To prove the converse inclusion it suffices to show $[\alpha]\cdot[\alpha\backslash\beta] \subseteq [\beta]$. If $X \Rightarrow_T \alpha$ and $Y \Rightarrow_T \alpha\backslash\beta$, then $(X,Y) \Rightarrow_T \beta$, since $(\alpha,\alpha\backslash\beta) \Rightarrow_T \beta$, by (Id), ($\backslash$L), and one applies (CUT). The proof of the third equation (7) is similar. Proofs of (8) are left to the reader.

We say that formulas $\alpha,\beta \in T$ are $T-$equivalent, if $\alpha \Rightarrow_T \beta$ and $\beta \Rightarrow_T \alpha$. By (Id), (CUT), $T-$equivalence is an equivalence relation. By $\overline{T}$ we denote the smallest set of formulas which contains all formulas from T and is closed under subformulas and $\wedge,\vee$. If T is closed under subformulas, then $\overline{T}$ is the closure of T under $\wedge,\vee$. The following lemma is a syntactic variant of the well-known fact that distributive lattices are locally finite (it means: every finitely generated distributive lattice is finite).

Lemma 2. *If T is a finite set of formulas, then $\overline{T}$ is finite up to $\overline{T}-$equivalence.*

Proof. If T is finite, then the set T' of subformulas of formulas from T is also finite. $\overline{T}$ is the closure of T' under $\wedge,\vee$. The converse of (D) $(\alpha\wedge\beta)\vee(\alpha\wedge\gamma) \Rightarrow \alpha\wedge(\beta\vee\gamma)$ is provable in **FNL** (it is valid in all lattices); if $\alpha,\beta,\gamma \in \overline{T}$, then the proof uses $\overline{T}-$sequents only. Consequently, for $\alpha,\beta,\gamma \in \overline{T}$, both sides of (D) are $\overline{T}-$equivalent. It follows that every formula from $\overline{T}$ is $\overline{T}-$equivalent to a finite disjunction of finite conjunctions of formulas from T'. If one omits repetitions, then there are only finitely many formulas of the latter form. □

Recall that an assignment in a model M is a homomorphism from the formula algebra into M.

Lemma 3. *Let T be a nonempty, finite set of formulas. Then, $M(\overline{T},\Phi)$ is a finite distributive lattice-ordered residuated groupoid. For any assignment f in $M(\overline{T},\Phi)$ such that $f(p) = [p]$, for any $p \in \overline{T}$, and any $\overline{T}-$sequent $X \Rightarrow \alpha$, f satisfies $X \Rightarrow \alpha$ in $M(\overline{T},\Phi)$ if and only if $X \Rightarrow_{\overline{T}} \alpha$.*

Proof. As shown above, $M(\overline{T},\Phi)$ is a lattice-ordered residuated groupoid. We prove the second part of the lemma. Let f satisfy $f(p) = [p]$, for any variable $p \in \overline{T}$. Using (7), (8), one proves $f(\alpha) = [\alpha]$, for all $\alpha \in \overline{T}$, by easy formula induction.

Assume that f satisfies the $\overline{T}-$sequent $X \Rightarrow \alpha$. For any formula β appearing in X, we have $\beta \in [\beta] = f(\beta)$, whence $X \in f(F(X))$. Since $f(F(X)) \subseteq f(\alpha)$, then $X \in f(\alpha) = [\alpha]$. Thus $X \Rightarrow_{\overline{T}} \alpha$. Assume $X \Rightarrow_{\overline{T}} \alpha$. Then, there exists a $\overline{T}-$deduction of $X \Rightarrow \alpha$ from Φ in **DFNL**. By induction on this deduction, we prove that f satisfies $X \Rightarrow \alpha$ in $M(\overline{T}, \Phi)$. f obviously satisfies axioms (Id). Assumptions from Φ and instances of (D), restricted to $\overline{T}-$sequents, are of the form $\beta \Rightarrow \gamma$, where $\beta, \gamma \in \overline{T}$. Since $\beta \Rightarrow_{\overline{T}} \gamma$, then $[\beta] \subseteq [\gamma]$, by (CUT), which yields $f(\beta) \subseteq f(\gamma)$. The rules of **FNL** are sound for any assignment in a lattice-ordered residuated groupoid, which finishes this part of proof.

