Symmetries in Natural Language Syntax and Semantics: The Lambek-Grishin Calculus

Michael Moortgat*

Utrecht Institute of Linguistics OTS, The Netherlands
moortgat@let.uu.nl

Abstract. In this paper, we explore the Lambek-Grishin calculus **LG**: a symmetric version of categorial grammar based on the generalizations of Lambek calculus studied in Grishin [1]. The vocabulary of **LG** complements the Lambek product and its left and right residuals with a dual family of type-forming operations: coproduct, left and right difference. The two families interact by means of structure-preserving distributivity principles. We present an axiomatization of **LG** in the style of Curry's combinatory logic and establish its decidability. We discuss Kripke models and Curry-Howard interpretation for **LG** and characterize its notion of type similarity in comparison with the other categorial systems. From the linguistic point of view, we show that **LG** naturally accommodates non-local semantic construal and displacement — phenomena that are problematic for the original Lambek calculi.

1 Background

The basic Lambek calculus [2] is a logic without *any* structural rules: grammatical material cannot be duplicated or erased without affecting well-formedness (absence of Contraction and Weakening); moreover, structural rules affecting word order and constituent structure (Commutativity and Associativity) are unavailable. What remains (in addition to the preorder axioms for derivability) is the pure logic of residuation of (1).

$$\text{RESIDUATED TRIPLE} \qquad A \to C/B \quad \text{iff} \quad A \otimes B \to C \quad \text{iff} \quad B \to A\backslash C \tag{1}$$

The type-forming operations have two kinds of semantics. One is a *structural* semantics, where they are interpreted with respect to a ternary composition relation (or 'Merge', as it is called in generative grammar). The truth conditions for

* Linguistic exploration of Lambek-Grishin calculus started at ESSLLI'04 (Nancy) in discussion with Raffaella Bernardi and Rajeev Goré; I thank them, and Natasha Kurtonina, for the stimulating exchange of ideas throughout the period reported on here. The material of this paper was the subject of a series of lectures at Moscow State University (Sept 2006, April 2007). I thank Mati Pentus and Barbara Partee for accommodating these talks in their seminars and for providing such perceptive audiences. The support of the Dutch-Russian cooperation program NWO/RFBR, project No. 047.017.014 "Logical models of human and mechanical reasoning" is gratefully acknowledged.

D. Leivant and R. de Queiroz (Eds.): WoLLIC 2007, LNCS 4576, pp. 264–284, 2007.

this interpretation are given in (2); one finds the basic soundness/completeness results in [3]. The second interpretation is a *computational* one, along the lines of the Curry-Howard formulas-as-types program. Under this second interpretation, Lambek derivations are associated with a linear (and structure-sensitive) sublanguage of the lambda terms one obtains for proofs in positive Intuitionistic logic. The slashes $/, \backslash$ here are seen as directional implications; elimination of these operations corresponds to function application, introduction to lambda abstraction.

$$\begin{array}{l} x \Vdash A \otimes B \text{ iff } \exists yz.R_{\otimes}xyz \text{ and } y \Vdash A \text{ and } z \Vdash B \\ y \Vdash C/B \text{ iff } \forall xz.(R_{\otimes}xyz \text{ and } z \Vdash B) \text{ implies } x \Vdash C \\ z \Vdash A\backslash C \text{ iff } \forall xy.(R_{\otimes}xyz \text{ and } y \Vdash A) \text{ implies } x \Vdash C \end{array} \tag{2}$$

The original Lambek calculus, like its predecessors the Ajdukiewicz/Bar Hillel (AB) calculi, and later systems such as Combinatory Categorial Grammar (CCG), adequately deals with linguistic subcategorization or valency. It greatly improves on AB and CCG systems in fully supporting hypothetical reasoning: the bidirectional implications of the residuation laws are fully symmetric with respect to *putting together* larger phrases out of their subphrases, and *taking apart* compound phrases in their constituent parts. AB systems lack the second feature completely; the combinatory schemata of CCG provide only a weak approximation. Consequences of hypothetical reasoning are the theorems of type lifting and argument lowering of (3) below; type transitions of this kind have played an important role in our understanding of natural language semantics.

$$A \to B/(A\backslash B) \qquad (B/(A\backslash B))\backslash B \to A\backslash B \tag{3}$$

It is ironic that precisely in the hypothetical reasoning component the Lambek grammars turn out to be deficient. As one sees in (3), hypothetical reasoning typically involves higher order types, where a slash occurs in a negative environment as in the schema (4) below. Given Curry-Howard assumptions, the associated instruction for meaning assembly has an application, corresponding to the elimination of the main connective $/$, and an abstraction, corresponding to the introduction of the embedded $\backslash$.

$$C/(A\backslash B) \qquad (M\ \lambda x^A.N^B)^C \tag{4}$$

The minimal Lambek calculus falls short in its characterization of which A-type hypotheses are 'visible' for the slash introduction rule: for the residuation rules to be applicable, the hypothesis has to be structurally *peripheral* (left peripheral for $\backslash$, right peripheral for $/$). One can distinguish two kinds of problems.

Displacement. The A-type hypothesis occurs *internally* within the domain of type B. Metaphorically, the functor $C/(A\backslash B)$ seems to be displaced from the site of the hypothesis. Example: *wh* 'movement'.

Non-local semantic construal. The functor (e.g. $C/(A\backslash B)$) occupies the structural position where in fact the A-type hypothesis is needed, and realizes its semantic effect at a higher structural level. (The converse of the above.) Example: quantifying expressions.

The initial reaction to these problems has been to extend the basic system with structural rules of Associativity and/or Commutativity, resulting in the hierarchy of Fig 1. From a linguistic point of view, a *global* implementation of these structural options is undesirable, as it entails a complete loss of sensitivity for word order and/or constituent structure.

