Galois Connections in Categorial Type Logic

Carlos Areces [a,1] [,2]   Raffaella Bernardi [b,3]
Michael Moortgat [b,4]

[a] *ILLC, University of Amsterdam, Amsterdam, The Netherlands*
[b] *UiL-OTS, University of Utrecht, Utrecht, The Netherlands*

Abstract

The introduction of unary connectives has proved to be an important addition to the categorial vocabulary. The connectives considered so far are order-preserving; in this paper instead, we consider the addition of order-reversing, Galois connected operators. In §2 we do the basic model-theoretic and proof-theoretic groundwork. In §3 we use the expressive power of the Galois connected operators to restrict the scopal possibilities of generalized quantifier expressions, and to describe a typology of polarity items.

1   Introduction

Categorial type logic provides a vocabulary of logical constants for the assembly of form and meaning in natural language. The binary product operator $\bullet$ captures the composition of grammatical parts, the residual implications $/$ and $\backslash$ express incompleteness with respect to the composition relation. In [11,18] the categorial vocabulary has been extended with a pair of unary operators, $\diamondsuit$ and its residual $\square^{\downarrow}$. This addition greatly increases the analytical power of the categorial type language. In combination with the binary connectives, the unary operators can be used as licensing features, providing lexically anchored control over structural reasoning. But already in the grammatical base logic $\mathsf{NL}(\diamondsuit)$, the unary con-

stants yield refinements of type assignments that would be over-generating without the $\Diamond, \Box^{\downarrow}$ decoration.

The $\Diamond, \Box^{\downarrow}$ connectives form a *residuated pair*, which means they are order-preserving operators with respect to the derivability relation. The algebraic structure of the base logic also provides room for a pair of order-reversing *Galois connected* operators, which in this paper we will write as $^{\mathbf{0}}\cdot, \cdot^{\mathbf{0}}$. To understand the relation between these two concepts, it may be useful to situate them in their natural algebraic context. Residuated and Galois connected pairs of mappings were studied in the work of Birkhoff [5] and Ore [20], among others. The relevance of this early work for current research on substructural logics has been emphasized by Michael Dunn, from whose [7] we draw the following definitions.

Definition 1.1 Consider two posets $\mathcal{A} = (A, \leq_A)$ and $\mathcal{B} = (B, \leq_B)$, and functions $f : A \to B, \; g : B \to A$. The pair (f, g) is said to be *residuated* iff

$$[\text{RES}] \qquad fa \leq_B b \;\; \text{iff} \;\; a \leq_A gb.$$

The pair (f, g) is said to be *Galois connected* iff

$$[\text{GC}] \qquad b \leq_B fa \;\; \text{iff} \;\; a \leq_A gb.$$

Keeping the posets $\mathcal{A}$ and $\mathcal{B}$ distinct helps understanding the connection between residuated and Galois connected pairs of mappings. Let's introduce a third poset $\mathcal{B}' = (B, (\leq_B)^{-1})$ where $(\leq_B)^{-1} = \{(b, a) \mid a \leq_B b\}$, and consider a function $h : B \to A$. Following [RES] the functions f, h form a residuated pair iff the biconditional $fa \; (\leq_B)^{-1} \; b \Leftrightarrow a \leq_A hb$ holds. But now, replacing $(\leq_B)^{-1}$ by $\leq_B$, we obtain that $b \leq_B fa \Leftrightarrow a \leq_B hb$, i.e. the pair f, h is Galois connected with respect to the orders $\leq_B$ and $\leq_A$. As Dunn [7] puts it, the Galois connected pair is obtained by turning around the inequality $\leq_B$ in the characterization of residuation.

When we cast this algebraic discussion in terms of categorial type logics the objects we will be considering are types, ordered by their derivability relation. Galois connected operators have been also studied in the context of Linear Logic [12,1,9,21] where they are intended to exhibit negation-like behavior. This means that the Galois properties have to be mixed with extra features guaranteeing, for example, a double negation law $^{\mathbf{0}}(A^{\mathbf{0}}) = A = (^{\mathbf{0}}A)^{\mathbf{0}}$. In related work, Jim Lambek [13,14] considers algebraic structures he calls *pregroups*, where each element a has a left and a right *adjoint*, written a^l and a^r. Also in these structures, one has $a^{lr} = a = a^{rl}$. In this paper, we do not consider these stronger notions, but we concentrate on the pure Galois properties and investigate the effect of adding $^{\mathbf{0}}\cdot, \cdot^{\mathbf{0}}$ to the base logic $\mathsf{NL}(\Diamond)$. We are interested in the base logic because we think it opens a window on the *invariants* of grammatical composition — the laws of the base logic are universals in the sense that they do not depend on structural postulates (that is, non-logical axioms).

The paper is organized as follows. In §2 we provide the basic model-theoretic and proof-theoretic groundwork. We show how the completeness result for the standard Kripke-style semantics for $\mathsf{NL}(\Diamond)$ can be extended to the Galois connected operators, and how a cut-free Gentzen style presentation can be obtained. In §3, we turn to Beghelli and Stowell's analysis of ways of scope taking [3] and to Giannakidou's theory of polarity items distribution [8], to provide linguistic motivation for the Galois connected operators. We exploit the new derivability patterns introduced by the Galois operators to constrain the scopal possibilities of generalized quantifier expressions and describe a typology of polarity items. In this way we improve on the analysis of [4], which was given in terms of the residuated operators $\Diamond, \Box^{\downarrow}$.

2 Formal preliminaries

2.1 Axiomatic presentation, completeness

There are two ways to extend the standard axiomatic presentation of $\mathsf{NL}(\Diamond)$ (see [19]) with Galois operators. The system $\mathsf{NL}(\Diamond, \cdot^{\mathbf{0}})$ can be obtained by extending $\mathsf{NL}(\Diamond)$ with the axioms $(A1)$, $(A2)$ and the rules $(R1)$, $(R2)$ below. It is easy to show that (GC) is a derived rule in this setting. Alternatively, one adds (GC) to $\mathsf{NL}(\Diamond)$. It can be shown then that $(A1)$, $(A2)$ and the rules $(R1)$, $(R2)$ are derivable.

$$(A1) \ \vdash A \Rightarrow {}^{\mathbf{0}}(A^{\mathbf{0}}).$$
$$(A2) \ \vdash A \Rightarrow ({}^{\mathbf{0}}A)^{\mathbf{0}}.$$
$$(R1) \ \text{From} \vdash A \Rightarrow B \text{ infer} \vdash B^{\mathbf{0}} \Rightarrow A^{\mathbf{0}}.$$
$$(R2) \ \text{From} \vdash A \Rightarrow B \text{ infer} \vdash {}^{\mathbf{0}}B \Rightarrow {}^{\mathbf{0}}A.$$

$$(\text{GC}) \ \vdash A \Rightarrow {}^{\mathbf{0}}B \text{ if and only if} \vdash B \Rightarrow A^{\mathbf{0}}.$$

