Analyzing the Core of Categorial Grammar

Carlos Areces (carlos.areces@loria.fr)
LORIA, Nancy, France.

Raffaella Bernardi (raffaella.bernardi@unibz.it)
Free University of Bolzano-Bozen, Bolzano-Bozen, Italy.

Abstract. Even though residuation is at the core of Categorial Grammar (Lambek, 1958), it is not always immediate to realize how standard logical systems like Multi-modal Categorial Type Logics (MCTL) (Moortgat, 1997) actually *embody* this property. In this paper, we focus on the basic system NL (Lambek, 1961) and its extension with unary modalities NL($\Diamond$) (Moortgat, 1996), and we spell things out by means of Display Calculi (DC) (Belnap, 1982; Goré, 1998). The use of structural operators in DC permits a sharp distinction between the core properties we want to impose on the logical system and the way these properties are projected into the logical operators. We will show how we can obtain Lambek residuated triple $\backslash$, / and $\bullet$ of binary operators, and how the operators $\Diamond$ and $\Box^{\downarrow}$ introduced by Moortgat in (Moortgat, 1996) are indeed their unary counterpart.

In the second part of the paper we turn to other important algebraic properties which are usually investigated in conjunction with residuation (Birkhoff, 1967): Galois and dual Galois connections. Again, DC let us readily define logical calculi capturing them. We also provide preliminary ideas on how to use these new operators when modeling linguistic phenomena.

Keywords: Categorial Grammar, Categorial Type Logics, Display Calculi, Residuation, Galois Connections

1. Categorial Grammar

In the Categorial Grammar approach to natural language analysis (Ajdukiewicz, 1935; Bar-Hillel, 1953) sentences are seen as sequences of function applications starting from the categories assigned to the lexical items in the lexicon. In (1958) Lambek shows that the categorial grammar intuitions can be formalized into a logical calculus: the grammaticality of a sentence can be decided by means of logical rules, if we consider categories as logical formulas. This idea is at the heart of what are today called Categorial Type Logics (CTLs). By means of these logics we can investigate logical properties of linguistic composition (like the impact of associativity and permutation in natural language phenomena). In other words, CTLs can account for grammaticality in a purely proof-theoretical way, and moreover, typical CTLs are *decidable*, that is they are amenable to computational treatment. More precisely, the standard categorial approach is to develop the correct "type assignment" for basic lexical items, from which certain linguistic phenomena

will be predicted by the logic. Once the basic types have been fixed, parsing a linguistic expression to check its membership to a given type, amounts to inferring the type in the logical system from the types assigned to its lexical components.

In addition to this pure proof-theoretical use of categorial systems, there is also an important semantic byproduct: the interpretation of (formally, the lambda term associated to) a linguistic expression can be obtained while inferring its type (Moortgat, 1997). This connection offers a rich framework where linguistic issues can be investigated from all three different points of view: purely syntactic checking of *type composition*, the *compositional meaning* of the linguistic expression, and their interface. Actually, the CTL approach can be understood as the result of analyzing this three sided linguistic picture (of syntax, semantics and their interface) from the standpoint of its syntactic vertex. Types, how to form them and which inference patterns they give rise to, are core issues in categorial type logics. In this paper, we will study some of these systems, analyze how their inference rules have been obtained, and in which way they define the behavior of their type forming operators.

The original sequent system NL, introduced by Lambek in (1961), consists of the operators $\backslash$, / and $\bullet$, indirectly governed by the algebraic law of residuation. While $\bullet$ can be seen as playing the role of linguistic composition, $\backslash$ and / allow the definition of functional types, which are sensitive to order. In Section 2.1 we will show how display calculi (DC) (Belnap, 1982; Goré, 1998) let us *directly* specify the residuation law by means of structural rules. The residuation behavior is then projected into the logical operators in a standard way.

Modern systems like those discussed in (Moortgat, 1997) are richer than NL. They incorporate unary logical operators and special devices to handle structures. In Section 2.2, we study NL($\Diamond$) introduced by Moortgat in (1996) by extending NL with unary operators $\Diamond$ and $\Box^{\downarrow}$, and show that they are indeed the unary counterpart of Lambek's residuated triple. This is no new result, the standard proof proceeds by showing that derivations in NL($\Diamond$) exactly match theorems obtained from the algebraic laws governing (unary and binary) residuation (see (Moortgat, 1996)). But this method provides little help on how to *actually obtain* the sequent rules of NL($\Diamond$) —capturing residuation and no more— in the first place; DC instead achieve this straightforwardly.

Interestingly, DC do not let us handle only residuation. The method described in Section 2 can be applied to other algebraic properties. In Section 3 we investigate Galois and dual Galois connections, and define logical calculi capturing them. Galois connections are interesting

because they give rise to new derivability relations among types while their composition is still a closure operator (as we discuss in Section 4).

We believe that the main contribution of this paper is methodological, and that display calculi do provide new light and help understand standard categorial type logics. More generally, we describe a recipe that help us explore a landscape of algebraic principles. We exemplify this method by showing how to extend the basic calculus NL with residuated, Galois and dual Galois connected operators of different arities in a systematic way.

2. Capturing Residuation

We start by formally introducing residuation and its ramifications in modern categorial type logics (see (Moortgat, 1996) for a much more complete discussion).

The property of residuation arises in the study of order-preserving mappings (Fuchs, 1963; Blyth and Janowitz, 1872; Dunn, 1991). Intuitively, in any partial order with a "product-like" operator $\bullet$, the (right) *residual* for an element a with respect to b is the largest c such that $b \bullet c$ is less than or equal to a. When the residual always exists for any two elements a, b in the structure, we can define the function $\cdot \backslash \cdot$ returning it. $\backslash$ and $\bullet$ are said to be *residuated*. If $\bullet$ is not commutative, then also a notion of left-residual naturally arises. For example, in the rational numbers (without 0), given any rational a, the residual with respect to b is simply $\frac{a}{b}$, and in this structure product and division are residuated functions.

More abstractly, the notion of residuated functions can be generally introduced for maps with n-ary arguments, but we restrict ourselves to unary and binary functions.