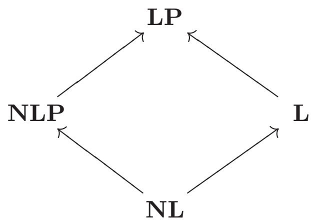


Fig. 1. The Lambek hierarchy

A significant enhancement of the Lambek systems occurred in the mid Nineties of the last century, when the vocabulary was extended with a pair of *unary* type-forming operations (Moortgat [4]). Like their binary relatives, $\Diamond, \Box$ form a residuated pair, satisfying (5), and with interpretation (6).

$$\text{RESIDUATED PAIR} \qquad \Diamond A \rightarrow B \quad \text{iff} \quad A \rightarrow \Box B \tag{5}$$

$$\begin{array}{l} x \Vdash \Diamond A \text{ iff } \exists y\ (R_\Diamond xy \text{ and } y \Vdash A) \\ y \Vdash \Box A \text{ iff } \forall x\ (R_\Diamond xy \text{ implies } x \Vdash A) \end{array} \tag{6}$$

From a purely logic point of view, the unary modalities introduce facilities for *subtyping*, in the sense that any type A is now derivationally related to a more specific type $\Diamond\Box A$ and a more general type $\Box\Diamond A$.

$$\Diamond\Box A \rightarrow A \rightarrow \Box\Diamond A \tag{7}$$

Bernardi [5] makes good use of the patterns in (7) to fine-tune the construal of scope-bearing expressions, and to capture the selectional restrictions governing the distribution of polarity sensitive items. In addition to this logical use of $\Diamond, \Box$, one can also use them to provide *controlled* versions of structural rules that, in a global form, would be deleterious. The set of postulates in (8) make left branches ($P1$, $P2$) or right branches ($P3$, $P4$) accessible for hypothetical reasoning.

$$\begin{array}{llll} (P1) & \Diamond A \otimes (B \otimes C) \rightarrow (\Diamond A \otimes B) \otimes C & (C \otimes B) \otimes \Diamond A \rightarrow C \otimes (B \otimes \Diamond A) & (P3) \\ (P2) & \Diamond A \otimes (B \otimes C) \rightarrow B \otimes (\Diamond A \otimes C) & (C \otimes B) \otimes \Diamond A \rightarrow (C \otimes \Diamond A) \otimes B & (P4) \end{array} \tag{8}$$

Vermaat [6] uses these postulates in a cross-linguistic study of *wh* extraction constructions, relating the choice between $P1$, $P2$ and $P3$, $P4$ to the typological distinction between head-initial versus head-final languages.

Structural control. The general situation with respect to the expressivity of modally controlled structural rules is captured by Thm 1. We consider a source logic $\mathcal{L}$ and a target logic, which is either an upward ($\mathcal{L}^{\uparrow}$) or a downward ($\mathcal{L}^{\downarrow}$) neighbor in the Lambek hierarchy. The target logic has control modalities $\Diamond, \Box$ which are lacking in the source logic. In terms of these modalities, one defines translations from the formulas of the source logic to the formulas of the target logic. The $\cdot^{\downarrow}$ translations impose the structure-sensitivity of the source logic in a logic with a more liberal structural regime; the $\cdot^{\uparrow}$ translations recover the flexibility of an upward neighbor by adding $\mathcal{R}_{\Diamond}$ — the image under $\cdot^{\uparrow}$ of the structural rules that discriminate source from target. An example is the translation $(A \otimes B)^{\downarrow} = \Diamond(A^{\downarrow} \otimes B^{\downarrow})$, which blocks associativity by removing the structural conditions for its application.

Theorem 1. *Structural control (Kurtonina and Moortgat [7]). For logics $\mathcal{L}$, $\mathcal{L}^{\uparrow}$, $\mathcal{L}^{\downarrow}$ and translations $\cdot^{\downarrow}$ and $\cdot^{\uparrow}$ as defined above,*

$$\begin{array}{llrl}
\mathcal{L} \vdash A \to B & \textit{iff} & \mathcal{L}^{\uparrow}_{\Diamond} \vdash A^{\downarrow} \to B^{\downarrow} & \textsc{constraining} \\
\mathcal{L} \vdash A \to B & \textit{iff} & \mathcal{L}^{\downarrow}_{\Diamond} + \mathcal{R}_{\Diamond} \vdash A^{\uparrow} \to B^{\uparrow} & \textsc{licensing}
\end{array}$$

Summarizing. Viewed from a foundational point of view, one can see Lambek-style categorial grammars as modal logics of natural language resources. In such logics, the vocabulary for analyzing the assembly of form and meaning consists of n-ary type-forming operations (or connectives, under Curry's formulas-as-types view); these operations are given a Kripke-style interpretation in terms of $(n+1)$-ary 'merge'/composition relations. Grammatical *invariants*, in this approach, are laws that do not impose restrictions on the interpreting composition relations; language *diversity* results from the combination of the invariant base logic with a set of non-logical axioms (and the corresponding frame constraints). These axioms (possibly language-specific) characterize the structural deformations under which the basic form-meaning correspondences are preserved. The reader is referred to [8] for an overview of this line of work.

2 Lambek-Grishin Calculus

Assessing the merits of the above approach, one can identify two problematic aspects. One is of a computational nature, the other relates to the cognitive implications of the Lambek framework.

Complexity. Structural rules (whether implemented globally or under modal control) are computationally expensive. Whereas the basic Lambek calculus has a polynomial recognition problem [9], already the simplest extension with an associative regime is known to be NP complete [10]; one reaches a PSPACE upper bound for the extension with a $\Diamond, \Box$ controlled structural module consisting of resource-respecting (i.e. linear, non-expanding) axioms [11].

Invariants versus structural postulates. On the cognitive level, the limited expressivity of the standard vocabulary means one is forced to accept that a considerable part of grammatical organization is beyond the reach of the type-forming constants. By considering a broader vocabulary of connectives it becomes possible to characterize larger portions of a language's grammar in terms of linguistic *invariants*, rather than through non-logical postulates.