$\mathsf{NL}(\Diamond)$ has a Kripke-style semantics [10] which can be straightforwardly extended to $\mathsf{NL}(\Diamond, \cdot^{\mathbf{0}})$. A model for $\mathsf{NL}(\Diamond, \cdot^{\mathbf{0}})$ is a tuple $\mathcal{M} = \langle W, R^3, R_1^2, R_2^2, V \rangle$ where W is a non-empty set, $R^3 \subseteq W^3$, $R_i^2 \subseteq W^2$, and V is a valuation $V : \mathsf{PROP} \to 2^W$. The R^3 relation governs the residuated triple $\bullet, \backslash, /$, the R_1^2 relation governs the residuated pair $\Diamond, \Box^{\downarrow}$, while R_2^2 governs the Galois connected pair $\cdot^{\mathbf{0}}, {}^{\mathbf{0}}\cdot$. For simplicity, in what follows we will restrict ourselves to models $\mathcal{M} = \langle W, R, V \rangle$ where R is the relation governing the Galois operators.

Given a model $\mathcal{M} = \langle W, R, V \rangle$ and $m \in W$ we define

$$\mathcal{M}, m \Vdash A^{\mathbf{0}} \ \text{iff} \ \forall m'.(Rmm' \Rightarrow \mathcal{M}, m' \nVdash A).$$
$$\mathcal{M}, m \Vdash {}^{\mathbf{0}}A \ \text{iff} \ \forall m'.(Rm'm \Rightarrow \mathcal{M}, m' \nVdash A).$$

Given an arrow $A \Rightarrow B$, a model $\mathcal{M} = \langle W, R, V \rangle$ and $m \in W$, we say that $\mathcal{M}, m \models A \Rightarrow B$ iff $\mathcal{M}, m \Vdash A$ implies $\mathcal{M}, m \Vdash B$. $\mathcal{M} \models A \Rightarrow B$ iff for all $m \in W$, $\mathcal{M}, m \Vdash A \Rightarrow B$. We say that $A \Rightarrow B$ is *valid* (notation $\models A \Rightarrow B$), iff for any model $\mathcal{M}$, $\mathcal{M} \models A \Rightarrow B$.

It is easy to show that the axioms $(A1)$ and $(A2)$ are true in all Kripke models, and that the rules $(R1)$ and $(R2)$ preserve validity, establishing soundness. For completeness, we can extend the formula-based canonical construction for $\mathsf{NL}(\Diamond)$ (cf. [6,10]). The *canonical model* $\mathcal{M}^c = \langle W^c, R^c, V^c \rangle$ has

$$
\begin{aligned}
W^c &= \mathsf{FORM} \text{ (the set of all formulas in the language)}, \\
\neg R^c AB &\text{ iff } \vdash A \Rightarrow B^{\mathbf{0}}, \text{ and} \\
A \in V^c(p) &\text{ iff } \vdash A \Rightarrow p.
\end{aligned}
$$

Notice that we define when two elements of W are *not* related by R. This, of course, defines also which elements are related. But we can do even better than $\mathcal{M}^c$. Given an arrow $A \Rightarrow B$, we can restrict W^c to be simply $W^c = \mathsf{Sub}(A) \cup \mathsf{Sub}(B)$ (the set of subformulas of A and B) and prove the following truth lemma.

Lemma 2.1 (Truth Lemma) *Given $A \Rightarrow B$, then for all $A', B' \in \mathsf{Sub}(A) \cup \mathsf{Sub}(B)$ $\mathcal{M}^c, A' \Vdash B'$ iff $\vdash A' \Rightarrow B'$.*

With this lemma, we can prove completeness with respect to a class of finite models, and hence obtain also decidability (actually, even an upper bound on complexity).

Proof. The proof proceeds by induction on the complexity of the consequent formula. For $B \in \mathsf{PROP}$, $\mathcal{M}^c, A \Vdash B$ iff $A \in V^c(B)$ iff, by definition of V^c, $\vdash A \Rightarrow B$. We assume as induction hypothesis (IH) that the lemma is true for formulas of lower or equal complexity than B.

We consider $^{\mathbf{0}}B$ (the case for $B^{\mathbf{0}}$ being even simpler).

$[\Rightarrow]$ direction. $\mathcal{M}^c, A \Vdash {}^{\mathbf{0}}B$ iff for all $B' \in W^c$ if $R^c B' A$ then $\mathcal{M}^c, B' \nVdash B$. By contraposition and definition of R^c, for all B', $\mathcal{M}^c, B' \Vdash B$ implies $\vdash B' \Rightarrow A^{\mathbf{0}}$. By definition of W^c, B' is in $\mathsf{Sub}(A) \cup \mathsf{Sub}(B)$ and we can apply IH to obtain that for all $B' \in W^c$, $\vdash B' \Rightarrow B$ implies $\vdash B' \Rightarrow A^{\mathbf{0}}$. In particular, $B \in W^c$ and by (REFL) $\vdash B \Rightarrow B$, hence $\vdash B \Rightarrow A^{\mathbf{0}}$. By (GC), $\vdash A \Rightarrow {}^{\mathbf{0}}B$.

$[\Leftarrow]$ direction. Assume $\vdash A \Rightarrow {}^{\mathbf{0}}B$ to prove $\mathcal{M}^c, A \Vdash {}^{\mathbf{0}}B$. Take B' such that $R^c B' A$, we should prove $\mathcal{M}^c, B' \nVdash B$. Notice that by definition of R^c, we have that $\nvdash B' \Rightarrow A^{\mathbf{0}}$. For contradiction, suppose $\mathcal{M}^c, B' \Vdash B$, then by IH, $\vdash B' \Rightarrow B$, but then we can prove $\vdash B' \Rightarrow A^{\mathbf{0}}$ as follows

$$\frac{\vdash A \Rightarrow {}^{\mathbf{0}}B \quad \dfrac{\vdash B' \Rightarrow B}{\vdash {}^{\mathbf{0}}B \Rightarrow {}^{\mathbf{0}}B'}\ (R2)}{\dfrac{\vdash A \Rightarrow {}^{\mathbf{0}}B'}{\vdash B' \Rightarrow A^{\mathbf{0}}}\ (\text{GC})}\ (\text{TRANS}) \tag{1}$$

$\square$

Theorem 2.2 (Completeness) *Given* $A \Rightarrow B$*, then* $\models A \Rightarrow B$ *implies* $\vdash A \Rightarrow B$.

Proof. Suppose $\nvdash A \Rightarrow B$. Then by Lemma 2.1 $\mathcal{M}^c, A \nVdash B$. As $\mathcal{M}^c, A \Vdash A$, we have $\mathcal{M}^c \not\models A \Rightarrow B$ and hence $\not\models A \Rightarrow B$. $\square$

As we already said, Lemma 2.1 actually establishes a strong finite model property (an arrow $A \Rightarrow B$ is valid iff B is satisfied in $\mathcal{M}^c, A$, a (pointed) model whose size is polynomial in $|A| \cup |B|$). From this, an NP upper bound in the complexity of the validity problem for $\mathsf{NL}(\diamondsuit, \cdot^{\mathbf{0}})$ follows.