DEFINITION 1 (Residuation). *Let $\mathcal{A}_i = (A_i, \sqsubseteq_{A_i})$ be a partially ordered set. A pair of functions (f, g) such that $f : A_1 \to A_2$ and $g : A_2 \to A_1$ forms a* residuated pair *if* [RES$_1$] *holds.*

$$[\text{RES}_1] \qquad \forall x \in A_1, y \in A_2 \left(\begin{array}{rcll} fx & \sqsubseteq_{A_2} & y & \textit{iff} \\ x & \sqsubseteq_{A_1} & gy & \end{array} \right).$$

A triple of functions (f, g, h) such that $f : A_1 \times A_2 \to A_3$, $g : A_1 \times A_3 \to A_2$, $h : A_3 \times A_2 \to A_1$ forms a residuated triple *if* [RES$_2$] *holds.*

$$[\text{RES}_2] \qquad \forall x \in A_1, y \in A_2, z \in A_3 \left(\begin{array}{rcll} f(x,y) & \sqsubseteq_{A_3} & z & \textit{iff} \\ y & \sqsubseteq_{A_2} & g(x,z) & \textit{iff} \\ x & \sqsubseteq_{A_1} & h(z,y) & \end{array} \right).$$

In both cases the function f is said to be the head *of the residuated pair or triple.*

REMARK 1. *It is important to mention that residuation has an impact on monotonicity behavior. In fact, saying that (f, g) is a residuated pair is equivalent to the conditions 1) and 2) below, where we write f is a $[\uparrow]$-function (f is a $[\downarrow]$-function) meaning that f is upwards (downwards) monotonic in its argument,*

1. *f and g are $[\uparrow]$-functions.*
2. *$\forall y \in A_2(fgy \sqsubseteq_{A_2} y)$ and $\forall x \in A_1(x \sqsubseteq_{A_1} gfx)$.*

Similarly, saying that (f, g, h) is a residuated triple is equivalent to requiring

1. *f is a $[\uparrow, \uparrow]$-function, g is an $[\downarrow, \uparrow]$-function and h is an $[\uparrow, \downarrow]$-function.*
2. *$\forall x \in A_1, y \in A_2, z \in A_3((f(x, g(x, z)) \sqsubseteq_{A_3} z)$ & $(y \sqsubseteq_{A_2} g(x, f(x, y)))$ & $(f(h(z, y), y) \sqsubseteq_{A_3} z)$ & $(x \sqsubseteq_{A_1} h(f(x, y), y)))$.*

In what follows we will be mainly interested in logical operations $O_i : \mathsf{FORM}^k \to \mathsf{FORM}$, and we will investigate their behavior with respect to the poset $\langle \mathsf{FORM}, \vdash \rangle$ where $\vdash$ is the derivability relation.

We now turn to categorial type logics which are also commonly known as "logics of residuation." Let us see why by considering the system NL introduced in (Lambek, 1961).

DEFINITION 2 (Logical language of NL). *Given the set* ATOM *of atomic propositional symbols, the logical language of* NL *is defined recursively as*

$$\mathsf{FORM} ::= \mathsf{ATOM} \mid \mathsf{FORM}/\mathsf{FORM} \mid \mathsf{FORM}\backslash\mathsf{FORM} \mid \mathsf{FORM} \bullet \mathsf{FORM}.$$

An axiomatic deductive system for NL is given as follows.

DEFINITION 3 (NL: Axiomatic system). *The system* NL *is defined by the axiom and rules below. Given $A, B, C \in$ FORM*

$$\begin{array}{rl} \text{[REFL]} & A \vdash A, \\ \text{[TRANS]} & \textit{If } A \vdash B \textit{ and } B \vdash C, \textit{ then } A \vdash C, \\ \text{[RES]} & A \bullet B \vdash C \textit{ iff } B \vdash A\backslash C \textit{ iff } A \vdash C/B. \end{array}$$

NL is commonly called the pure logic of residuation and rightly so as we can see from its axiomatic presentation. [REFL] and [TRANS] define minimal properties for the inference relation $\vdash$ while [RES] characterizes $\bullet$, $\backslash$ and $/$ as a residuated triple[1].

The axiomatic presentation of NL clearly shows that residuation directly governs the behavior of its type forming operators. But even though axiomatic deductive system can be effectively used when establishing model-theoretical properties like completeness, it is not appropriate for proof-theoretical investigations. In particular, the [TRANS] rule above violates the sub-formula property, introducing non determinism in the proof search. As with classical propositional logic, an alternative is the formulation of an equivalent Gentzen style presentation, in which the use of the counterpart of [TRANS], the [Cut] rule, can be proved to be redundant ([Cut] elimination).

Standard Gentzen systems in which [Cut] can be eliminated do enjoy the sub-formula property, and hence the search space for proofs of a given sequent is finite. The sequent presentation of NL is proof theoretically well behaved: it enjoys the subformula property and provides a backward-chaining decision procedure (Lambek, 1958; Lambek, 1961). This good computational behavior makes these systems well suited to the study of the inference relations between types.

While in the axiomatic presentation the derivability relation holds between formulas of the logical language, in a Gentzen system it is stated in terms of *sequents*: pairs $\Gamma \vdash A$ where Γ is a structured configuration of formulas or *structural terms* and A is a logical formula. The set TERM of structural terms needed for a sequent presentation of NL is very simple

$$\mathsf{TERM} ::= \mathsf{FORM} \mid (\mathsf{TERM}, \mathsf{TERM}).$$

The logical rules in the Gentzen system for NL are given in Figure 1 below. In the figure, A, B, C are formulas, Γ, Δ are structural terms and the notation $\Gamma[\varphi]$ is used to single out a particular instance of the substructure φ in Γ.