Let R be a selector of the family of equivalence classes of $\overline{T}-$equivalence (R chooses one formula from each equivalence class). By Lemma 2, R is a nonempty finite subset of $\overline{T}$. We show that every nontrivial (i.e. nonempty and not total) closed subset of $\overline{T}^*$ equals $[\alpha]$, for some $\alpha \in R$. Let U be nontrivial and closed.. Let $X \in U$. There exists a set $[Z[\circ], \beta] \in B(\overline{T})$ such that $U \subseteq [Z[\circ], \beta]$. So, $Z[X] \Rightarrow_{\overline{T}} \beta$. By Lemma 1, there exists $\delta \in \overline{T}$ such that $Z[\delta] \Rightarrow_{\overline{T}} \beta$ and $X \Rightarrow_{\overline{T}} \delta$. We get $X \in [\delta]$ and $[\delta] \subseteq [Z[\circ], \beta]$, by (CUT). Clearly we can take $\delta \in R$. We can find such a formula $\delta \in R$, for any set $[Z[\circ], \beta] \in B(\overline{T})$ such that $U \subseteq [Z[\circ], \beta]$. Let γ_X be the conjunction of all formulas δ, fulfilling the above. By (8), (C3) and (6), $X \in [\gamma_X]$ and $[\gamma_X] \subseteq U$. We can replace γ_X by a $\overline{T}-$equivalent formula from R. So, we stipulate $\gamma_X \in R$. Let α be the disjunction of all formulas γ_X, for $X \in U$. By (8), $[\alpha] \subseteq U$ and, evidently, $U \subseteq [\alpha]$. We can assume $\alpha \in R$.

It follows that $M(\overline{T}, \Phi)$ is finite. We prove that it is distributive. It suffices to prove $U \wedge (V \vee W) \subseteq (U \wedge V) \vee (U \wedge W)$, for any closed sets U, V, W. This inclusion is true, if at least one of the sets U, V, W is empty or total, since $M(\overline{T}, \Phi)$ is a lattice. So, assume U, V, W be nontrivial. By the above paragraph, $U = [\alpha]$, $V = [\beta]$, $W = [\gamma]$, for some $\alpha, \beta, \gamma \in R$. Then, the inclusion follows from (8) and the fact that $[\alpha \wedge (\beta \vee \gamma)] \subseteq [(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)]$. □

We are ready to prove an extended subformula property and an interpolation lemma for **DFNL**.

Lemma 4. *Let T be a finite set of formulas, containing all formulas appearing in $X \Rightarrow \alpha$ and Φ. If $\Phi \vdash X \Rightarrow \alpha$ then $X \Rightarrow_{\Phi, \overline{T}} \alpha$.*

Proof. Let f be an asignment in $M(\overline{T}, \Phi)$, satisfying $f(p) = [p]$, for any variable $p \in \overline{T}$. Let $\beta \Rightarrow \gamma$ be a sequent from Φ. Then $\beta \Rightarrow_{\overline{T}} \gamma$, which yields $f(\beta) \subseteq f(\gamma)$, by Lemma 3. So, f satisfies all sequents from Φ.