Theorem 2.3 *Given* $A \Rightarrow B \in \mathsf{NL}(\diamondsuit, \cdot^{\mathbf{0}})$*, deciding whether* $A \Rightarrow B$ *is valid can be done in non-deterministic polynomial time.*

2.2 Gentzen presentation, cut-elimination

In this section, we extend the Gentzen presentation of $\mathsf{NL}(\diamondsuit)$ of [18] to $\mathsf{NL}(\diamondsuit, \cdot^{\mathbf{0}})$. We show that the Cut rule can be eliminated, yielding decidable proof search. Our Gentzen presentation has sequents $\Gamma \Rightarrow \Delta$, with $\Gamma, \Delta \in$ STRUC. Structures are defined as STRUC ::= FORM | $\flat$STRUC | $\sharp$STRUC, with structural connectives $\flat$ and $\sharp$ matching the logical connectives $^{\mathbf{0}}\cdot$ and $\cdot^{\mathbf{0}}$, respectively.

In [9,21], the proof theory for the Galois connected operators is given in the framework of Display Calculus. [2] shows how to move from display calculus to a Gentzen calculus for (unary and binary) residuated and Galois connected operators. In display calculus, $\flat$ and $\sharp$ are related by (DGC) (the structural counterpart of (GC)) $\Gamma \Rightarrow \flat\Delta$ iff $\Delta \Rightarrow \sharp\Gamma$, which makes it possible to give the following logical rules for the connectives:

$$\frac{\Delta \Rightarrow A}{{}^{\mathbf{0}}A \Rightarrow \flat\Delta}\ (D1) \quad \frac{\Delta \Rightarrow A}{A^{\mathbf{0}} \Rightarrow \sharp\Delta}\ (D2) \quad \frac{\Delta \Rightarrow \flat A}{\Delta \Rightarrow {}^{\mathbf{0}}A}\ (D3) \quad \frac{\Delta \Rightarrow \sharp A}{\Delta \Rightarrow A^{\mathbf{0}}}\ (D4) \tag{2}$$

In the Gentzen presentation, we want to compile away (DGC) (and part of the Cut rule) in the logical rules. Because the Galois operators are order-reversing, we have to distinguish positive and negative contexts in the statement of the Cut rule. We write $\Gamma[\Delta]$ for a structure Γ with a substructure Δ in an isotone position (dominated by an even number of structural connectives), and $\Gamma\{\Delta\}$ for a structure Γ with Δ in an antitone position

5

(dominated by an odd number of structural connectives). In (3) we give the four instances of the Cut rule we have to consider.

$$\frac{\Delta \Rightarrow A \quad \Gamma[A] \Rightarrow \Delta'}{\Gamma[\Delta] \Rightarrow \Delta'} \ (C1) \qquad \frac{\Delta' \Rightarrow \Gamma[A] \quad A \Rightarrow \Delta}{\Delta' \Rightarrow \Gamma[\Delta]} \ (C2)$$

(3)

$$\frac{\Delta \Rightarrow A \quad \Delta' \Rightarrow \Gamma\{A\}}{\Delta' \Rightarrow \Gamma\{\Delta\}} \ (C3) \qquad \frac{\Gamma\{A\} \Rightarrow \Delta' \quad A \Rightarrow \Delta}{\Gamma\{\Delta\} \Rightarrow \Delta'} \ (C4)$$

The left introduction logical rules below are as in the display calculus presentation, the right introduction rules compile in an application of (DGC).

$$\frac{\Delta \Rightarrow A}{{}^{\mathbf{0}}A \Rightarrow \flat\Delta} \ ({}^{\mathbf{0}}\!\cdot\! L) \quad \frac{\Delta \Rightarrow A}{A^{\mathbf{0}} \Rightarrow \sharp\Delta} \ (\cdot^{\mathbf{0}}L) \quad \frac{A \Rightarrow \sharp\Delta}{\Delta \Rightarrow {}^{\mathbf{0}}A} \ ({}^{\mathbf{0}}\!\cdot\! R) \quad \frac{A \Rightarrow \flat\Delta}{\Delta \Rightarrow A^{\mathbf{0}}} \ (\cdot^{\mathbf{0}}R) \qquad (4)$$

The logical rules in (4) swap around antecedent and succedent of a sequent. For cut elimination to go through, we also need contextual versions of the rules, compiling in the axiom schemata $(A1)/(A2)$ with $(C2)/(C4)$, and the complementary rules (L^-), (R^+) replacing the logical connectives by their respective structural counterparts.

$$\frac{\Gamma\{A\} \Rightarrow \Delta}{\Gamma\{\sharp^{\mathbf{0}}A\} \Rightarrow \Delta} \ ({}^{\mathbf{0}}\!\cdot\! L^+) \qquad \frac{\Gamma\{A\} \Rightarrow \Delta}{\Gamma\{\flat A^{\mathbf{0}}\} \Rightarrow \Delta} \ (\cdot^{\mathbf{0}}L^+) \qquad (5)$$

$$\frac{\Delta \Rightarrow \Gamma[A]}{\Delta \Rightarrow \Gamma[\sharp^{\mathbf{0}}A]} \ ({}^{\mathbf{0}}\!\cdot\! R^-) \qquad \frac{\Delta \Rightarrow \Gamma[A]}{\Delta \Rightarrow \Gamma[\flat A^{\mathbf{0}}]} \ (\cdot^{\mathbf{0}}R^-) \qquad (6)$$

We will call Seq-NL($\diamondsuit$,$\cdot^{\mathbf{0}}$) the Gentzen presentation of NL($\diamondsuit$,$\cdot^{\mathbf{0}}$), to distinguish it from its Hilbert presentation Hil-NL($\diamondsuit$,$\cdot^{\mathbf{0}}$).

Theorem 2.4 (Cut elimination) *In Seq-NL($\diamondsuit$,$\cdot^{\mathbf{0}}$), every valid sequent $A \Rightarrow B$ has a cut-free proof.*

The proof proceeds by induction on the complexity of the Cut inferences. Below, we present the principal cases of the cut elimination transformation: the cases where a cut on a complex cut formula is replaced by a cut on its subformula, thus decreasing the complexity. The other cases follow the same ideas.

In (7) and (8), isotone cuts $(C1, C3)$ on the complex formula $A^{\mathbf{0}}$ are replaced by antitone cuts $(C4, C2)$. Similarly for cuts on ${}^{\mathbf{0}}A$. (We use double lines for the instantiation of the premise that makes a logical rule applicable.)