As we can see from inspecting the rules in Figure 1, it is not immediately obvious that they are characterizing the same derivability relation as the one characterized by the axiomatic presentation of NL. To establish the equivalence between the two presentations, define the translation $\cdot^t : \mathsf{TERM} \to \mathsf{FORM}$ as

$$\begin{aligned} (\Gamma_1, \Gamma_2)^t &= (\Gamma_1^t \bullet \Gamma_2^t), \\ A^t &= A, \text{ for } A \in \mathsf{FORM}. \end{aligned}$$

[1] [RES] could also be understood as a kind of deduction theorem. But while a deduction theorem is better seen as a link between the meta-language and the object language, [RES] relates three operators in the object language.

$$\frac{}{A \vdash A}\ [\text{Ax}] \qquad \frac{\Delta \vdash A \quad \Gamma[A] \vdash C}{\Gamma[\Delta] \vdash C}\ [\text{Cut}]$$

$$\frac{\Delta \vdash B \quad \Gamma[A] \vdash C}{\Gamma[(A/B, \Delta)] \vdash C}\ [/\text{L}] \qquad \frac{(\Gamma, B) \vdash A}{\Gamma \vdash A/B}\ [/\text{R}]$$

$$\frac{\Delta \vdash B \quad \Gamma[A] \vdash C}{\Gamma[(\Delta, B\backslash A)] \vdash C}\ [\backslash\text{L}] \qquad \frac{(B, \Gamma) \vdash A}{\Gamma \vdash B\backslash A}\ [\backslash\text{R}]$$

$$\frac{\Gamma[(A, B)] \vdash C}{\Gamma[A \bullet B] \vdash C}\ [\bullet\text{L}] \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{(\Gamma, \Delta) \vdash A \bullet B}\ [\bullet\text{R}]$$

Figure 1. Gentzen sequent calculus for NL

PROPOSITION 1 (See (Lambek, 1958; Lambek, 1961)). *If $A \vdash B$ is a theorem of the axiomatic presentation of* NL *then there is a Gentzen proof of $A \vdash B$. And for every proof of a sequent $\Gamma \vdash B$, $\Gamma^t \vdash B$ is a theorem.*

The system presented in Figure 1 includes the [Cut] rule but Lambek proved, also in (Lambek, 1958), that the rule is admissible, in the sense that it does not increase the set of theorems that can already be derived using just the other rules.

PROPOSITION 2 ([Cut] elimination and decidability). *The* [Cut] *rule is admissible in* NL*, and the system is decidable.*

There seems to be a tension in the standard approach we described above. On the one hand, while the axiomatic calculus crisply captures the notion of residuation we are interested in, it is not appropriate for proof-theoretical manipulation. On the other hand, the decidable [Cut] free sequent presentation hides the residuated behavior of the operators, requiring a "verification check" as shown in Proposition 1. In the next section we will explain how display calculi are able to resolve this tension.

2.1. Displaying Residuation

Display calculi, introduced by Belnap in (1982), provide a Gentzen style proof-theoretical framework designed to capture many different logics in one uniform setting. DC generalize Gentzen's notion of structures by

using multiple, complex, structural connectives. One of the main characteristics of DC is a general cut-elimination theorem, which applies whenever the rules of the display calculus obey certain, easily checked, conditions. We will base our presentation on the system introduced by Goré in (1998). The main innovation of Goré's system over Belnap's concerns the use of additional structural connectives to capture the inherent duality of every logic, by means of dual sets of display postulates. Building on these features, DC obtain the "display property" which gives them their name: any particular constituent of a sequent can be turned into the whole of the right or left side by moving other constituents to the other side. This property is strongly used in the general cut-elimination method. But for our approach more interesting than the display property is the ability of DC to define the behavior of their logical operators in terms of *structural properties* —sequent rules involving only structural operators.

Let us start by introducing the appropriate logical and structural language for the DC we want to investigate.

DEFINITION 4 (DC language). *Given a set* ATOM *of atomic propositional symbols and the sets* $\mathsf{OP}_s^2 = \{;, <, >\}$ *and* $\mathsf{OP}_l^2 = \{\otimes, \leftarrow, \rightarrow\}$ *of structural and logical operators respectively, the set* FORM *of logical formulas and the set* STRUCT *of structural formulas are defined as*

$$\begin{aligned} \mathsf{FORM} &::= \mathsf{ATOM} \mid \mathsf{FORM} * \mathsf{FORM} \text{ for } * \in \mathsf{OP}_l^2. \\ \mathsf{STRUCT} &::= \mathsf{FORM} \mid \mathsf{STRUCT} * \mathsf{STRUCT} \text{ for } * \in \mathsf{OP}_s^2. \end{aligned}$$

The behavior of the structural operators is explicitly expressed by means of display postulates. In what follows, we will use variables X, Y, Z, V, W to denote structural formulas, and reserve A, B, C for logical formulas. In the case of residuation, we can directly express that $(;, <, >)$ is a residuated triple by encoding the relation holding among the operators (Definition 1) in the following structural rule.

$$[\text{rp}]\ \frac{\dfrac{X; Y \vdash Z}{\overline{\overline{Y \vdash X > Z}}}}{\overline{\overline{X \vdash Z < Y}}}$$

Notice that $\vdash$ and the double lines replace the $\sqsubseteq_{A_i}$ and the "iff condition" of Definition 1, respectively.

What remains to be done is to project the residuation behavior of $(;, <, >)$ into the corresponding logical operators $(\otimes, \leftarrow, \rightarrow)$. The general methodology is described in detail in (Goré, 1998). In a nutshell, it works as follows. We are in search of a right and left introduction rule for each of the logical operators, we can obtain $[\otimes \vdash]$, $[\vdash \leftarrow]$ and $[\vdash \rightarrow]$

directly from [rp] by projection. In the literature on DC these rules are usually called *rewrite rules*.