Assume $\Phi \vdash X \Rightarrow \alpha$. Since **DFNL** is strongly sound with respect to distributive lattice-ordered residuated groupoids, then f satisfies $X \Rightarrow \alpha$. Consequently, $X \Rightarrow_{\overline{T}} \alpha$, by Lemma 3. □

Lemma 5. *Let T be a finite set of formulas, containing all formulas appearing in $X[Y] \Rightarrow \alpha$ and Φ. If $\Phi \vdash X[Y] \Rightarrow \alpha$ then there exists $\delta \in \overline{T}$ such that $\Phi \vdash X[\delta] \Rightarrow \alpha$ and $\Phi \vdash Y \Rightarrow \delta$.*

Proof. Assume $\Phi \vdash X[Y] \Rightarrow \alpha$. By Lemma 4, $X[Y] \Rightarrow_{\overline{T}} \alpha$. Apply Lemma 1. □

Lemma 3 except for the finiteness and distributivity of $M(\overline{T}, \Phi)$ and Lemmas 4 and 5 also hold for **FNL**. For **NL**, Lemma 4 and Lemma 5 hold with $\overline{T}$ defined as the closure of T under subformulas [9] (for pure **NL**, a weaker form of the latter lemma was proved in [15]).

This yields Strong Finite Model Property (SFMP) of **DFNL**: if $\Phi \vdash X \Rightarrow \alpha$ does not hold in **DFNL**, then there exist a finite distributive lattice-ordered residuated groupoid $\mathcal{M}$ and an assignment f such that all sequents from Φ are true under f, but $X \Rightarrow \alpha$ is not (take T as in Lemma 4 and use Lemma 3). Equivalently, every Horn formula which is not valid in distributive lattice-ordered residuated groupoids is falsified in a finite model. If a class of algebras $\mathcal{K}$ is closed under (finite) products, then SFMP for $\mathcal{K}$ is equivalent to FEP for $\mathcal{K}$: every finite partial subalgebra of an algebra from $\mathcal{K}$ is embeddable into a finite algebra from $\mathcal{K}$ [13]. (The latter is equivalent to FMP of the universal theory of $\mathcal{K}$.) Then, FEP holds for the class of distributive lattice-ordered residuated groupoids (proved in [11] in collaboration with the first author; the present proof is simplified).

4 Categorial Grammars Based on DFNL

A categorial grammar based on a logic $\mathcal{L}$ (presented as a sequent system) is defined as a tuple $G = (\Sigma, I, \alpha_0, \Phi)$ such that Σ is a finite alphabet, I is a nonempty finite relation between elements of Σ and formulas of $\mathcal{L}$, α_0 is a formula of $\mathcal{L}$, and Φ is a finite set of sequents of $\mathcal{L}$. Elements of Σ are usually interpreted as words from the lexicon of a language and strings on Σ as phrases. Formulas of $\mathcal{L}$ are called types. I assigns finitely many types to each word from Σ. If I is fixed, then one often writes $a \mapsto \beta$ for $(a, \beta) \in I$. α_0 is a designated type; often one takes a designated variable s (the type of sentences). $\mathcal{L}$ is the logic of type change and composition. Φ is a finite set of assumptions added to $\mathcal{L}$.

Our logic $\mathcal{L}$ is **DFNL**. Let $G = (\Sigma, I, \alpha_0, \Phi)$ be a categorial grammar. By $T(G)$ we denote the set of all types appearing in the range of I. Let T be the smallest set containing $T(G)$, all types from Φ and α_0. For any type β, we define $L(G, \beta) = \{X \in \overline{T}^* : \Phi \vdash X \Rightarrow \beta\}$. Elements of $\overline{T}^*$ can be seen as finite binary trees whose leaves are labeled by types from $\overline{T}$. The tree language of G, denoted by $L_t(G)$, consists of all trees which can be obtained from trees in $L(G, \alpha_0) \cap T(G)^*$ by replacing each type γ by some $a \in \Sigma$ such that $(a, \gamma) \in I$. The language of G, denoted by $L(G)$, is the yield of $L_t(G)$.

Theorem 1. *$L(G)$ is a context-free language, for any categorial grammar G based on* **DFNL**.