6

$$\dfrac{\dfrac{A \Rightarrow \flat\Delta}{\Delta \Rightarrow A^{\mathbf{0}}}\ (\cdot^{\mathbf{0}}R) \quad \dfrac{\dfrac{\Gamma\{A\} \Rightarrow \Delta'}{\Gamma\{\flat(A^{\mathbf{0}})\} \Rightarrow \Delta'}\ (\cdot^{\mathbf{0}}L^{+})}{\Gamma[A^{\mathbf{0}}] \Rightarrow \Delta'}\ (C1)}{\dfrac{\Gamma[\Delta] \Rightarrow \Delta'}{\Gamma\{\flat\Delta\} \Rightarrow \Delta'}} \quad\rightsquigarrow\quad \dfrac{\Gamma\{A\} \Rightarrow \Delta' \quad A \Rightarrow \flat\Delta}{\Gamma\{\flat\Delta\} \Rightarrow \Delta'}\ (C4) \qquad (7)$$

$$\dfrac{\dfrac{A \Rightarrow \flat\Delta}{\Delta \Rightarrow A^{\mathbf{0}}}\ (\cdot^{\mathbf{0}}R) \quad \dfrac{\dfrac{\Delta' \Rightarrow \Gamma[A]}{\Delta' \Rightarrow \Gamma[\flat(A^{\mathbf{0}})]}\ (\cdot^{\mathbf{0}}R^{-})}{\Delta' \Rightarrow \Gamma\{A^{\mathbf{0}}\}}\ (C3)}{\dfrac{\Delta' \Rightarrow \Gamma\{\Delta\}}{\Delta' \Rightarrow \Gamma[\flat\Delta]}} \quad\rightsquigarrow\quad \dfrac{\Delta' \Rightarrow \Gamma[A] \quad A \Rightarrow \flat\Delta}{\Delta' \Rightarrow \Gamma[\flat\Delta]}\ (C2) \qquad (8)$$

In (9) and (10), antitone cuts ($C4, C2$) are replaced by isotone cuts ($C1, C3$).

$$\dfrac{\dfrac{\Gamma\{\sharp A\} \Rightarrow \Delta'}{\Gamma\{A^{\mathbf{0}}\} \Rightarrow \Delta'}\ (\cdot^{\mathbf{0}}L^{-}) \quad \dfrac{\Delta \Rightarrow A}{A^{\mathbf{0}} \Rightarrow \sharp\Delta}\ (\cdot^{\mathbf{0}}L)}{\Gamma\{\sharp\Delta\} \Rightarrow \Delta'}\ (C4) \quad\rightsquigarrow\quad \dfrac{\Delta \Rightarrow A \quad \Gamma\{\sharp A\} \Rightarrow \Delta'}{\Gamma\{\sharp\Delta\} \Rightarrow \Delta'}\ (C1) \qquad (9)$$

$$\dfrac{\dfrac{\Delta' \Rightarrow \Gamma[\sharp A]}{\Delta' \Rightarrow \Gamma[A^{\mathbf{0}}]}\ (\cdot^{\mathbf{0}}R^{+}) \quad \dfrac{\Delta \Rightarrow A}{A^{\mathbf{0}} \Rightarrow \sharp\Delta}\ (\cdot^{\mathbf{0}}L)}{\Delta' \Rightarrow \Gamma[\sharp\Delta]}\ (C2) \quad\rightsquigarrow\quad \dfrac{\Delta \Rightarrow A \quad \Delta' \Rightarrow \Gamma[\sharp A]}{\Delta' \Rightarrow \Gamma[\sharp\Delta]}\ (C3) \qquad (10)$$

2.3   Soundness and completeness of Seq-NL($\Diamond, \cdot^{\mathbf{0}}$)

We start by proving the following.

Proposition 2.5 *Let* $\mathcal{M} = \langle W, R, V \rangle$ *be a model, and* $m \in W$ *then*

(i) $\mathcal{M}, m \models \Delta \Rightarrow \Delta'[A]$ *and* $\models A \Rightarrow B$ *then* $\Delta \Rightarrow \Delta'[B]$.

(ii) $\mathcal{M}, m \models \Delta[A] \Rightarrow \Delta'$ *and* $\models B \Rightarrow A$ *then* $\Delta[B] \Rightarrow \Delta'$.

(iii) $\mathcal{M}, m \models \Delta \Rightarrow \Delta'\{A\}$ *and* $\models B \Rightarrow A$ *then* $\Delta \Rightarrow \Delta'\{B\}$.

(iv) $\mathcal{M}, m \models \Delta\{A\} \Rightarrow \Delta'$ *and* $\models A \Rightarrow B$ *then* $\Delta\{B\} \Rightarrow \Delta'$.

Proof. By induction on the number of operators surrounding A. $\square$

Now define the following forgetting function.

7

Definition 2.6 We define the following translation $Tr : \mathsf{STRUC} \to \mathsf{FORM}$ as follows,

$$Tr(p) = p \text{ for } p \in \mathsf{PROP}$$

$$Tr(\mathbf{^0}(A)) = \mathbf{^0}(Tr(A)) \quad Tr(\flat(A)) = \mathbf{^0}(Tr(A))$$

$$Tr((A)\mathbf{^0}) = (Tr(A))\mathbf{^0} \quad Tr(\sharp(A)) = (Tr(A))\mathbf{^0}.$$

Theorem 2.7 (Soundness of $\mathsf{Seq}\text{-}\mathsf{NL}(\diamondsuit,\cdot\mathbf{^0})$**)** *The sequent presentation of the logic $NL(\diamondsuit,\cdot\mathbf{^0})$ is sound.*

Proof. Given a rule

$$\frac{A \Rightarrow B}{C \Rightarrow D}$$

we prove that if $\models Tr(A) \Rightarrow Tr(B)$ then $\models Tr(C) \Rightarrow Tr(D)$, and similarly for rules with two premises.

Notice that Proposition 2.5 proves soundness of the Cut rules. For rules $(\mathbf{^0}\cdot R)$, $(\cdot\mathbf{^0}R)$, $(\mathbf{^0}\cdot L)$ and $(\cdot\mathbf{^0}L)$ use the fact that the (GC) rule is sound. For rules $(\mathbf{^0}\cdot R^-)$, $(\cdot\mathbf{^0}R^-)$, $(\mathbf{^0}\cdot L^+)$ and $(\cdot\mathbf{^0}L^+)$ use Proposition 2.5 plus the fact that axioms (A1) and (A2) are valid. $\qquad\square$