To obtain the still missing rules we have to work only slightly harder. As we pointed out in Remark 1, from the fact that $(;, <, >)$ are residuated, we know their monotonicity behavior, and this is exactly what we need. Let s be a structural operator and l its corresponding logical counterpart. In the schema below we will select whether the consequent of the rule is $s(X,Y) \vdash l(V,W)$ or $l(X,Y) \vdash s(V,W)$ depending on the needed rule.

$$\frac{V \vdash X \quad W \vdash Y}{[l,s](X,Y) \vdash [s,l](V,W)} \text{ if } s \text{ is } [\downarrow,\downarrow]$$

$$\frac{X \vdash V \quad Y \vdash W}{[l,s](X,Y) \vdash [s,l](V,W)} \text{ if } s \text{ is } [\uparrow,\uparrow]$$

$$\frac{X \vdash V \quad W \vdash Y}{[l,s](X,Y) \vdash [s,l](V,W)} \text{ if } s \text{ is } [\uparrow,\downarrow]$$

$$\frac{V \vdash X \quad Y \vdash W}{[l,s](X,Y) \vdash [s,l](V,W)} \text{ if } s \text{ is } [\downarrow,\uparrow]$$

Applying the schema above, we obtain $[\vdash \otimes]$, $[\leftarrow\vdash]$, and $[\rightarrow\vdash]$. The full set of rules is as follows.

$$\frac{A \vdash X \quad Y \vdash B}{A \leftarrow B \vdash X < Y}\ [\leftarrow\vdash] \qquad \frac{Z \vdash A < B}{Z \vdash A \leftarrow B}\ [\vdash\leftarrow]$$

$$\frac{A; B \vdash Z}{A \otimes B \vdash Z}\ [\otimes\vdash] \qquad \frac{Y \vdash B \quad X \vdash A}{Y; X \vdash B \otimes A}\ [\vdash \otimes]$$

$$\frac{X \vdash A \quad B \vdash Y}{A \rightarrow B \vdash X > Y}\ [\rightarrow\vdash] \qquad \frac{Z \vdash A > B}{Z \vdash A \rightarrow B}\ [\vdash\rightarrow]$$

These rules will immediately encode the proper tonicity of the operators. It is also easy to prove that the logical operators indeed satisfy the residuation property. We show two of the required four derivations below.

$$\dfrac{\dfrac{B \vdash A \rightarrow C \quad \dfrac{A \vdash A \quad C \vdash C}{A \rightarrow C \vdash A > C}\ [\rightarrow\vdash]}{B \vdash A > C}\ [\text{Cut}]}{\dfrac{A; B \vdash C}{A \otimes B \vdash C}\ [\otimes\vdash]}\ [\text{rp}]$$

$$\dfrac{\dfrac{\dfrac{A \vdash A \quad B \vdash B}{A; B \vdash A \otimes B}\,[\vdash \otimes] \qquad A \otimes B \vdash C}{A; B \vdash C}\,[\text{Cut}]}{\dfrac{B \vdash A > C}{B \vdash A \rightarrow C}\,[\vdash \rightarrow]}\,[\text{rp}]$$

Similarly, we can also prove the "composition property" we mentioned in Remark 1.

As we can see, DC provide guidance in our logic engineering task of designing a sequent calculus characterizing the behavior of a triple of residuated operators. Moreover, we can readily verify the conditions specified by Belnap and conclude that the calculus is cut-free. If we compare the calculus just obtained with the one introduced in Figure 1 we immediately notice similarities, but also important differences, the most relevant being the presence of only one structural operator, and the restriction to a single formula in the right hand side of sequents. It is not too difficult to restrict the language to obtain a perfect match (but of course, in doing so we would be giving up the display property, and "abandoning" DC and its general theorem concerning cut-elimination). Consider, for example, the $[\rightarrow\vdash]$ rule

$$\dfrac{X \vdash A \quad B \vdash C}{A \rightarrow B \vdash X > C}\,[\rightarrow\vdash]$$

by [rp]

$$\dfrac{\dfrac{X \vdash A \quad B \vdash C}{A \rightarrow B \vdash X > C}\,[\rightarrow\vdash]}{X; A \rightarrow B \vdash C}\,[\text{rp}]$$

hence

$$\dfrac{X \vdash A \quad B \vdash C}{X; A \rightarrow B \vdash C}\,.$$

By replacing ';' by ',' and '$\rightarrow$' by '$\backslash$' and adding structural contexts (which are now required given that we have lost the display property) we obtain [$\backslash$L]

$$\dfrac{X \vdash A \quad \Gamma[B] \vdash C}{\Gamma[(X, A\backslash B)] \vdash C}\,[\backslash\text{L}].$$

In the next section we will treat in detail unary residuation and investigate the pair of operators $\Diamond$ and $\Box^{\downarrow}$.

2.2. The Unary Operators

The system $\mathsf{NL}(\diamond)$ introduced in (Moortgat, 1996; Moortgat, 1997) is obtained from NL by the addition of the unary modalities $\diamond$ and $\Box^{\downarrow}$, and it is actively used in the analysis of linguistic phenomena. But $\diamond$ and $\Box^{\downarrow}$ have been sometimes "looked down upon" as extraneous to a calculus of pure residuation. We will show that we can straightforwardly mimic what we did in the previous section. Starting by spelling out the law of residuation for unary functions, we derive a sequent calculus that can be compiled into the standard calculus for $\mathsf{NL}(\diamond)$.

Given a set ATOM of atomic propositional symbols and the sets $\mathsf{OP}^1_s = \{\bullet, \circ\}$ and $\mathsf{OP}^1_l = \{\blacklozenge, \Box\}$ of unary structural and logical operators and the sets $\mathsf{OP}^2_s = \{;, <, >\}$ and $\mathsf{OP}^2_l = \{\otimes, \leftarrow, \rightarrow\}$ of binary structural and logical operators, the set FORM of logical formulas and the set STRUCT of structural formulas for a display calculus presentation of $\mathsf{NL}(\diamond)$ are in the standard way[2].

Again we start by specifying the behavior of the residuated structural pair $(\circ, \bullet)$,

$$\frac{\bullet X \vdash Y}{X \vdash \circ Y}\ [\mathrm{rp}],$$

and we obtain the rules for the logical operators by projection and monotonicity behavior. The full set of rules is given below.

$$\frac{A \vdash X}{\Box A \vdash \circ X}\ [\Box \vdash] \qquad \frac{X \vdash \circ A}{X \vdash \Box A}\ [\vdash \Box]$$

$$\frac{\bullet A \vdash X}{\blacklozenge A \vdash X}\ [\blacklozenge \vdash] \qquad \frac{X \vdash A}{\bullet X \vdash \blacklozenge A}\ [\vdash \blacklozenge]$$

We can prove that $(\blacklozenge, \Box)$ is a residuated pair.