Proof. Fix $G = (\Sigma, I, \alpha_0, \Phi)$. We define a context-free grammar G' such that $L(G') = L(G)$. Let T be defined as above. By Lemma 2, $\overline{T}$ is finite up to $\overline{T}-$equivalence. We choose a set $R \subseteq \overline{T}$ which contains one formula from each equivalence class. For $\beta \in \overline{T}$, by $r(\beta)$ we denote the unique type from R which is $\overline{T}-$equivalent to β.

G' is defined as follows. The terminal alphabet is Σ. The nonterminal alphabet is R. Production rules are: (R1) $\beta \mapsto \gamma$, for $\beta, \gamma \in R$ such that $\Phi \vdash \gamma \Rightarrow \beta$, (R2) $\beta \mapsto \gamma\delta$, for $\beta, \gamma, \delta \in R$ such that $\Phi \vdash (\gamma, \delta) \Rightarrow \beta$, (R3) $r(\beta) \mapsto a$, for $\beta \in T(G)$, $a \in \Sigma$ such that $(a, \beta) \in I$. The initial symbol is $r(\alpha_0)$.

Every derivation tree in G' can be treated as a deduction from Φ in **DFNL** which is based on deducible sequents appearing in (R1), (R2) and (CUT). Then, $L(G') \subseteq L(G)$. The converse inclusion follows from Lemma 5. Let $x \in L(G)$. Then, x is the yield of some $Y \in L_t(G)$. There exists $X \in L(G, \alpha_0)$ such that Y is obtained from X in the way described above. Let $r(X)$ denote the tree resulting from X after one has replaced each type β by $r(\beta)$. Clearly, if $X \in L(G, \gamma)$, then $r(X) \in L(G, r(\gamma))$. It suffices to prove that, for any $\gamma \in \overline{T}$ and any $X \in L(G, \gamma)$, there exists a derivation of $r(X)$ from $r(\gamma)$ (as a derivation tree) in G'. We proceed by induction on the number of commas in X. Let $X \in L(G, \gamma)$ be a single type, say, $X = \beta$. Then, $\Phi \vdash \beta \Rightarrow \gamma$, whence $\Phi \vdash r(\beta) \Rightarrow r(\gamma)$. Then, $r(X) = r(\beta)$ is derivable from $r(\gamma)$, by (R1). Let $X \in L(G, \gamma)$ contain a comma. Then, X must contain a substructure of the form (δ_1, δ_2), where $\delta_i \in \overline{T}$. We write $X = Z[(\delta_1, \delta_2)]$. By Lemma 5, there exists $\delta \in \overline{T}$ such that $\Phi \vdash Z[\delta] \Rightarrow \gamma$ and $(\delta_1, \delta_2) \Rightarrow \delta$. By the induction hypothesis, $r(Z[\delta])$ can be derived from $r(\gamma)$ in G'. Then, $r(X)$ can be derived from $r(\gamma)$, by (R2). ☐

It has been shown in [7,17] that every $\epsilon-$free context-free language can be generated by a categorial grammar based on **NL** which uses very restricted types only: p, $p\backslash q$, $p\backslash(q\backslash r)$, where p, q, r are variables; the designated type is also a variable s. Now, we use the fact that **DFNL** is conservative over **NL**, since every residuated groupoid can be embedded into a powerset algebra over a groupoid [8]. Accordingly, every $\epsilon-$free context-free language is generated by some categorial grammar based on **DFNL**.