In a remarkable paper written in 1983, V.N. Grishin [1] has proposed a framework for generalizing the Lambek calculi that provides an alternative to the structural rule approach. The starting point for Grishin's generalization is a symmetric extension of the vocabulary of type-forming operations: in addition to the familiar $\otimes, \backslash, /$ (product, left and right division), one also considers a dual family $\oplus, \oslash, \obslash$: coproduct, right and left difference.[1]

$$\begin{array}{lll} A, B ::= p \mid & & \text{atoms: } s \text{ sentence, } np \text{ noun phrases, } \ldots \\ & A \otimes B \mid B \backslash A \mid A / B \mid & \text{product, left vs right division} \\ & A \oplus B \mid A \oslash B \mid B \obslash A & \text{coproduct, right vs left difference} \end{array} \tag{9}$$

We saw that algebraically, the Lambek operators form a residuated triple; likewise, the $\oplus$ family forms a dual residuated triple.

$$\begin{array}{lccccc} \textsc{residuated triple} & A \to C/B & \text{iff} & A \otimes B \to C & \text{iff} & B \to A \backslash C \\ \textsc{dual residuated triple} & C \oslash B \to A & \text{iff} & C \to A \oplus B & \text{iff} & A \obslash C \to B \end{array} \tag{10}$$

Dunn's [12] framework of gaggle theory brings out the underlying algebraic structure in a particularly clear way. In Fig 2, we consider ordered sets $(X, \leq)$, $(Y, \leq')$ with mappings $f : X \longrightarrow Y$, $g : Y \longrightarrow X$. In the categorial setting, we have $X = Y = \mathcal{F}$ (the set of types/formulas). The pair of operations (f, g) is called residuated if it satisfies the defining biconditionals of the upper left cell; the lower right cell characterizes dual residuated pairs. Whereas the concept of residuation pairs the (co)product with a (co)implication, the closely related concept of (dual) Galois connected pairs links the (co)implications among themselves. The defining biconditionals fill the lower left and upper right cells. For $* \in \{/, \otimes, \backslash, \obslash, \oplus, \oslash\}$, we write $-*$ $(*-)$ for the operation that suffixes (prefixes) a fixed type to its operand, for example: $A \to C/B$ iff $B \to A\backslash C$ instantiates the pattern $(/-, -\backslash)$.

iff	$x \leq gy$	$gy \leq x$
$fx \leq' y$	$(-\otimes, -/)$ $(\otimes-, \backslash-)$	$(-\obslash, \oslash-)$ $(\oslash-, -\obslash)$
$y \leq' fx$	$(-\backslash, /-)$ $(/-, -\backslash)$	$(-\oplus, -\oslash)$ $(\oplus-, \obslash-)$

Fig. 2. Residuated and Galois connected pairs and their duals

[1] A little pronunciation dictionary: read $B\backslash A$ as 'B under A', A/B as 'A over B', $B \obslash A$ as 'B from A' and $A \oslash B$ as 'A less B'.

The patterns of Fig 2 reveal that on the level of types and derivability the Lambek-Grishin system exhibits two kinds of mirror symmetry characterized by the bidirectional translation tables in (11): $\bowtie$ is order-preserving, ∞ order-reversing: $A^{\bowtie} \to B^{\bowtie}$ iff $A \to B$ iff $B^{\infty} \to A^{\infty}$.

$$\bowtie \frac{C/D \quad A \otimes B \quad B \oplus A \quad D \oslash C}{\overline{D \backslash C \quad B \otimes A \quad A \oplus B \quad C \oslash D}} \qquad \infty \frac{C/B \quad A \otimes B \quad A \backslash C}{\overline{B \oslash C \quad B \oplus A \quad C \oslash A}} \tag{11}$$

Interaction principles. The minimal symmetric categorial grammar (which we will refer to as $\mathbf{LG}_{\emptyset}$) is given by the preorder axioms for the derivability relation, together with the residuation and dual residuation principles of (10). $\mathbf{LG}_{\emptyset}$ by itself does not offer us the kind of expressivity needed to address the problems discussed in §1. The real attraction of Grishin's work derives from the *interaction principles* he proposes for structure-preserving communication between the $\otimes$ and the $\oplus$ families. In all, the type system allows eight such principles, configured in two groups of four. Consider first the group in (12) which we will collectively refer to as $\mathcal{G}^{\uparrow}$.[2]

$$\begin{array}{llll} (G1) & (A \oslash B) \otimes C \to A \oslash (B \otimes C) & C \otimes (B \oslash A) \to (C \otimes B) \oslash A & (G3) \\ (G2) & C \otimes (A \oslash B) \to A \oslash (C \otimes B) & (B \oslash A) \otimes C \to (B \otimes C) \oslash A & (G4) \end{array} \tag{12}$$

On the lefthand side of the derivability arrow, one finds a $\otimes$ formula which has a formula with the difference operation ($A \oslash B$ or $B \oslash A$) in its first or second coordinate. The Grishin principles rewrite this configuration in such a way that the difference operations $\oslash$, $\oslash$ become the main connective. Combined with transitivity, the principles take the form of the inference rules in (13).

$$\begin{array}{ll} \dfrac{A \oslash (B \otimes C) \to D}{(A \oslash B) \otimes C \to D}\ G1 & \dfrac{(A \otimes B) \oslash C \to D}{A \otimes (B \oslash C) \to D}\ G3 \\[2ex] \dfrac{B \oslash (A \otimes C) \to D}{A \otimes (B \oslash C) \to D}\ G2 & \dfrac{(A \otimes C) \oslash B \to D}{(A \oslash B) \otimes C \to D}\ G4 \end{array} \tag{13}$$

These rules, from a backward-chaining perspective, have the effect of bringing the A subformula to a position where it can be shifted to the righthand side of $\to$ by means of the dual residuation principles. The images of (12) under $\cdot^{\infty}$ are given in (14): in rule form, they rewrite a configuration where a left or right slash is trapped within a $\oplus$ context into a configuration where the A subformula can be shifted to the lefthand side of $\to$ by means of the residuation principles. One easily checks that the forms in (14) are derivable from (12) — the derivation of $G1'$ from $G1$ is given as an example in (19).

$$\begin{array}{llll} (G1') & (C \oplus B)/A \to C \oplus (B/A) & A \backslash (B \oplus C) \to (A \backslash B) \oplus C & (G3') \\ (G2') & (B \oplus C)/A \to (B/A) \oplus C & A \backslash (C \oplus B) \to C \oplus (A \backslash B) & (G4') \end{array} \tag{14}$$

[2] In Grishin's original paper, only one representative of each group is discussed; computation of the remaining three is left to the reader. Earlier presentations of [1] such as [13,14] omit $G2$ and $G4$. The full set of (12) is essential for the intended linguistic applications.