$$\dfrac{\dfrac{\dfrac{A \vdash \Box B \quad \dfrac{\dfrac{B \vdash B}{\Box B \vdash \circ B}\ [\Box \vdash]}{}\ [\mathrm{Cut}]}{A \vdash \circ B}\ [\mathrm{rp}]}{\bullet A \vdash B}\ [\blacklozenge \vdash]}{\blacklozenge A \vdash B} \qquad \dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{\bullet A \vdash \blacklozenge A}\ [\vdash \blacklozenge] \quad \blacklozenge A \vdash B}{}\ [\mathrm{Cut}]}{\bullet A \vdash B}\ [\mathrm{rp}]}{A \vdash \circ B}\ [\mathrm{rp}]}{A \vdash \Box B}\ [\vdash \Box]$$

Now we "compile" the structural postulate [rp] to obtain the logical rules in the standard Gentzen presentation of $\mathsf{NL}(\diamond)$, as we did in the case of binary operators. We spell out the needed steps for the $\Box$ operator and obtain the rules $[\Box^{\downarrow}\mathrm{L}]$ and $[\Box^{\downarrow}\mathrm{R}]$ as presented in (Moortgat, 1997).

[2] Again we will first follow the notation of Goré in (Goré, 1998) to revert to the standard notation used in $\mathsf{NL}(\diamond)$ during the compilation step.

$$\frac{A \vdash B}{\Box A \vdash \circ B}\ [\Box \vdash] \quad \text{by [rp]} \quad \frac{\dfrac{A \vdash B}{\Box A \vdash \circ B}\ [\Box \vdash]}{\bullet \Box A \vdash B}\ [\text{rp}]$$

$$\text{by compilation} \quad \frac{\Gamma[A] \vdash B}{\Gamma[\langle \Box^{\downarrow} A \rangle] \vdash B}\ [\Box^{\downarrow}\text{L}].$$

$$\frac{X \vdash \circ A}{X \vdash \Box A}\ [\Box \vdash] \quad \text{by [rp]} \quad \frac{\dfrac{\bullet X \vdash A}{X \vdash \circ A}\ [\text{rp}]}{X \vdash \Box A}\ [\Box \vdash]$$

$$\text{by compilation} \quad \frac{\langle X \rangle \vdash A}{X \vdash \Box^{\downarrow} A}\ [\Box^{\downarrow}\text{R}].$$

The logical rules for $\Diamond$ are obtained straightforwardly in a similar way.

As we said in the introduction, DC are not limited to residuation properties (even though they are an important example, as residuation aids in achieving the display property). The method we have used above can handle other kind of algebraic properties, assuming that they can be encoded in terms of structural rules. In the next section we turn to Galois and dual Galois connections.

3. Capturing Galois and Dual Galois Connections

If we look at NL and at examples of how it is used in modeling linguistic phenomena, we notice that sometimes only the $\backslash$ and / operators are required, and their behavior is not characterized by a residuation law. Actually, $\backslash$ and / form a Galois connection when their positive argument is fixed. This is exactly what is used in CTL, for example, to account for the rising of noun phrases np to generalized quantifiers $(s/np)\backslash s$. The fact that / and $\backslash$ are Galois connected means exactly that for any two types A and B we can infer $(B/A)\backslash B$ from A.

Under this light, even though calling NL the pure calculus of residuation is correct, it is also misleading as it hides the fact that a Galois connected pair is also "living" inside as a subsystem. And in line with the work we did in Section 2.2 we could wonder whether we can also extend NL($\Diamond$) with a pair of independent unary Galois connected operators. But let us start by formally introducing the algebraic properties we want to investigate.

DEFINITION 5 (Galois connections). *Let $\mathcal{A}_i = (A_i, \sqsubseteq_{A_i})$ be a partially ordered set. Consider a pair of functions $f : A_1 \to A_2$ and*

$g : A_2 \to A_1$. *The pair* (f, g) *is called a* Galois connection *if* [GC] *below holds.*

$$[\text{GC}] \qquad \forall x \in A_1, y \in A_2 \left(\begin{array}{lll} y & \sqsubseteq_{A_2} & fx \quad \textit{iff} \\ x & \sqsubseteq_{A_1} & gy \end{array} \right).$$

The pair (f, g) *is called a* dual Galois connection *if* [DGC] *below holds.*

$$[\text{DGC}] \qquad \forall x \in A_1, y \in A_2 \left(\begin{array}{lll} fx & \sqsubseteq_{A_2} & y \quad \textit{iff} \\ gy & \sqsubseteq_{A_1} & x \end{array} \right).$$

As with residuation, there is an equivalent formulation of these properties in terms of their monotonicity behavior and a composition rule. [GC], for example, is equivalent to require that f and g are both $[\downarrow]$-functions, and that for all x, $x \sqsubseteq fgx$, and $x \sqsubseteq gfx$ (here again, we just consider f and g as functions defined on the same poset).

Galois connected operators have been also studied in the context of Linear Logic (see (Lambek, 1993; Abrusci, 1991; Goré, 1998; Restall, 2000)), and by Lambek in, e.g., (2001). In contrast with these line of work, in which Galois properties are mixed with extra features guaranteeing, for example, a double negation law, we focus on the pure Galois properties and investigate the effects of adding Galois connected operators to the base multimodal logic of residuation, $\mathsf{NL}(\diamond)$.

The steps we will take to provide a display calculus encoding [GC] and [DGC] should be by now well known. We only provide details for [GC]. We start by explicitly writing the algebraic property characterizing a Galois connection for a pair of structural operators $(\sharp, \flat)$.

$$\frac{Y \vdash \flat X}{X \vdash \sharp Y} \ [\text{gc}].$$

We now project this behavior into the logical operators $({}^0(\cdot), (\cdot)^0)$ as it is shown below.

$$\frac{Z \vdash A}{{}^0(A) \vdash \flat Z} \ [{}^0(\cdot) \vdash] \qquad \frac{Z \vdash \flat A}{Z \vdash {}^0(A)} \ [\vdash {}^0(\cdot)]$$

$$\frac{Z \vdash A}{(A)^0 \vdash \sharp Z} \ [(\cdot)^0 \vdash] \qquad \frac{Z \vdash \sharp A}{Z \vdash (A)^0} \ [\vdash (\cdot)^0]$$