The tree language $L_t(G)$ is regular, for any grammar G based on **DFNL**. It follows from the above theorem (and its proof), since $L_t(G)$ equals the tree language determined by derivation trees of G'. A direct proof uses the well-known fact that a tree language $L \subseteq T^*$ is regular if and only if there exists a congruence of finite index on T^* which is compatible with L (it means: L is the join of some family of equivalence classes of $\sim$). For $X, Y \in T(G)^*$, define $X \sim Y$ iff, for all $\alpha \in \overline{T}$, $\Phi \vdash X \Rightarrow \alpha$ iff $\Phi \vdash Y \Rightarrow \alpha$. One can replace 'for all $\alpha \in \overline{T}$' by 'for all $\alpha \in R$', and consequently, there are at most 2^n equivalence classes of $\sim$, where n is the cardinality of R. So, $\sim$ is of finite index. Clearly $\sim$ is compatible with $L(G, \alpha_0)$. We prove that $\sim$ is a congruence on $T(G)^*$. Assume $X \sim Y$. We show $(Z, X) \sim (Z, Y)$, for any $Z \in T(G)^*$. Assume $\Phi \vdash (Z, X) \Rightarrow \alpha$, $\alpha \in \overline{T}$. By Lemma 5, there exists $\delta \in \overline{T}$ such that $\Phi \vdash (Z, \delta) \Rightarrow \alpha$ and $\Phi \vdash X \Rightarrow \delta$. Then, $\Phi \vdash Y \Rightarrow \delta$, whence $\Phi \vdash (Z, Y) \Rightarrow \alpha$, by (CUT). We have shown that $\Phi \vdash (Z, X) \Rightarrow \alpha$ implies $\Phi \vdash (Z, Y) \Rightarrow \alpha$. The converse implication has a similar proof. Therefore $(Z, X) \sim (Z, Y)$. Similarly $(X, Z) \Rightarrow (Y, Z)$. Consequently $L(G, \alpha_0) \cap T(G)^*$ is regular. By the construction of $L_t(G)$, the latter tree language is also regular.

The proof of Theorem 1 provides a construction of a context-free grammar equivalent to a given categorial grammar based on **DFNL**, since $\vdash$ for **DFNL** is decidable (it follows from FEP), and R can effectively be constructed. Actually we can construct a set R' such that $R \subseteq R'$. For T, defined as above, let T' denote the set of all subformulas of formulas from T. Then, $\overline{T}$ is the closure of T' under $\wedge, \vee$. Let $\alpha_1, \ldots, \alpha_k$ be all formulas from T'. We form all finite conjunctions $\alpha_{i_1} \wedge \cdots \wedge \alpha_{i_l}$ such that $1 \leq i_1 < \cdots < i_l \leq k$ and arrange them in a list $\beta_1, \ldots, \beta_{2^k-1}$. We define R' as the set of all finite disjunctions $\beta_{i_1} \vee \cdots \vee \beta_{i_l}$ such that $1 \leq i_1 < \cdots < i_l \leq 2^k - 1$. Clearly every formula from $\overline{T}$ is equivalent to some formula from R'. Using a decision method for **DFNL**, one can extract R from R'.

A syntactic decision method for **DFNL** can be designed like the one for **NL** in [9]. One can assume that R' is closed under $\wedge, \vee$ up to $\overline{T}-$equivalence. More precisely, for $\alpha, \beta \in R'$, one defines $\alpha \wedge' \beta$ as the formula from R' which results from a natural reduction of $\alpha \wedge \beta$ to a formula from R'. $\alpha \vee' \beta$ is defined in a similar way. Sequents of the form $\alpha \Rightarrow \beta$ and $(\alpha, \beta) \Rightarrow \gamma$ such that $\alpha, \beta, \gamma \in R'$ are called basic sequents. As in [9], one proves that a basic sequent is deducible from Φ if and only if there exists a deduction of this sequent from Φ which consists of basic sequents only (in rules for $\wedge, \vee$, these connectives are replaced by $\wedge'$ and $\vee'$, respectively). This yields a doubly exponential decision method for **DFNL** and a construction of G' of the same complexity. (In [9], it yields a polynomial time decision procedure for $\vdash$ in **NL**.)

Analogous results can be obtained for the $\vee-$free fragment of **FNL** (with assumptions). In the proof of Lemma 3, $\vee$ cannot be used; the finiteness of the model follows from the fact that every closed set is the join of a family of equivalence classes of $\sim$ (see above). The construction of G' requires exponential time.