Theorem 2.8 (Equivalence of $\mathsf{Seq}\text{-}\mathsf{NL}(\diamondsuit,\cdot\mathbf{^0})$ **and** $\mathsf{Hil}\text{-}\mathsf{NL}(\diamondsuit,\cdot\mathbf{^0})$**)** *If $A \Rightarrow B$ is a theorem of Hil-NL$(\diamondsuit,\cdot\mathbf{^0})$ then there is a proof of $A \Rightarrow B$ in Seq-NL$(\diamondsuit,\cdot\mathbf{^0})$. And for every proof of a sequent $\Gamma \Rightarrow \Delta$ in Seq-NL$(\diamondsuit,\cdot\mathbf{^0})$, $\mathrm{Tr}(\Gamma) \Rightarrow \mathrm{Tr}(\Delta)$ is a theorem of Hil-NL$(\diamondsuit,\cdot\mathbf{^0})$.*

3 Linguistic application

Now that the logical properties of $\mathsf{NL}(\diamondsuit,\cdot\mathbf{^0})$ have been explained in detail, let us turn to its linguistic application. In order to explore the possible uses of the Galois unary operators it is important to look at the new properties they introduce. As we pointed out in the beginning, these new operators differ from the residuated ones in their monotonicity properties, and in their derivability relation with respect to their argument when considering their composition (viz. while A derives both $(^0A)^0$ and $^0(A^0)$, it derives $\Box^\downarrow \diamondsuit A$ but not $\diamondsuit \Box^\downarrow A$).

If we consider complex types as composition of functions, we realize that having at our disposal downward monotonic operators gives a way to modify the polarity position of their arguments. For example, if we consider the composition of $^0\cdot$ with the binary function $/$, which is positive in its first argument and negative in the other, we see that their monotonic properties are reversed. More specifically, if ^0A composes with $B/\cdot$ giving $B/^0(A)$, A will be in a positive position, and if it composes with $\cdot/B$, resulting in $^0(A)/B$, A will be in a negative position. Since the monotonicity properties of the connectives govern the derivability relation among

types, having added the Galois operators increases the connections between types which is the main feature of categorial type logic.

In the next part of this section we will show how to use the derivability patterns between the types of $NL(\diamond, \cdot^{\mathbf{0}})$ to account for the different distribution of generalized quantifiers with respect to negation and to describe a typology of polarity items.

3.1 Generalized Quantifiers Scope

We start with an example of how to account for scope phenomena using the composition of the Galois operators and the fact that $(^0 s)^0 \not\leftrightarrow \Box^\downarrow \diamond s$, where $A \longrightarrow B$ means that the formula is a tautology, viz. $A \longrightarrow B$ iff $\models A \Rightarrow B$.

A well known problem of general quantifiers expressions (GQs) in the categorial literature is the proper characterization of an *in situ* binding operation such as required for the scope possibilities of GQs. In [17] a "scoping constructor" operator $q(A, B, C)$ is introduced to capture this behavior. A GQ is assigned the type $q(np, s, s)$ because it acts locally like a noun phrase but takes scope semantically at a higher sentential level. The logical rule governing this operator is:

$$\frac{\Delta[A] \Rightarrow B \quad \Gamma[C] \Rightarrow D}{\Gamma[\Delta[q(A, B, C)]] \Rightarrow D} \ (qL).$$

The introduction of this connective raises a number of model-theoretical and proof-theoretical questions which have been addressed in [16], where it is shown that the q connective can be obtained as a defined operator of standard categorial type systems, and the above (qL) rule as a derived rule of inference. The solution there proposed can be further simplified, but for reason of space we cannot go in the detail of this aspect.

The scoping constructor q is intended to provide the full set of combinatorially possible scope relations in a multiple quantifier context — in that sense, it could be seen as the deductive version of May's [15] Scope Uniformity thesis. Beghelli and Stowell [3] have convincingly shown that GQs have non-uniform scope possibilities. The main claim of their theory is that, for certain combinations of quantifier types, the natural language grammar simply excludes certain logically possible scope construals. In our setting this requires type-refinement of the subtypes of a $q(np, s, s)$ assignment.

A first analysis following these ideas was provided in [4]. In this paper, the derivability relation, $\diamond \Box^\downarrow s \longrightarrow s \longrightarrow \Box^\downarrow \diamond s$ is used to distinguish three different sentential levels: the one lower than negation ($\diamond \Box^\downarrow s$), the negative one ($s$), and the one higher than negation ($\Box^\downarrow \diamond s$). The different distribution of GQs like *every N, a N* and *some N*, with respect to these sentential levels is encoded in their type assignments.

9

However, the linear derivability relation given by a pair of residuated operators, is not enough to account for more complex linguistic phenomena. For example, the type assignment proposed in [4] could not block the occurrences of negative polarity expressions like *any N* in a positive sentence.

We will show by means of an example, how we can use the type $(^0s)^0$ to solve this problem and how the different scope possibilities of positive vs. negative polarity items can be accounted for. Let's start with the lexicon entries we are interested in.

$$
\begin{array}{rcl}
\textit{didn't} & \in & ((s/(np\backslash s))\backslash s)/(np\backslash(^0s)^0). \\
\textit{any N} & \in & q(np, (^0s)^0, (^0s)^0). \\
\textit{some N} & \in & q(np, \Box^\downarrow\Diamond s, \Box^\downarrow\Diamond s).
\end{array}
$$

The occurrence of *any* in a positive context, e.g. *anybody left*, is blocked as shown by the proof below (as $(^{\mathbf{0}}s)^{\mathbf{0}} \longrightarrow \Box^\downarrow\Diamond s$ cannot be derived in the calculus):

$$
\dfrac{\dfrac{np \Rightarrow np \quad s \Rightarrow (^0s)^0}{np \bullet np\backslash s \Rightarrow (^0s)^0}\ (\backslash L) \quad \boxed{(^0s)^0 \not\Rightarrow \Box^\downarrow\Diamond s}}{q(np, (^0s)^0, (^0s)^0) \bullet np\backslash s \Rightarrow \Box^\downarrow\Diamond s}\ (qL)
$$

The ungrammaticality of, e.g., *anybody didn't leave* is proved in a similar way.

We now turn to the non-uniform behavior of polarity items with respect to negation. The combination of GQ and negation gives rise to scope ambiguity, which in type logic grammars corresponds to multiple proofs. In the case of polarity items (PI) not all the logical possibilities are allowed. In particular, *negative* PI will allow only for the reading with the negation having wide scope (¬GQ), while *positive* PI will produce the reading with negation having narrow scope (GQ¬). The proof-schemata in Figures 2 and 3 illustrate how the lexicon entries above correctly predict these linguistic phenomena.

When instantiating the quantifier $q(np, s_1, s_2)$ with a positive polarity item like *some book*, e.g. *John didn't read some book*, the proof-schema in Figure 2 will fail, while the one in Figure 3 will not. Applying the type assigned above to *some book*, $q(np, \Box^\downarrow\Diamond s, \Box^\downarrow\Diamond s)$, the sentential types of the quantifier, s_1, s_2, will both be instantiated with $\Box^\downarrow\Diamond s$. Therefore in Figure 2 the derivation $s_2 \longrightarrow (^0s)^0$ will fail (as $\Box^\downarrow\Diamond s \not\longrightarrow (^0s)^0$). While both the displayed derivations in Figure 3 will be provable, viz. $s \longrightarrow \Box^\downarrow\Diamond s$ and $\Box^\downarrow\Diamond s \longrightarrow \Box^\downarrow\Diamond s$.