To move closer to standard sequent presentations of CTL, we need to compile [gc] into the logical rules. We can take $[{}^0(\cdot) \vdash]$ and $[(\cdot)^0 \vdash]$ as they are as our $[{}^0(\cdot)\text{L}]$ and $[(\cdot)^0\text{L}]$. To obtain $[{}^0(\cdot)\text{R}]$ and $[(\cdot)^0\text{R}]$ we need to apply [gc],

$$\frac{Z \vdash \flat A}{Z \vdash {}^0(A)}\ [\vdash {}^0(\cdot)] \quad \text{by [gc]} \quad \frac{\dfrac{A \vdash \sharp Z}{Z \vdash \flat A}\ [\text{gc}]}{Z \vdash {}^0(A)}\ [\vdash {}^0(\cdot)] \quad \text{by compilation}$$

$$\text{by compilation} \quad \frac{A \vdash \sharp Z}{Z \vdash {}^0(A)}\ [{}^0(\cdot)\text{R}].$$

$$\frac{Z \vdash \sharp A}{Z \vdash (A)^0}\ [\vdash (\cdot)^0] \quad \text{by [gc]} \quad \frac{\dfrac{A \vdash \flat Z}{Z \vdash \sharp A}}{Z \vdash (A)^0}\ [\vdash (\cdot)^0]$$

$$\text{by compilation} \quad \frac{A \vdash \flat Z}{Z \vdash (A)^0}\ [(\cdot)^0\text{R}].$$

The full set of rules obtained is shown below. Notice that given the nature of Galois connections (which involves a permutation in the order of the poset), it is not possible to eliminate the structural operators from the right hand side of the sequents. This is an important difference with respect to what we obtained in the previous sections. For similar reasons, the system looses its cut elimination property. See (Areces et al., 2001) for a cut-free version.

$$\frac{Z \vdash A}{{}^0(A) \vdash \flat Z}\ [{}^0(\cdot)\text{L}] \qquad \frac{A \vdash \sharp Z}{Z \vdash {}^0(A)}\ [{}^0(\cdot)\text{R}]$$

$$\frac{Z \vdash A}{(A)^0 \vdash \sharp Z}\ [(\cdot)^0\text{L}] \qquad \frac{A \vdash \flat Z}{Z \vdash (A)^0}\ [(\cdot)^0\text{R}]$$

$$\frac{X \vdash Y \quad Y \vdash Z}{X \vdash Z}\ [\text{Cut}]$$

The proofs below show that the $(\cdot)^0$ and ${}^0(\cdot)$ operators are indeed Galois connected,

$$\frac{A \vdash (B)^0 \quad \dfrac{B \vdash B}{(B)^0 \vdash \sharp B}\ [(\cdot)^0\text{L}]}{\dfrac{A \vdash \sharp B}{B \vdash {}^0(A)}\ [{}^0(\cdot)\text{R}]}\ [\text{Cut}] \qquad \frac{A \vdash {}^0(B) \quad \dfrac{B \vdash B}{{}^0(B) \vdash \flat B}\ [{}^0(\cdot)\text{L}]}{\dfrac{A \vdash \flat B}{B \vdash (A)^0}\ [(\cdot)^0\text{R}]}\ [\text{Cut}]$$

That is, the rule [gc] holds for ${}^0(\cdot)$ and $(\cdot)^0$. Moreover, the operators satisfy the appropriate Galois composition laws [gcl].

$$\frac{\dfrac{A \vdash A}{(A)^0 \vdash \sharp A}\,[(\cdot)^0\mathrm{L}]}{A \vdash {}^0((A)^0)}\,[{}^0(\cdot)\mathrm{R}] \qquad \frac{\dfrac{A \vdash A}{{}^0(A) \vdash \flat A}\,[{}^0(\cdot)\mathrm{L}]}{A \vdash ({}^0(A))^0}\,[(\cdot)^0\mathrm{R}]$$

From these, the fact that the operators are $[\downarrow]$-functions follows immediately.

$$\frac{\dfrac{A \vdash B}{A \vdash {}^0((B)^0)}\,[\mathrm{gcl}]}{(B)^0 \vdash (A)^0}\,[\mathrm{gc}] \qquad \frac{\dfrac{A \vdash B}{A \vdash ({}^0(B))^0}\,[\mathrm{gcl}]}{{}^0(B) \vdash {}^0(A)}\,[\mathrm{gc}]$$

4. New Derivability Relations

In this section, we discuss possible applications of some of the logical properties of the systems we have been investigating above.

When working with a logic to reason with linguistic resources, one of the most important features are the derivability relations among types the proof system can establish. A well known application of this aspect of the Lambek calculi is the logical treatment they offer of the *lifting* of np, $(np \vdash (s/np)\backslash s)$ first used by Montague (1974), and the *value raising principle* (e.g. $np/n \vdash (s/(np\backslash s))/n)$) introduced as a primitive postulate in (Partee and Rooth, 1983). If we compare Definition 5 of Galois connections with the inferences used in the analysis of these properties, we clearly see that they hinge on the fact that $(/, \backslash)$ is a Galois connected pair. The lifting and value rising properties are indeed instantiations of the composition law for Galois connections.

As pointed out in (Moortgat, 1997) the composition of Galois connections defines an upwards monotonic function * which is a *closure* operator satisfying $A \vdash A^*$ and $(A^*)^* \vdash A^*$. The same holds for the composition of residuated pairs. I.e.

$$A \vdash ({}^0(A))^0 \dashv ({}^0(({}^0(A))^0))^0 \qquad A \vdash \Box^{\downarrow}\Diamond A \dashv \Box^{\downarrow}\Diamond\Box^{\downarrow}\Diamond A.$$

Similar derivability relations are used in (Bernardi and Moot, 2002) to account for scope ambiguity phenomena. In that paper the attention is focused on the different scope possibilities of generalized quantifiers (GQs) with respect to negation. The derivability relation $\Diamond\Box^{\downarrow}s \vdash s \vdash \Box^{\downarrow}\Diamond s$ 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 ($\Box^{\downarrow}\Diamond s$) (notice how we can use the type hierarchy obtained by means of the residuated unary operators to account for the

variety among expressions of the same linguistic category). The different scope possibilities of generalized quantifiers like *any n*, *a n* and *some n* are anchored to their type assignments. However, the one dimensional derivability relation given by a pair of residuated operators may not be enough to account for more intriguing linguistic phenomena, as we will exemplify below in the modeling of GQs sensitive to the polarity of their context. In particular, the proposal presented in (Bernardi and Moot, 2002) does not account for the fact that negative polarity expressions like *any* cannot occur in a positive sentence. We will show how Galois operators can be used to solve this problem.