5 Variants

The methods of this paper cannot be applied to associative systems **L**, **FL**, **DFL**. Consequence relations for these systems are undecidable; see [9,13]. Analogues of Theorem 1 are false (see Introduction).

They can be applied to several other nonassociative systems. The first example is $\mathbf{DFNL}_e$, i.e. **DFNL** with the exchange rule:

$$\text{(EXC)}\ \frac{X[(Y,Z)] \Rightarrow \alpha}{X[(Z,Y)] \Rightarrow \alpha}.$$

$\mathbf{DFNL}_e$ is complete with respect to distributive lattice-ordered commutative residuated groupoids ($ab = ba$, for all elements a, b). Then, $a \backslash b = b/a$, and one considers one residual only, denoted $a \rightarrow b$. All results from sections 2, 3 and 4 can be proved for $\mathbf{DFNL}_e$, and proofs are similar as above. Exception: not every $\epsilon-$free context-free language can be generated by a categorial grammar based on $\mathbf{DFNL}_e$.

One can add the multiplicative constant 1, interpreted as the unit in unital groupoids. We need the axiom (1R): $\Rightarrow 1$, and the rules:

$$\text{(1Ll)}\ \frac{X[Y] \Rightarrow \alpha}{X[(1,Y)] \Rightarrow \alpha},\ \text{(1Lr)}\ \frac{X[Y] \Rightarrow \alpha}{X[(Y,1)] \Rightarrow \alpha}.$$

The empty antecedent is understood as the empty structure Λ, and one admits $(\Lambda, X) = X$, $(X, \Lambda) = X$ in metatheory. Again, there are no problems with adapting our results for **DFNL** with 1 and **DFNL**$_e$ with 1. Caution: $\overline{T}$ contains 1, for any set T, and T^* contains Λ.

Additive constants $\bot$ and $\top$ can also be added, with axioms:

$$(\bot\mathrm{L})\ X[\bot] \Rightarrow \alpha\,,\ (\top\mathrm{R})\ X \Rightarrow \top\,.$$

They are interpreted as the lower bound and the upper bound, respectively, of the lattice. $\overline{T}$ must contain these constants. In the proof of (an analogue of) Lemma 1, one must consider new cases: $X[Y] \Rightarrow \alpha$ is an axiom ($\bot$L) or ($\top$R). For the first case, if Y contains $\bot$ then $\bot$ is an interpolant of Y; otherwise, $\top$ is an interpolant of Y. For the second case, $\top$ is an interpolant of Y. In the proof of Lemma 3, $M(\overline{T}, \Phi)$ interprets $\bot$ as $C_{\overline{T}}(\emptyset)$ and $\top$ as $\overline{T}^*$. Now, every closed set equals α, for some $\alpha \in \overline{T}$.

Instead of one binary product $\cdot$ one may admit a finite number of operations $o, o', \ldots$ of arbitrary arity: unary, binary, ternary and so on. Each $n-$ary operation o is associated with n residual operations o^i, for $i = 1, \ldots, n$. In models, one assumes the (generalized) residuation law:

$$o(a_1, \ldots, a_n) \leq b \text{ iff } a_i \leq o^i(a_1, \ldots, b, \ldots, a_n)\,, \tag{9}$$

for all $i = 1, \ldots, n$ (on the right-hand side, b is the $i-$th argument of o^i). The corresponding formal system, called Generalized Lambek Calculus **GLC**, was presented in [8,9]. To each $n-$ary operation o one attributes a structure constructor $(X_1, \ldots, X_n)_o$, and formula structures can contain different structure constructors. Unary operations can be identified with (different) unary modalities. **GLC** represents a multi-modal variant of **NL**. The introduction rules for o and o^i look as follows:

$$\frac{X[(\alpha_1, \ldots, \alpha_n)_o] \Rightarrow \beta}{X[o(\alpha_1, \ldots, \alpha_n)] \Rightarrow \beta}\,,\ \frac{X_1 \Rightarrow \alpha_1; \ldots;\ X_n \Rightarrow \alpha_n}{(X_1, \ldots, X_n)_o \Rightarrow o(\alpha_1, \ldots, \alpha_n)}\,,$$

$$\frac{X[\alpha_i] \Rightarrow \beta;\ Y_j \Rightarrow \alpha_j,\ \text{for all } j \neq i}{X[(Y_1, \ldots, o^i(\alpha_1, \ldots, \alpha_n), \ldots, Y_n)_o] \Rightarrow \beta}\,,\ \frac{(\alpha_1, \ldots, X, \ldots, \alpha_n)_o \Rightarrow \alpha_i}{X \Rightarrow o^i(\alpha_1, \ldots, \alpha_n)}\,.$$

The consequence relation of **GLC** is polytime, and the corresponding categorial grammars generate $\epsilon-$free context-free languages [9]. All results of this paper can easily be adapted for **GLC** with $\wedge, \vee$ and distribution (and, possibly, (EXC) for some binary operations, multiplicative units for them, and $\bot, \top$). In powerset algebras, one defines $o(U_1, \ldots, U_n)$ as the set of all elements $o(a_1, \ldots, a_n)$ such that $a_i \in U_i$, for $i = 1, \ldots, n$; o admits n residual operations o^i, $i = 1, \ldots, n$: $o^i(U_1, \ldots, U_n) = \{a \in M : o(U_1, \ldots, \{a\}, \ldots, U_n) \subseteq U_i\}$. (C4) takes the form: $o(C(U_1), \ldots, C(U_n)) \subseteq C(o(U_1, \ldots, U_n))$. Each operation o in the powerset algebra induces an operation on closed sets: $o_C(U_1, \ldots, U_n) = C(o(U_1, \ldots, U_n))$.

At the end, we consider **DFNL** with $\bot, \top$ and negation $\neg$, satisfying the laws of boolean algebra. To axioms for $\bot$ and $\top$ we add:

$$(\neg 1)\ \alpha \wedge \neg\alpha \Rightarrow \bot\,,\ (\neg 2)\ \top \Rightarrow \alpha \vee \neg\alpha\,.$$

The resulting system, denoted **BFNL**, is strongly complete with respect to lattice-ordered residuated groupoids whose underlying lattice is a boolean algebra.

All results of this paper can be proved for **BFNL**. We consider an auxiliary system **S** which is **DFNL** with $\top, \bot$ and an additional product connective & with residual $\rightarrow$ (so, it is a **GLC**-like system). We assume (EXC) for the structure constructor of &, whence & is commutative. We define $\neg\alpha = \alpha \rightarrow \bot$ and admit axioms ($\neg$1), ($\neg$2). It is easy to show that **S** is a conservative extension of **BFNL** (every model of the latter can be expanded to a model of the former, by identifying & with $\wedge$ and $\rightarrow$ with boolean implication). Now, for **S** we proceed as for **GLC** with additives. $\overline{T}$ must contain $\top, \bot$ and be closed under $\wedge, \vee, \neg$; then, Lemma 2 remains true, since boolean algebras are locally finite.

If we replace in **S** axioms ($\neg$1), ($\neg$2) by axioms $\alpha \wedge \beta \Rightarrow \alpha \& \beta$ and $\alpha \& \beta \Rightarrow \alpha \wedge \beta$, then we obtain **HFNL**: a complete logic of Heyting-ordered residuated groupoids. Now, $\overline{T}$ must be closed under $\wedge, \vee, \&$, but Lemma 2 is still true. All results of this paper can be proved for **HFNL** as well.