An alternative direction for generalizing the Lambek calculus is given by the *converses* of the $\mathcal{G}^{\uparrow}$ principles obtained by turning around the derivability arrow. We refer to these individually as Gn^{-1}, and to the group as $\mathcal{G}^{\downarrow}$. The general picture that emerges is a landscape where the minimal symmetric Lambek calculus $\mathbf{LG}_{\emptyset}$ can be extended either with $G1$–$G4$ or with their converses, or with the combination of the two. We discuss potential linguistic applications for $\mathcal{G}^{\uparrow}$ and $\mathcal{G}^{\downarrow}$ in §3. First, we review some prooftheoretic and modeltheoretic results relating the Lambek-Grishin calculus to the original Lambek systems. These results have focused on the combination $\mathbf{LG}_{\emptyset} + \mathcal{G}^{\uparrow}$, which in the remainder we will refer to simply as $\mathbf{LG}$.

2.1 Decidable Proof Search

The axiomatization we have considered so far contains the rule of transitivity (from $A \to B$ and $B \to C$ conclude $A \to C$) which, in the presence of complexity-increasing type transitions, is an unpleasant rule from a proof search perspective. For decidable proof search, we are interested in an axiomatization which has transitivity as an admissible rule. Such an axiomatization for $\mathbf{LG}$ can be given in terms of the identity axiom $1_A : A \to A$ together with the residuation principles (10), the Grishin axioms in rule form (16), and the monotonicity rules of (17).[3] We give these rules with combinator proof terms, so that in the remainder we can succinctly refer to derivations by their combinator. First the residuation rules of (15).

$$\frac{f : A \otimes B \to C}{\triangleleft f : B \to A \backslash C} \quad \frac{f : C \to A \oplus B}{\blacktriangleleft f : A \obslash C \to B}$$
$$\frac{f : A \otimes B \to C}{\triangleright f : A \to C / B} \quad \frac{f : C \to A \oplus B}{\blacktriangleright f : C \oslash B \to A} \tag{15}$$

These rules are invertible; we write $\triangleleft', \triangleright', \blacktriangleleft', \blacktriangleright'$ for the reverse direction. Next the Grishin axioms in rule form, for use in the lhs of derivability statements.

$$\frac{f : A \obslash (B \otimes C) \to D}{\olessthan f : (A \obslash B) \otimes C \to D} \quad \frac{f : (A \otimes B) \oslash C \to D}{\ogreaterthan f : A \otimes (B \oslash C) \to D}$$
$$\frac{f : B \obslash (A \otimes C) \to D}{\olessthan^{\star} f : A \otimes (B \obslash C) \to D} \quad \frac{f : (A \otimes C) \oslash B \to D}{\ogreaterthan^{\star} f : (A \oslash B) \otimes C \to D} \tag{16}$$

Finally, (17) gives the monotonicity rules. As is well-known, the monotonicity rules are *derivable* rules of inference in an axiomatization with residuation and transitivity (cut). The purpose of the present axiomatization is to show that the

[3] The axiomatization presented here is a close relative of Display Logic, see [14] for a comprehensive view on the substructural landscape. In Display Logic, the Grishin rules and residuation principles are expressed at the *structural* level; structural connectives are introduced by explicit rewriting steps. Our combinator presentation is entirely formula-based, i.e. the distinction between a 'logical' and a 'structural' occurrence of a type-forming operation is implicit.

combination monotonicity plus residuation effectively *absorbs* cut. Admissibility of cut is established in Appendix A, extending the earlier result of [15] to the case of symmetric **LG**.

$$\frac{f : A \to B \quad g : C \to D}{f \otimes g : A \otimes C \to B \otimes D} \quad \frac{f : A \to B \quad g : C \to D}{f \oplus g : A \oplus C \to B \oplus D}$$
$$\frac{f : A \to B \quad g : C \to D}{f/g : A/D \to B/C} \quad \frac{f : A \to B \quad g : C \to D}{f \oslash g : A \oslash D \to B \oslash C} \tag{17}$$
$$\frac{f : A \to B \quad g : C \to D}{g \backslash f : D \backslash A \to C \backslash B} \quad \frac{f : A \to B \quad g : C \to D}{g \obslash f : D \obslash A \to C \obslash B}$$

The symmetries we have studied before on the level of types and theorems now manifest themselves on the level of proofs.

$$\infty \; \frac{h/g \quad f \otimes g \quad f \backslash h}{\overline{g \obslash h \quad g \oplus f \quad h \oslash f}} \quad \infty \; \frac{\lhd f \quad \rhd f \quad \lhd' f \quad \rhd' f}{\overline{\blacktriangleright f \quad \blacktriangleleft f \quad \blacktriangleright' f \quad \blacktriangleleft' f}} \tag{18}$$

As an example, consider (19). On the left, we derive $G1' = G1^{\infty}$ from $G1$; on the right we derive $G1$ from $G1'$. Notice that these derivations provide the motivation for our choice to posit ∞ as the basic order-reversing duality, rather than $\natural$ which we define as $\bowtie \infty$.

$$\begin{array}{cl}
(c \oplus b)/a \to (c \oplus b)/a & \\ \cline{1-1}
((c \oplus b)/a) \otimes a \to c \oplus b & \rhd' \\ \cline{1-1}
c \obslash (((c \oplus b)/a) \otimes a) \to b & \blacktriangleleft \\ \cline{1-1}
(c \obslash ((c \oplus b)/a)) \otimes a \to b & \olessthan \\ \cline{1-1}
c \obslash ((c \oplus b)/a) \to b/a & \rhd \\ \cline{1-1}
(c \oplus b)/a \to c \oplus (b/a) & \blacktriangleleft'
\end{array}
\qquad
\begin{array}{cl}
a \obslash (b \otimes c) \to a \obslash (b \otimes c) & \\ \cline{1-1}
b \otimes c \to a \oplus ((a \obslash (b \otimes c)) & \blacktriangleleft' \\ \cline{1-1}
b \to (a \oplus ((a \obslash (b \otimes c)))/c & \rhd \\ \cline{1-1}
b \to a \oplus ((a \obslash (b \otimes c))/c) & \olessthan^{\infty} \\ \cline{1-1}
a \obslash b \to (a \obslash (b \otimes c))/c & \blacktriangleleft \\ \cline{1-1}
(a \obslash b) \otimes c \to a \obslash (b \otimes c) & \rhd'
\end{array} \tag{19}$$

$$(\blacktriangleleft' \rhd \olessthan \blacktriangleleft \rhd' \; 1_{(c \oplus b)/a})^{\infty} = \rhd' \blacktriangleleft \olessthan^{\infty} \rhd \blacktriangleleft' \; 1_{a \obslash (b \otimes c)}$$

We close this section with an open question on complexity. The minimal symmetric system $\mathbf{LG}_{\emptyset}$ is proved to preserve the polynomiality of the asymmetric **NL** in [16]. Capelletti [17] provides a constructive polynomial algorithm for a combinator-style axiomatization of $\mathbf{LG}_{\emptyset}$ that allows agenda-driven chart-based parsing. Whether the methods developed in [17] can be extended to include the Grishin interaction principles remains to be investigated.