On the other hand, when considering the derivations as a proof of *John didn't read any book*, the sentential types assigned to the negative polarity item *any book*, will be $s_1 = s_2 = (^0s)^0$, which give the correct derivations $s \longrightarrow (^0s)^0$ and $(^0s)^0 \longrightarrow (^0s)^0$ in Figure 2, and block the derivation in

10

Figure 3 with $(^0s)^0 \not\longrightarrow \Box^{\downarrow}\Diamond s$.

These simple examples show that different scope distribution exhibited by items which belong to the same syntactic category can be accounted for by means of derivability patterns among types. More precisely, the unary operators $(\Diamond, \Box^{\downarrow}, {}^{\mathbf{0}}\cdot, \cdot^{\mathbf{0}})$ provide a way to encode the different features which characterizes items of the same category. In the above examples we have used only a small part of the rich pattern of types at our disposal in $\mathsf{NL}(\Diamond, \cdot^{\mathbf{0}})$. The picture in Figure 1 summarizes some of the relations we have. Note that these relations can be further extended with a third level since besides the derivation $A \longrightarrow (^{\mathbf{0}}A)^{\mathbf{0}}$, $A \longrightarrow {}^{\mathbf{0}}(A^{\mathbf{0}})$ holds as well. In the next pages we will refer to the types using the corresponding abbreviations s_i given in Figure 1.
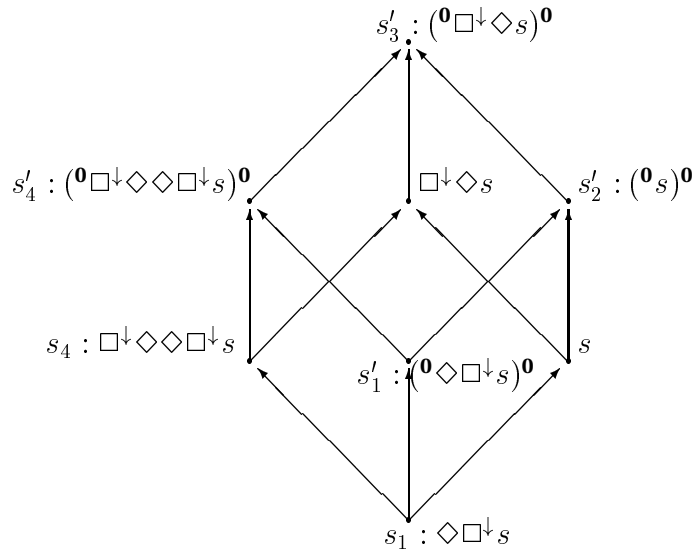

Fig. 1. Some derivability patterns in $\mathsf{NL}(\Diamond, \cdot^{\mathbf{0}})$

11

$$\cfrac{\cfrac{np \Rightarrow np \qquad \boxed{s \Rightarrow s_1}}{\cfrac{np \Rightarrow np \qquad \cfrac{np \bullet np\backslash s \Rightarrow s_1}{np \bullet ((np\backslash s)/np \bullet np) \Rightarrow s_1}\,(/L)}{\cfrac{np \bullet ((np\backslash s)/np \bullet \boxed{q(np,s_1,s_2)}) \Rightarrow (^0s)^0 \qquad \boxed{s_2 \Rightarrow (^0s)^0}}{(np\backslash s)/np \bullet q(np,s_1,s_2) \Rightarrow np\backslash(^0s)^0}\,(\backslash R)}}{\underbrace{np}_{\text{subject}} \bullet (\underbrace{(((s/(np\backslash s))\backslash s)/(np\backslash(^0s)^0)}_{\text{negation}} \bullet (\underbrace{(np\backslash s)/np}_{\text{trans. verb}} \bullet \underbrace{q(np,s_1,s_2)}_{\text{gen. quantifier}}))) \Rightarrow \Box^{\downarrow}\Diamond s}\,(qL) \quad \cfrac{np \Rightarrow s/(np\backslash s) \quad s \Rightarrow \Box^{\downarrow}\Diamond s}{np \bullet (s/(np\backslash s))\backslash s \Rightarrow \Box^{\downarrow}\Diamond s}\,(\backslash L) \,(/L)$$

Fig. 2. Wide scope negation (¬GQ)

$$\cfrac{np \Rightarrow np \qquad \cfrac{\cfrac{s \Rightarrow (^0s)^0 \quad np \Rightarrow np}{np\backslash s \Rightarrow np\backslash(^0s)^0}\,(\backslash R,L) \quad \cfrac{np \Rightarrow s/(np\backslash s) \quad \boxed{s \Rightarrow s_1}}{np \bullet (s/(np\backslash s))\backslash s \Rightarrow s_1}\,(\backslash L)}{\cfrac{np \bullet (((s/(np\backslash s))\backslash s)/(np\backslash(^0s)^0) \bullet np\backslash s) \Rightarrow s_1}{np \bullet (((s/(np\backslash s))\backslash s)/(np\backslash(^0s)^0) \bullet ((np\backslash s)/np \bullet np)) \Rightarrow s_1}\,(/L)}\,(/L)}{\underbrace{np}_{\text{subject}} \bullet (\underbrace{(((s/(np\backslash s))\backslash s)/(np\backslash(^0s)^0)}_{\text{negation}} \bullet (\underbrace{(np\backslash s)/np}_{\text{trans. verb}} \bullet \boxed{\underbrace{q(np,s_1,s_2)}_{\text{gen. quantifier}}})) \Rightarrow \Box^{\downarrow}\Diamond s}\,(qL) \quad \boxed{s_2 \Rightarrow \Box^{\downarrow}\Diamond s}$$

Fig. 3. Narrow scope negation (GQ¬)

In the remaining part of the paper we show how the above derivability patterns give a precise way to (i) gain a deeper understanding of the typology of polarity items proposed in the literature of formal linguistics; (ii) clarify the consequences predicted by the typologies, opening the way to further investigations; (iii) carry out cross-linguistic comparisons. In particular we will look at the typology of Greek polarity items discussed in [8] and compare it with Italian data and types.

3.2 A typology of polarity items in $NL(\diamondsuit, \cdot^{\mathbf{0}})$

Extending the results presented in [23,22] Giannakidou gives in [8] an in depth analysis of polarity items in terms of sensitivity to (non)-veridicality — where, intuitively, a non-veridical expression NV is such that when composed with a proposition p it does not entail that p is true. Besides the standard distinctions we introduced above between negative and positive polarity items, she gives evidence for a further classification among items of the former group. For reasons which will become clear soon, she refers to the whole group as affective PI (API) leaving the adjective "negative" to denote a particular kind of API. The polarity items which do not belong to API are considered as positive polarity items (PPI). We now roughly present the theory and sketch its typological account.