Theorem 2. *Categorial grammars based on* **BFNL**, **HFNL** *generate context-free languages. FEP holds for the classes of boolean-ordered and Heyting-ordered residuated groupoids (algebras).*

References

1. Avron, A.: The Semantics and Proof Theory of Linear Logic. Theoretical Computer Science 57, 161–184 (1988)
2. Belardinelli, F., Jipsen, P., Ono, H.: Algebraic Aspects of Cut Elimination. Studia Logica 77, 209–240 (2004)
3. van Benthem, J.: Language in Action: Categories, Lambdas and Dynamic Logic. North-Holland, Amsterdam (1991)
4. van Benthem, J., ter Meulen, A. (eds.): Handbook of Logic and Language. Elsevier, Amsterdam (1997)
5. Blok, W.J., van Alten, C.J.: On the finite embeddability property for residuated ordered groupoids. Transactions of AMS 357(10), 4141–4157 (2005)
6. Buszkowski, W.: A note on the Lambek - van Benthem calculus. Bulletin of The Section of Logic 131, 31–37 (1984)
7. Buszkowski, W.: Generative Capacity of Nonassociative Lambek Calculus. Bull. Polish Acad. Scie. Math. 34, 507–516 (1986)
8. Buszkowski, W.: Mathematical Linguistics and Proof Theory. In: [4], pp. 683–736
9. Buszkowski, W.: Lambek Calculus with Nonlogical Axioms. In: Casadio, C., Scott, P.J., Seely, R. (eds.) Language and Grammar. Studies in Mathematical Linguistics and Natural Language, pp. 77–93. CSLI Publications, Stanford (2005)
10. Buszkowski, W.: On Action Logic: Equational Theories of Action Algebras. Journal of Logic and Computation 17(1), 199–217 (2007)
11. Farulewski, M.: Finite Embeddabilty Property for Residuated Groupoids. Reports on Mathematical Logic 43, 25–42 (2008)
12. Francez, N., Kaminski, M.: Commutation-Augmented Pregroup Grammars and Mildly Context-Sensitive Languages. Studia Logica 87(2-3), 295–321 (2007)

13. Galatos, N., Jipsen, P., Kowalski, T., Ono, H.: Residuated Lattices: An Algebraic Glimpse at Substructural Logics. Elsevier, Amsterdam (2007)
14. de Groote, P., Lamarche, F.: Classical Nonassociative Lambek Calculus. Studia Logica 71(2), 355–388 (2002)
15. Jäger, G.: Residuation, structural rules and context-freeness. Journal of Logic, Language and Information 13, 47–59 (2004)
16. Kanazawa, M.: The Lambek Calculus Enriched with Additional Connectives. Journal of Logic, Language and Information 1(2), 141–171 (1992)
17. Kandulski, M.: The equivalence of nonassociative Lambek categorial grammars and context-free grammars. Zeitschrift für mathematische Logik und Grundlagen der Mathematik 34, 41–52 (1988)
18. Lambek, J.: The mathematics of sentence structure. American Mathematical Monthly 65, 154–170 (1958)
19. Lambek, J.: On the calculus of syntactic types. In: Jakobson, R. (ed.) Structure of Language and Its Mathematical Aspects, pp. 166–178. AMS, Providence (1961)
20. Lambek, J.: Type Grammars Revisited. In: Lecomte, A., Perrier, G., Lamarche, F. (eds.) LACL 1997. LNCS (LNAI), vol. 1582, pp. 1–27. Springer, Heidelberg (1999)
21. Moortgat, M.: Categorial Type Logic. In: [4], pp. 93–177
22. Morrill, G.: Type Logical Grammar. Categorial Logic of Signs. Kluwer, Dordrecht (1994)
23. Okada, M., Terui, K.: The finite model property for various fragments of intuitionistic linear logic. Journal of Symbolic Logic 64, 790–802 (1999)
24. Pentus, M.: Lambek Grammars are Context-Free. In: Proc. 8th IEEE Symp. Logic in Computer Sci., pp. 429–433 (1993)
25. Pentus, M.: Lambek calculus is NP-complete. Theoretical Computer Science 357, 186–201 (2006)