2.2 The Group of Grishin Interactions

The $\bowtie$ and ∞ dualities deserve closer scrutiny: as a matter of fact, they hide some interesting grouptheoretic structure.[4] First of all, from $\bowtie$ and the identity

[4] This section reports on work in progress with Lutz Strassburger (Ecole Polytechnique, Paris) and with Anna Chernilovskaya (Moscow State University).

transformation 1 we obtain two further order-preserving symmetries $\sharp$ and $\flat$: on the $\otimes$ family $\sharp$ acts like $\bowtie$, on the $\oplus$ family it is the identity; $\flat$ acts like $\bowtie$ on the $\oplus$ family, and as the identity on the $\otimes$ family.

	$\sharp$	$\flat$
$\mathrm{Frm}(/, \otimes, \backslash)$	$\bowtie$	1
$\mathrm{Frm}(\obslash, \oplus, \oslash)$	1	$\bowtie$

(20)

One easily checks that the Grishin postulates (12) are related horizontally by $\bowtie$, vertically by $\sharp$ and diagonally by $\flat$. Together with the identity transformation, $\bowtie$, $\sharp$ and $\flat$ constitute D_2 — the dihedral group of order 4 (also known as the Klein group). This is the smallest non-cyclic abelian group. Its Cayley table is given in (21) below.

$\circ$	1	$\bowtie$	$\sharp$	$\flat$
1	1	$\bowtie$	$\sharp$	$\flat$
$\bowtie$	$\bowtie$	1	$\flat$	$\sharp$
$\sharp$	$\sharp$	$\flat$	1	$\bowtie$
$\flat$	$\flat$	$\sharp$	$\bowtie$	1

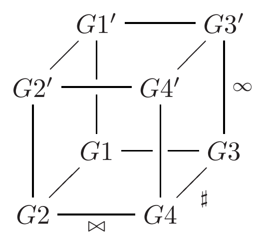
(21)

Similarly, from ∞ we obtain three further order-reversing symmetries: $\natural = \bowtie \infty$, and the mixed forms $\dagger$ and $\ddagger$ in (22) below.

	$\dagger$	$\ddagger$
$\mathrm{Frm}(/, \otimes, \backslash)$	$\natural$	∞
$\mathrm{Frm}(\obslash, \oplus, \oslash)$	∞	$\natural$

(22)

Together with the order-preserving transformations $\{1, \bowtie, \sharp, \flat\}$, the set $\{\infty, \natural, \dagger, \ddagger\}$ constitutes a group of order 8. We can now consider a *cube* of Grishin interactions.



(23)

The lower plane is the square (12) which we saw is characterized by D_2. The transformation ∞ reflects (12) to the upper plane (14). The remaining transformations connect vertices of the lower plane to those of the upper plane via the diagonals. It will not come as a surprise that we have D_4, which contains $2 \times D_2$ as subgroups: $\{1, \bowtie, \sharp, \flat\}$ and $\{1, \infty, \natural, \bowtie\}$. Notice that D_4, unlike its D_2 subgroups, is not abelian. Consider for example $\dagger\infty = \flat \neq \sharp = \infty\dagger$. The composition $\dagger\infty$ relates $G1$ to $G4$ via $G1'$; the composition $\infty\dagger$ brings us from $G1$ to $G2$ via $G2'$.

2.3 Type Similarity

The notion of type similarity is a powerful tool to study the expressivity of categorial logics with respect to derivational relationships. The similarity relation $\sim$ is introduced in the algebraic remarks at the end of [18] as the reflexive, transitive, symmetric closure of the derivability relation: $A \sim B$ iff there exists a sequence $C_1 \ldots C_n$ $(1 \leq n)$ such that $C_1 = A$, $C_n = B$ and $C_i \rightarrow C_{i+1}$ or $C_{i+1} \rightarrow C_i$ $(1 \leq i < n)$. Lambek proves that $A \sim B$ if and only if one of the following equivalent statements hold (the so-called diamond property): (i) $\exists C$ such that $A \rightarrow C$ and $B \rightarrow C$; (ii) $\exists D$ such that $D \rightarrow A$ and $D \rightarrow B$. In other words, given a join type C for A and B, one can compute a meet type D, and vice versa. The solutions for D and C in [18] are given in (24). It is shown in [19] that these solutions are in fact adequate for the pure logic of residuation, i.e. the non-associative calculus **NL**.

$$\mathbf{NL} : D = (A/((C/C)\backslash C)) \otimes ((C/C)\backslash B),\ C = (A \otimes (D\backslash D))/(B\backslash(D \otimes (D\backslash D))) \tag{24}$$

For associative **L**, [20] has the shorter solution in (25). The possibility of rebracketing the types for D and C is what makes this solution work. In **LG** also a length 5 solution is available, this time dependent on the Grishin interaction principles, see (26).

$$\mathbf{L} : D = (A/C) \otimes (C \otimes (C\backslash B)), \quad C = (D/A)\backslash(D/(B\backslash D)) \tag{25}$$

$$\mathbf{LG} : D = (A/C) \otimes (C \oslash (B \obslash C)), \quad C = (A \oslash D) \oplus (D/(B\backslash D)) \tag{26}$$

Table 1. Models for type equivalence

CALCULUS	INTERPRETATION
NL	free quasigroup (Foret [19])
L	free group (Pentus [20])
LP	free Abelian group (Pentus [20])
LG	free Abelian group (Moortgat and Pentus [21])

The similarity relation for various calculi in the categorial hierarchy has been characterized in terms of an algebraic interpretation of the types $[\![\cdot]\!]$, in the sense that $A \sim B$ iff $[\![A]\!] =_{\mathcal{S}} [\![B]\!]$ in the relevant algebraic structures $\mathcal{S}$. Table 1 gives an overview of the results. For the pure residuation logic **NL**, $\mathcal{S}$ is the free quasigroup generated by the atomic types, with $[\![\cdot]\!]$ defined in the obvious way: $[\![p]\!] = p$, $[\![A/B]\!] = [\![A]\!]/[\![B]\!]$, $[\![B\backslash A]\!] = [\![B]\!]\backslash[\![A]\!]$, $[\![A \otimes B]\!] = [\![A]\!] \cdot [\![B]\!]$.[5] In associative **L**, type similarity coincides with equality in the free group generated by the atomic types (free Abelian group for associative/commutative **LP**). The group interpretation is (27).