To give a categorial type logic (CTL) analysis of polarity items the main observations to keep in mind are the following. (a) Episodic sentences (E) can be either veridical or non-veridical, where the difference is marked by the occurrences of veridical vs. non-veridical expressions (e.g. *now* vs. *may*, respectively). (b) Among the non-veridical expressions we can distinguish the subset of the anti-veridical ones (AV) which are negation-like operators (e.g. *without, does not*). (c) Among the affective polarity items we have to differentiate negative polarity items (NPIs) which are defined as the items which require to be the argument of AV, and APIs which are felicitous also within the scope of NV. In CTL terms (a) means that the application of NV (and therefore AV) expressions will return an episodic sentence; (b) means that the type of AV must derive the type of NV; (c) says that APIs are of a weaker type than NPIs, hence the type assigned to API must derive the one of NPI. Let us illustrate this by means of the derivations sketched below.

Let $NV \in E/PI$, $AV \in E/NPI$ and $E/NPI \longrightarrow E/PI$, then

$$\frac{AV \in E/NPI \quad \text{NPI} \in NPI}{AV \circ \text{NPI} \in E} \qquad \frac{NV \in E/PI \quad \text{API} \in API}{NV \circ \text{API} \in E}$$

$$\frac{\dfrac{AV \in E/NPI}{AV \in E/API} \quad \text{PI} \in API}{AV \circ \text{API} \in E} \qquad \frac{\dfrac{NV \in E/API}{NV \in E/NPI} \, [*] \quad \text{NPI} \in NPI}{{}^{*}NV \circ \text{NPI} \in E}$$

The derivations show that if we assign to AV and NV types governed by the relations above, then an anti-veridical expression will compose both with a `NPI` and with an `API`, but a non veridical operator will not compose with `NPI` (as the inference in [*] fails). Due to the logical properties of the $/$, $E/API \not\longrightarrow E/NPI$ implies $NPI \not\longrightarrow API$ and $E/NPI \longrightarrow E/API$ means $API \longrightarrow NPI$. Therefore point (c) simply follows from (b).

More formally, this simple fact can be expressed using (one of) the derivability relations given in Figure 1. But the cube offers a much richer hierarchy of types which will allow us to make more fine-grained distinctions among polarity items as actually is required by the linguistic data. We look at the Greek data presented in [8] and to Italian data for a cross-linguistic analysis. In Giannakidou's analysis free choice items (`FCIs`) are shown to be sensitive to (non)-veridicality, and in particular to be a special case in the group of affective polarity items. We will include them in our data. In the examples below the licensing operator is emphasized and the licensed item is underlined. The $>$ stands for the scope distribution, e.g. Neg $>$ API means Neg has scope over API.

Greek

NPI: ipe leksi, API: kanenan, FCI: opudhipote

1. *Dhen* idha <u>kanenan</u>.   Neg $>$ API

   (tr. I didn't see anybody)

2. *Dhen* <u>ipe leksi</u> oli mera   Neg $>$ NPI

   (tr. He didn't say a word all day)

3. **Dhen* idha <u>opjondhipote</u>   *Neg $>$ FCI

   (tr. I didn't see anybody)

4. <u>Opjosdhipote</u> fititis *bori* na lisi afto to provlima.   Modal $>$ FCI

   (tr. Any student can solve this problem.)

5. *An* dhis tin Elena [<u>puthena</u>/<u>opudhipote</u>], . . .   Cond $>$ API/FCI

   (tr. If you see Elena anywhere, . . .)

6. *An* <u>pis leksi</u> tha se skotoso.   Cond $>$ NPI

   (tr. If you say a word, I will kill you)

For the reasons discussed above the following types will correctly predict these data. For completeness we include also the type for the `PPI` *kapjos* (tr. somebody).

Lexicon

PPI: $q(np, s_4, s_4)$, *kapjos*   NPI: $np\backslash s_2'$, *ipe leksi*

API: $q(np, s_1', s_1')$, *kanenan*   FCI: $q(np, s_4', s_4')$, *opudhipote*

modal: $(((s_4'/np)\backslash s_4')\backslash s_1)/(np\backslash s_4')$, *bori*   neg.: $(np\backslash s_1)/(np\backslash s_2')$, *dhen*

cond.: $(s_1/s_1')/s_3'$, *an*

We can compare the Greek data and types with the Italian ones.

Italian

NPI: nessuno, API: mai, FCI: chiunque

1. *Non* gioco <u>mai</u>   Neg > API
   (tr. I don't play ever)

2. *Non* ho visto <u>nessuno</u>   Neg > NPI
   (tr. I haven't seen anybody)

3. *Non ho visto <u>chiunque</u>   *Neg > FCI
   (tr. I haven't seen anybody)

4. <u>Chiunque</u> *può* risolvere questo problema   Modal > FCI
   (tr. Anybody can solve this problem)

5. *Puoi giocare <u>mai</u>   *Modal > API
   (tr. You can play ever)

6. *Puoi prendere in prestito <u>nessun</u> libro   *Modal > NPI
   (tr. You can borrow any book)

7. *Se* verrai <u>mai</u> a trovarmi, . . .   Cond > API
   (tr. If you ever come to visit me, . . .)

Lexicon

PPI: $q(np, s_4, s_4)$, *qualcuno*   NPI: $q(np, s_2', s_2')$, *nessuno*

API: $(np\backslash s_1)\backslash(np\backslash s_1')$, *mai*   FCI: $q(np, s_4'', s_4'')$, *chiunque*

modal: $(((s_4''/np)\backslash s_4'')\backslash s_1)/(np\backslash s_4'')$, *può*   neg.: $(np\backslash s_1)/(np\backslash s_2')$, *non*

cond: $(s_1/s_1')/s_4'$, *se*

The reader is referred to [8] for the complete analysis of the Greek data.

We summarize the results in the table below comparing it with the Italian ones.