We consider a linguistic string to be a grammatical sentence if it is proved to be of type $\Box^{\downarrow}\Diamond s$. We use the following abbreviations: $s_1 := \Diamond\Box^{\downarrow}s$, $s_2 := s$ and $s_3 := \Box^{\downarrow}\Diamond s$, viz. $s_1 \vdash s_2 \vdash s_3$, to better visualize the different "sentential levels" encoded in the types. Consider the following type assignments,

$$\begin{array}{llll} \textit{didn't} & \in (np\backslash s_2)/(np\backslash s_2) & \textit{any } n & \in (s_1/np)\backslash s_1 \\ \textit{directed} & \in (np\backslash s_1)/np & \textit{a } n & \in (s_2/np)\backslash s_2 \\ & & \textit{some } n & \in (s_3/np)\backslash s_3 \end{array}$$

From these type assignments, it follows that when parsing a "negative sentence" with a GQ in object position, e.g. *Coppola didn't direct any movie* or *Coppola didn't direct some movie*, the proofs [1a] and [1b] below are obtained, providing the two readings with negation having wide and narrow scope, respectively.

$$\text{[1a] } \neg GQ \qquad \dfrac{\dfrac{np \vdash np \quad np \vdash np \quad s_1 \vdash s_x \quad \boxed{s_y \vdash s_2}}{\vdots}}{\dfrac{((np\backslash s_1)/np, (s_x/np)\backslash s_y) \vdash np\backslash s_2 \qquad \dfrac{np \vdash np \quad s_2 \vdash s_3}{np, np\backslash s_2 \vdash s_3}\,[\backslash\text{L}]}{\underbrace{np}_{\text{sub}}, \underbrace{((np\backslash s_2)/(np\backslash s_2)}_{\text{didn't}}, \underbrace{((np\backslash s_1)/np}_{\text{tv}}, \underbrace{(s_x/np)\backslash s_y)}_{\text{GQ}}) \vdash s_3}\,[/\text{L}]}$$

$$\text{[1b] } GQ\neg \qquad \dfrac{\dfrac{np \vdash np \quad np \vdash np \quad s_1 \vdash s_2 \quad np \vdash np \quad \boxed{s_2 \vdash s_x}}{\vdots}}{\underbrace{np}_{\text{sub}}, \underbrace{((np\backslash s_2)/(np\backslash s_2)}_{\text{didn't}}, \underbrace{((np\backslash s_1)/np}_{\text{tv}}, \underbrace{(s_x/np)\backslash s_y)}_{\text{GQ}}) \vdash s_3}\,[\backslash\text{L}]$$

When instantiating the GQ with *any movie* the reading [1a] with the GQ in the scope of the negation will be derivable, whereas the other will fail since s_x will be s_1 and $s_2 \nvdash s_1$. The opposite holds when considering *some movie*: since $s_y = s_3$, the proof in [1a] fails in $s_3 \nvdash s_2$, whereas [1b] is derivable. But from the types given above it also follows that

any movie can occur in a positive context, e.g. *Coppola directed any movie*, as shown below.

$$\text{[2a]}\qquad \frac{\begin{array}{c} np \vdash np \quad np \vdash np \quad s_1 \vdash s_1 \\ \vdots \\ np, (np\backslash s_1)/np \vdash s_1/np \end{array} \qquad \boxed{s_1 \vdash s_3}}{\underbrace{np}_{\text{Coppola}}\ ,\ (\underbrace{(np\backslash s_1)/np}_{\text{directed}}, \underbrace{(s_1/np)\backslash s_1}_{\text{any movie}}) \vdash s_3}\ [\backslash\text{L}]$$

In (Areces et al., 2001), Galois connections are used to add the needed expressivity to solve this problem by enlarging the type hierarchy with a fourth type $s_4 := ({}^0s)^0$ representing the ungrammatical sentential level. NPIs carry this ungrammatical features in their type assignments, e.g. *any* $n \in (s_4/np)\backslash s_4$. This type will block the occurrences of *any* n in positive contexts like [2a]. Similarly, *any* n will not be able to take wide scope over negation as the reader can check by instantiating the type of the GQ in [1b] with the one of *any* n. However, in order to be licensed by the negative auxiliary *didn't* and occur in its scope, *didn't* must be assigned a type compatible with the NPI, viz. *didn't* $\in (np\backslash s_2)/(np\backslash s_4)$. Since $s_2 \vdash s_4$, the new type assignment derives the one we have previously assigned to *didn't* and the derivations given above can still be proved.

In (Bernardi, 2002), the richer structure of derivability relations provided by the addition of Galois connections is exploited to describe a classification of NPIs. Essentially, this analysis is based on the fact that a function of type A/B composes with an expression of any type C such that $C \vdash B$. The main novelty introduced by Galois connections is the possibility of *reversing* the derivability relation between types. Schematically, a function of type $A/(C)^0$ composes with any expression of type $(B)^0$ such that $C \vdash B$. These two composition patterns are summarized by the inference schemata below, where $[\cdot]$ and $\{\cdot\}$ are positive and negative contexts, respectively (see (Bernardi, 2002) for details).

$$\frac{\Delta[B] \vdash A}{\Delta[C] \vdash A} \quad \text{and} \quad \frac{\Delta\{C\} \vdash A}{\Delta\{B\} \vdash A} \text{ for } C \vdash B$$

The two inferences can be used to model licensing and antilicensing relations. The first pattern is used to model an item which must be in a context satisfying certain property to be grammatical. The second pattern models an item that is allergic to a certain property and therefore cannot occur within contexts having such property. A concrete example is given bye the syntactic distribution of Dutch negative and positive polarity item with respect to downward monotone functions (van der

Wouden, 1994). Consider the following data, where the underlined words are the polarity items.