$$[\![p]\!] = p,\ [\![A \otimes B]\!] = [\![A]\!] \cdot [\![B]\!],\ [\![A/B]\!] = [\![A]\!] \cdot [\![B]\!]^{-1},\ [\![B\backslash A]\!] = [\![B]\!]^{-1} \cdot [\![A]\!] \tag{27}$$

We see in Table 1 that for the systems in the Lambek hierarchy, expressivity for $\sim$ is inversely proportional to structural discrimination: the structural rules

[5] Recall that a quasigroups is a set equipped with operations $/, \cdot, \backslash$ satisfying the equations $(x/y) \cdot y = x$, $y \cdot (y\backslash x) = x$, $(x \cdot y)/y = x$, $y\backslash(y \cdot x) = x$.

of associativity and commutativity destroy sensitivity for constituent structure and word order. The result below shows that **LG** achieves the same level of expressivity with respect to $\sim$ as the associative/commutative calculus **LP** and does so *without* inducing loss of structural discrimination.

Theorem 2. *(Moortgat and Pentus [21]) In* $\boldsymbol{LG}$ $A \sim B$ *iff* $[\![A]\!] = [\![B]\!]$ *in the free Abelian group generated by* $\boldsymbol{Atm} \cup \{\star\}$.

The group interpretation proposed in [21] for **LG** is a variant of that in (27); for **LG** one adds an extra element $\star$ to keep track of the *operator count*, defined as in (28). It is well known that Abelian group equality can be expressed in terms of a balancing count of (input/output) occurrences of literals. The operator count, together with the count of literals, is sufficient to characterize **LG** type similarity. It is in fact enough to keep track of only one of the operator counts since the equalities $|A|_\otimes = |B|_\otimes$ and $|A|_\oplus = |B|_\oplus$ are equivalent provided that $|A|_p = |B|_p$ for all p. The equality $\sum_p |A|_p - |A|_\otimes - |A|_\oplus = 1$ has an easy proof by induction on the structure of A.

$$|p|_\otimes = |p|_\oplus = 0$$

$$\begin{array}{rclrcl}
|A \otimes B|_\otimes & = & |A|_\otimes + |B|_\otimes + 1 & |A \otimes B|_\oplus & = & |A|_\oplus + |B|_\oplus \\
|A \oplus B|_\otimes & = & |A|_\otimes + |B|_\otimes & |A \oplus B|_\oplus & = & |A|_\oplus + |B|_\oplus + 1 \\
|A \,/\, B|_\otimes & = & |A|_\otimes - |B|_\otimes - 1 & |A \,/\, B|_\oplus & = & |A|_\oplus - |B|_\oplus \\
|B \backslash A|_\otimes & = & |A|_\otimes - |B|_\otimes - 1 & |B \backslash A|_\oplus & = & |A|_\oplus - |B|_\oplus \\
|A \oslash B|_\otimes & = & |A|_\otimes - |B|_\otimes & |A \oslash B|_\oplus & = & |A|_\oplus - |B|_\oplus - 1 \\
|B \obslash A|_\otimes & = & |A|_\otimes - |B|_\otimes & |B \obslash A|_\oplus & = & |A|_\oplus - |B|_\oplus - 1
\end{array} \tag{28}$$

Some examples: operator balance fails for a/b vs $a \oslash b$ (these formulas have balancing literal counts), and holds for pairs of formulas such as $(b \oslash c) \obslash a,\ c/(a\backslash b)$ and $a/b,\ (a \oslash c)/(b \oslash c)$ — pairs which are indeed in the $\sim$ relation.

We will discuss a possible use of $\sim$ in linguistic analysis in §3. Below we reproduce a step in the proof of Thm 2 which highlights the fact that **LG** has the kind of expressivity we expect for **LP**.

Claim. For arbitrary **LG** types A, B we have $B\backslash A \sim A/B$. To prove this claim, we provide a meet type, i.e. a type X such that $X \to B\backslash A$ and $X \to A/B$, which by residuation means

$$B \otimes X \to A \qquad \text{and} \qquad X \otimes B \to A$$

Let us put $X := Y \oslash Z$ and solve for

$$B \otimes (Y \oslash Z) \to A \qquad \text{and} \qquad (Y \oslash Z) \otimes B \to A$$

which by Grishin mixed associativity or commutativity follows from

$$B \otimes Y \to A \oplus Z \qquad \text{and} \qquad Y \otimes B \to A \oplus Z$$

We have a solution with $Z := (A \obslash B)$ and Y the meet for C the join of $B\backslash B$ and B/B, i.e.

$$C := ((b\backslash((b \otimes b) \oslash b)) \oplus (b/b))) \qquad \text{and} \qquad Y := ((b/b)/C) \otimes (C \oslash ((b\backslash b) \obslash C)) \,.$$

2.4 Relational Semantics

Let us turn now to the frame semantics for **LG**. We have seen in (2) that from the modal logic perspective, the binary type-forming operation $\otimes$ is interpreted as an existential modality with ternary accessibility relation $R_\otimes$. The residual / and \ operations are the corresponding universal modalities for the rotations of $R_\otimes$. For the coproduct $\oplus$ and its residuals, the dual situation obtains: $\oplus$ here is the universal modality interpreted w.r.t. an accessibility relation $R_\oplus$; the coimplications are the existential modalities for the rotations of $R_\oplus$. Notice that, in the minimal symmetric logic, $R_\oplus$ and $R_\otimes$ are *distinct* accessibility relations. Frame constraints corresponding to the Grishin interaction postulates will determine how their interpretation is related.

$$\begin{array}{l} x \Vdash A \oplus B \text{ iff } \forall yz.R_\oplus xyz \text{ implies } (y \Vdash A \text{ or } z \Vdash B) \\ y \Vdash C \oslash B \text{ iff } \exists xz.R_\oplus xyz \text{ and } z \nVdash B \text{ and } x \Vdash C \\ z \Vdash A \obslash C \text{ iff } \exists xy.R_\oplus xyz \text{ and } y \nVdash A \text{ and } x \Vdash C \end{array} \tag{29}$$