Greek	FCI	API	NPI
Veridical	*	*	*
Negation	*	Yes	Yes
Modal verb	Yes	Yes	*
Conditional	Yes	Yes	Yes

Italian	FCI	API	NPI
Veridical	*	*	*
Negation	*	Yes	Yes
Modal verb	Yes	*	*
Conditional	*	Yes	*

Finally, the lexicon type assignments given for Greek and Italian can be summarized as follows:

Greek   Italian
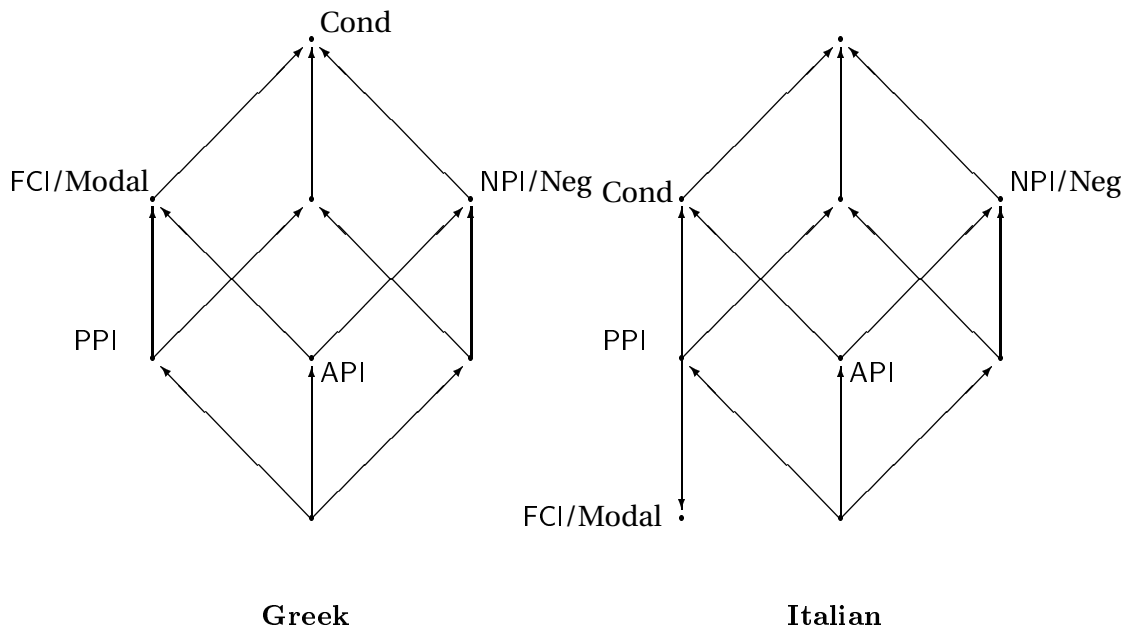
From this comparison the following conclusions can be drawn: (i) there could be contexts where both PPI and API are felicitous (e.g. conditionals and modal operators); (ii) there could be non-veridical operators which do not license polarity items (eg. in Italian FCI are not allowed in conditionals), (iii) there can be other sorts of polarity items sensitive to these and to other kinds of non-veridical expressions.

4   Conclusions

In this paper, we have focused on the minimal implementation of Galois connected operators in the base logic, and shown how this minimal addition enhances the accuracy of the grammatical logic.

An obvious topic for further research is the *communication* between the Galois operators and the unary and binary residuated families.

Another interesting direction is to investigate which is the true impact of the addition of *order-reversing* operators to the categorial types. In the main linguistic application in §3 we made use only of the *composition* of Galois operators $(^{\mathbf{0}}(\cdot))^{\mathbf{0}}, {}^{\mathbf{0}}((\cdot)^{\mathbf{0}})$ which are in themselves *order-preserving* operators.

Furthermore, our search for the right lexical type assignments suggests a possible connection between non-verdicality and $(^{\mathbf{0}}\cdot, \cdot^{\mathbf{0}})$, and veridicality and $(\square^{\downarrow}, \diamondsuit)$. This might shed light on the understanding of the semantic interpretations of these operators when used to reason with linguistic resources.

References

[1] Abrusci, V., *Phase semantics and sequent calculus for pure noncommutative classical linear propositional logic*, The Journal of Symbolic Logic **56** (1991), pp. 1403–1451.

[2] Areces, C. and R. Bernardi, *Analyzing the core of categorial grammar*, in: P. Blackburn and M. Kohlhase, editors, *Proceedings of ICoS-3*, 2001, pp. 7–28.

[3] Beghelli, F. and T. Stowell, *Distributivity and negation: The syntax of each and every*, in: A. Szabolcsi, editor, *Ways of Scope Taking*, Kluwer, 1997 pp. 72–107.

[4] Bernardi, R. and R. Moot, *Generalized quantifiers in declarative and interrogative sentences*, in: *Proceedings of ICoS-2*, Dagstuhl, 2000, pp. 1–15.

[5] Birkhoff, G., "Lattice Theory," American Mathematical Society, Providence, 1940, 1948, 1967.

[6] Došen, K., *A brief survey of frames for the Lambek calculus*, Zeitschrift für mathematischen Logik und Grundlagen der Mathematik **38** (1992), pp. 179–187.

[7] Dunn, J., *An abstraction of Galois connections and residuation with application to negation and various logical operations*, in: *JELIA 1990. Proceedings of the European Workshop on Logics in Artificial Intelligence*, 1991, pp. 31–51.

[8] Giannakidou, A., "The landscape of polarity items," Ph.D. thesis, University of Groningen (1997).

[9] Goré, R., *Substructural logics on display*, Logic Journal of the IGPL **6** (1998), pp. 451–504.

[10] Kurtonina, N., "Frames and labels. A modal analysis of categorial inference," Ph.D. thesis, OTS Utrecht University, ILLC Amsterdam University (1995).

[11] Kurtonina, N. and M. Moortgat, *Structural control*, in: P. Blackburn and M. de Rijke, editors, *Specifying syntactic structures (Amsterdam, 1994)*, CSLI Publications, Stanford, CA, 1997 pp. 75–113.

[12] Lambek, J., *From categorial to bilinear logic*, in: K. Došen and P. Schröder-Heister, editors, *Substructural Logics*, Oxford University Press, 1993 pp. 207–237.

[13] Lambek, J., *Type grammar revisited*, in: *Logical aspects of computational linguistics (Nancy, 1997)*, Springer, Berlin, 1999 pp. 1–27.

[14] Lambek, J., *Type grammars as pregroups*, Grammars **4** (2001).

[15] May, R., "The grammar of quantification," Ph.D. thesis, MIT (1977).

[16] Moortgat, M., *In situ binding: A modal analysis*, in: P. Dekker and M. Stokhof, editors, *Proceedings of the Tenth Amsterdam Colloqium*, ILLC, 1995, pp. 539–549.

[17] Moortgat, M., *Generalized quantification and discontinuous type constructors*, in: H. Bunt and A. van Horck, editors, *Discontinuous Constituency*, Mouton de Gruyter, Berlin, 1996 pp. 181–207.

[18] Moortgat, M., *Multimodal linguistic inference*, Journal of Logic, Language and Information **5** (1996), pp. 349–385.

[19] Moortgat, M., *Categorial type logics*, in: J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, The MIT Press, Cambridge, Massachusetts, 1997 pp. 93–178.

[20] Ore, O., *Galois connections*, Transactions of the American Mathematical Society **55** (1944), pp. 493–513.

[21] Restall, G., "An introduction to substructural logics," Routledge, 2000.

[22] van der Wouden, T., "Negative contexts," Ph.D. thesis, Unversity of Groningen (1994).

[23] Zwarts, F., *Nonveridical contexts*, Linguistic Analysis **25** (1995), pp. 286–312.