1a. *Niemand* zal ook maar iets bereiken.
(tr. Nobody will achieve anything.)

1b. *Niemand* hoeft te fietsen.
(tr. Nobody has to bike.)

2a. *Weinig* monniken zijn een beetje gelukkig.
(tr. Few monks are a bit happy.)

2b. *Weinig* kinderen wil nog Donne lezen.
(tr. Few children still want to read Donne.)

Let A/B and B be the types of the function *niemand* and of the NPI *ook maar iets* in its scope. A NPI is said to be 'weaker' than *ook maar iets* if it is felicitous also in 'less negative' contexts, e.g. *hoeft* which is grammatical also when composed with *zelden* (tr. seldom). This relation among these two kinds of NPIs can be modeled by assigning to *hoeft* a weaker type than the one of *ook maar ites*: $C \vdash B$. The function *niemand* composes with any NPI weaker than *ook maar iets* and in particular with *hoeft*, as predicted by the types.

Conversely, a property of a function like *weinig* is compatible with a positive polarity item like *een beetje* and consequently is compatible also with a weaker one, like *nog* which is allergic to stronger properties than *een beetje*. The relation among the PPIs forces the type assigned to *nog* to derive the one assigned to *een beetje*: $C \vdash B$. But this will require the function *weinig* to be of type $A/(C)^0$: it composes with any type stronger than C.

5. Conclusion

As we said in the introduction, the main aim of this paper is to provide new insight on categorial type logics by the use of display calculi. The logical systems encoding residuation we discussed (NL and $\mathsf{NL}(\Diamond)$) are well known in the field, and their meta-logical and proof-theoretical properties have been established long ago. Still, we feel that DC do provide further insight on how these systems *came to be*, and in which sense they indeed encode in a "pure" state important algebraic properties like residuation and Galois connections. In our analysis, we directly used systems introduced by Goré in (1998), our main contribution in the first part of the paper is puzzling out how these systems relate to those standard in the categorial grammar community. Building on our work in Section 2.2, we move on to define the system NL $(\Diamond, {}^0(\cdot))$,

having both unary residuated and Galois connected operators. In other words we define a system with operators that resemble a pair of split negations as primitives. Finally, in Section 4, we provide some ideas on how the systems can be used in analyzing linguistic phenomena.

References

Abrusci, M.: 1991, 'Phase semantics and sequent calculus for pure noncommutative classical linear propositional logic'. *The Journal of Symbolic Logic* **56**(4), 1403–1451.

Ajdukiewicz, K.: 1935, 'Die Syntaktische Konnexität'. *Studia Philosophica* **1**, 1–27. (English translation in Storrs McCall (ed.) Polish Logic, 1920-1939. Oxford (1996), 207–231).

Areces, C., R. Bernardi, and M. Moortgat: 2001, 'Galois connections in categorial type logic'. In: R. Oherle and L. Moss (eds.): *Proceedings of FGMOL'01*. Special issue of the *Electronic Notes in Theoretical Computer Science*, Volume 53.

Bar-Hillel, Y.: 1953, 'A quasi-arithmetical notation for syntactic description'. *Language* **29**, 47–58.

Belnap, N.: 1982, 'Display logic'. *Journal of Philosophical Logic* **11**(4), 375–417.

Bernardi, R.: 2002, 'Reasoning with Polarity in Categorial Type Logic'. Ph.D. thesis, UiL OTS, University of Utrecht.

Bernardi, R. and R. Moot: 2002, 'Scope Ambiguities from a Proof-theoretical Perspective'. In: J. Bos and M. Kohlhase (eds.): *Proceedings of ICoS-2*. To appear as special issue of the *Journal of Language and Computation*.

Birkhoff, G.: (1940, 1948, 1967), *Lattice Theory*. Providence: American Mathematical Society.

Blyth, T. and F. Janowitz: 1872, *Residuation Theory*. New York: Pergamon Press Inc.

Dunn, J.: 1991, 'Gaggle theory: an abstraction of Galois connections and residuation with applications to negation and various logical operations'. In: *JELIA 1990: Proceedings of the European Workshop on Logics in Artificial Intelligence*, Vol. LNCS 478. Springer.

Fuchs, L.: 1963, *Partially-ordered algebraic systems*. New York: Pergamon Press Inc.

Goré, R.: 1998, 'Gaggles, Gentzen and Galois: How to display your favourite substructural logic.'. *Logic Journal of the IGPL* **6**(5), 669–694.

Goré, R.: 1998, 'Substructural logics on display'. *Logic Journal of the IGPL* **6**(3), 451–504.

Lambek, J.: 1958, 'The mathematics of sentence structure'. *American Mathematical Monthly* **65**, 154–170.

Lambek, J.: 1961, 'On the calculus of syntactic types'. In: R. Jakobson (ed.): *Structure of Languages and Its Mathematical Aspects*. American Mathematical Society, pp. 166–178.

Lambek, J.: 1993, 'From categorial to bilinear logic'. In: K. D. P. Schröder-Heister (ed.): *Substructural Logics*. Oxford University Press, pp. 207–237.

Lambek, J.: 2001, 'Type grammars as pregroups'. *Grammars* **4**(1), 21–39.

Montague, R.: 1974, *Formal Philosophy: Selected Papers of Richard Montague*. New Haven: Yale University Press.

Moortgat, M.: 1996, 'Multimodal linguistic inference'. *Journal of Logic, Language and Information* **5**(3,4), 349–385.

Moortgat, M.: 1997, 'Categorial Type Logics'. In: J. van Benthem and A. ter Meulen (eds.): *Handbook of Logic and Language*. Cambridge: The MIT Press, Cambridge, Massachusetts, pp. 93–178.

Partee, B. and M. Rooth: 1983, 'Generalized conjunction and type ambiguity'. In: B. et al. (ed.): *Meaning, Use, and Interpretation of Language*. Berlin: De Gruyter, pp. 361–383.

Restall, G.: 2000, *An introduction to substructural logics*. Routledge.

van der Wouden, T.: 1994, 'Negative Contexts'. Ph.D. thesis, University of Groningen.