Completeness for **NL** and its extension with $\Diamond, \Box$ can be established on the basis of a canonical model construction where the worlds are simply formulas from $\mathrm{Frm}(/, \otimes, \backslash, \Diamond, \Box)$. For systems with richer vocabulary, Kurtonina [22] unfolds a systematic approach towards completeness in terms of *filter*-based canonical models. In the absence of the lattice operations[6], for **LG** we can do with the simplest filter-based construction, which construes the worlds as *weak filters*, i.e. sets of formulas closed under derivability. Let us write $\mathcal{F}_\uparrow$ for the set of filters over the **LG** formula language $\mathrm{Frm}(/, \otimes, \backslash, \obslash, \oplus, \oslash)$. The set of filters $\mathcal{F}_\uparrow$ is closed under the operations $\cdot \mathbin{\widehat{\otimes}} \cdot$, $\cdot \mathbin{\widehat{\obslash}} \cdot$ defined in (30).

$$\begin{array}{rcl} X \mathbin{\widehat{\otimes}} Y & = & \{C \mid \exists A, B\ (A \in X \text{ and } B \in Y \text{ and } A \otimes B \to C)\} \\ X \mathbin{\widehat{\obslash}} Y & = & \{B \mid \exists A, C\ (A \notin X \text{ and } C \in Y \text{ and } A \obslash C \to B\},\ \text{alternatively} \\ & = & \{B \mid \exists A, C\ (A \notin X \text{ and } C \in Y \text{ and } C \to A \oplus B\} \end{array} \tag{30}$$

To lift the type-forming operations to the corresponding operations in $\mathcal{F}_\uparrow$, let $\lfloor A \rfloor$ be the principal filter generated by A, i.e. $\lfloor A \rfloor = \{B \mid A \to B\}$ and $\lceil A \rceil$ its principal ideal, i.e. $\lceil A \rceil = \{B \mid B \to A\}$. Writing $X^\sim$ for the complement of X, we have

$$(\dagger) \quad \lfloor A \otimes B \rfloor = \lfloor A \rfloor \mathbin{\widehat{\otimes}} \lfloor B \rfloor \qquad (\ddagger) \quad \lfloor A \obslash C \rfloor = \lceil A \rceil^\sim \mathbin{\widehat{\obslash}} \lfloor C \rfloor \tag{31}$$

The equations of (31) can then be used to prove the usual truth lemma that for any formula $A \in \mathcal{F}$ and filter $X \in \mathcal{F}_\uparrow$, $X \Vdash A$ iff $A \in X$. The proof is by induction on the complexity of A. The base case is handled by the canonical valuation V^c in the model below.

[6] Allwein and Dunn [23] develop a richer theory of Kripke models for a hierarchy of substructural logics, accommodating both the lattice operations and (co)product and (co)implications.

Canonical model. Consider $\mathcal{M}^c = \langle W^c, R^c_\otimes, R^c_\oplus, V^c \rangle$ with

$W^c = \mathcal{F}_\uparrow$
$R^c_\otimes XYZ$ iff $Y \mathbin{\widehat{\otimes}} Z \subseteq X$
$R^c_\oplus XYZ$ iff $Y \mathbin{\widehat{\oslash}} X \subseteq Z$
$V^c(p) = \{X \in W^c \mid p \in X\}$

Theorem 3. *Soundness and completeness (Kurtonina and Moortgat [24]).*

$$\boldsymbol{LG}_\emptyset \vdash A \rightarrow B \textit{ iff } \models A \rightarrow B$$

For $\mathbf{LG}_\emptyset$ extended with the Grishin interaction principles, one imposes the frame constraints corresponding to the set of postulates one wishes to adopt ($\mathcal{G}^\uparrow$, $\mathcal{G}^\downarrow$ or both), and one shows that in the canonical model these constraints are satisfied. For example, for $(G1)$ we have the constraint in (32) (where $R^{(-2)}xyz = Rzyx$).

$$\forall xyzwv \;\; (R_\otimes xyz \;\wedge\; R^{(-2)}_\oplus ywv) \Rightarrow \exists t \; (R^{(-2)}_\oplus xwt \;\wedge\; R_\otimes tvz) \qquad (32)$$

Observe that the construction employed is neutral with respect to the direction of the Grishin interaction principles: it accommodates $G1$–$G4$ and the converse $G1^{-1}$–$G4^{-1}$ in an entirely similar way. Further research would have to show whether more concrete models exist with a bias towards either $G1$–$G4$ or the converse principles, and whether one could relate these models to the distinction between 'overt' and 'covert' forms of displacement which we illustrate in §3.

2.5 Computational Semantics

The second type of semantics we want to consider is the Curry-Howard interpretation of **LG** derivations. In Bernardi and Moortgat [25], one finds a continuation semantics based on (a directional refinement of) Curien and Herbelin's [26] work on the $\lambda\mu$ calculus. The source language for this interpretation is a term language coding **LG** sequent proofs. The sequent presentation of **LG** presented in [25] essentially relies on the Cut rule, which cannot be eliminated without losing completeness. In the present section, we give a semantics in the continuation-passing style (CPS) for the axiomatization of §2.1 which, as we have demonstrated, has an admissible cut rule.

Functions, under the CPS interpretation, rather than simply returning a value as in a direct interpretation, are provided with an extra argument for the continuation of the computation. We distinguish a call-by-value (cbv) and a call-by-name (cbn) interpretation regime. On the type level, the call-by-value transformation $\lceil\cdot\rceil$ is defined as in (33), where R is the distinguished type of responses. For p atomic, $\lceil p \rceil = p$. Notice that the $\lceil\cdot\rceil$ translations for the (co)implications are related vertically by left/right symmetry $\cdot^{\bowtie}$ and in the horizontal dimension by arrow reversal $\cdot^\infty$. Under cbv, for every type A of the source language, one has values $\lceil A \rceil$, continuations (functions from values into R) and computations (functions from continuations into R).

$$\begin{array}{lclclcl} \lceil A \backslash B \rceil & = & R^{\lceil A \rceil \times R^{\lceil B \rceil}} & \qquad & \lceil B \oslash A \rceil & = & \lceil B \rceil \times R^{\lceil A \rceil}; \\ \lceil B / A \rceil & = & R^{R^{\lceil B \rceil} \times \lceil A \rceil} & & \lceil A \obslash B \rceil & = & R^{\lceil A \rceil} \times \lceil B \rceil. \end{array} \qquad (33)$$

The call-by-name interpretation $\lfloor\cdot\rfloor$ is obtained by duality: $\lfloor A\rfloor \triangleq \lceil A^{\infty}\rceil$. Under cbn, for every type A of the source language, we have continuations $\lfloor A\rfloor$ and computations (functions from continuations into R).

Let us turn then to the interpretation of proofs, i.e. arrows $f : A \to B$. Assuming countably infinite sets of variables x_i and covariables α_i, we inductively associate each arrow $f : A \to B$ with two closed terms, $f^{\triangleright}$ and $f^{\triangleleft}$, depending on whether we focus on A or B as the active formula. The induction is set up in such a way that we obtain the mappings of (34).

$$\lceil f^{\triangleright}\rceil : \lceil A\rceil \mapsto R^{R^{\lceil B\rceil}} \qquad \text{and} \qquad \lceil f^{\triangleleft}\rceil : R^{\lceil B\rceil} \mapsto R^{\lceil A\rceil} \tag{34}$$

i.e. call-by-value $\lceil f^{\triangleright}\rceil$ maps from A values to B computations; $\lceil f^{\triangleleft}\rceil$ from B continuations to A continuations; call-by-name is dual, with $(f^{\triangleright})^{\infty} = (f^{\infty})^{\triangleleft}$, $(f^{\triangleleft})^{\infty} = (f^{\infty})^{\triangleright}$: $\lfloor f^{\triangleright}\rfloor = \lceil f^{\triangleleft}\rceil$ maps B computations to A computations. The basis of the induction is given by the interpretation of the identity arrow $1_A : A \to A$ in (35).[7] For the recursive cases, we define either the $f^{\triangleright}$ or the $f^{\triangleleft}$ version. The missing case is obtained by swapping the lhs and rhs arguments.

$$\lceil (1_A)^{\triangleright}\rceil\ x = \lambda k.(k\ x) \qquad \lceil (1_A)^{\triangleleft}\rceil\ \alpha = \lambda x.(\alpha\ x) \tag{35}$$

Monotonicity. Given $f : A \to B$ and $g : C \to D$ we have the monotonicity rules of (36). We use curly brackets to distinguish the meta-application of the function definition from the actual target language lambda term computed.

$$\begin{array}{rcl}
\lceil (g\backslash f)^{\triangleright}\rceil\ y &=& \lambda k.(k\ \lambda\langle x,\beta\rangle.(\{\lceil g^{\triangleright}\rceil\ x\}\ \lambda m.(y\ \langle m,\{\lceil f^{\triangleleft}\rceil\ \beta\}\rangle)))\\
\lceil (f/g)^{\triangleright}\rceil\ x &=& \lambda k.(k\ \lambda\langle \alpha,y\rangle.(\{\lceil g^{\triangleright}\rceil\ y\}\ \lambda m.(x\ \langle\{\lceil f^{\triangleleft}\rceil\ \alpha\},m\rangle)))\\
\lceil (f \oslash g)^{\triangleleft}\rceil\ \alpha &=& \lambda\langle x,\delta\rangle.(\{\lceil f^{\triangleright}\rceil\ x\}\ \lambda y.(\alpha\ \langle y,\{\lceil g^{\triangleleft}\rceil\ \delta\}\rangle))\\
\lceil (g \obslash f)^{\triangleleft}\rceil\ \beta &=& \lambda\langle \delta,x\rangle.(\{\lceil f^{\triangleright}\rceil\ x\}\ \lambda y.(\beta\ \langle\{\lceil g^{\triangleleft}\rceil\ \delta\},y\rangle))
\end{array} \tag{36}$$

Residuation. For the typing of the arrows f, see (15).

$$\begin{array}{llll}
\lceil (\triangleleft f)^{\triangleright}\rceil\ y & = \lambda k.(k\ \lambda\langle x,\gamma\rangle.(\{\lceil f^{\triangleright}\rceil\ \langle x,y\rangle\}\ \gamma)) & \lceil (\blacktriangleright f)^{\triangleleft}\rceil\ \alpha & = \lambda\langle z,\beta\rangle.(\{f^{\triangleleft}\ \langle\alpha,\beta\rangle\}\ z)\\
\lceil (\triangleright f)^{\triangleright}\rceil\ x & = \lambda k.(k\ \lambda\langle \gamma,y\rangle.(\{\lceil f^{\triangleright}\rceil\ \langle x,y\rangle\}\ \gamma)) & \lceil (\blacktriangleleft f)^{\triangleleft}\rceil\ \beta & = \lambda\langle \alpha,z\rangle.(\{f^{\triangleleft}\ \langle\alpha,\beta\rangle\}\ z)\\
\lceil (\triangleleft' f)^{\triangleright}\rceil\ \langle x,y\rangle & = \lambda\gamma.(\{\lceil f^{\triangleright}\rceil\ y\}\ \lambda n.(n\ \langle x,\gamma\rangle)) & \lceil (\blacktriangleright' f)^{\triangleleft}\rceil\ \langle\alpha,\beta\rangle & = \lambda z.(\{\lceil f^{\triangleleft}\rceil\ \alpha\}\ \langle z,\beta\rangle)\\
\lceil (\triangleright' f)^{\triangleright}\rceil\ \langle x,y\rangle & = \lambda\gamma.(\{\lceil f^{\triangleright}\rceil\ x\}\ \lambda m.(m\ \langle \gamma,y\rangle)) & \lceil (\blacktriangleleft' f)^{\triangleleft}\rceil\ \langle\alpha,\beta\rangle & = \lambda z.(\{\lceil f^{\triangleleft}\rceil\ \beta\}\ \langle \alpha,z\rangle)
\end{array} \tag{37}$$

Grishin interaction rules. The Grishin rules simply recombine the factors of the input structure. We work this out for $G1$ and its dual $G1'$, leaving the other cases for the reader.

$$\frac{f : A \obslash (B \otimes C) \to D}{\circledslash f : (A \obslash B) \otimes C \to D} \qquad \frac{f^{\infty} : D \to (C \oplus B)/A}{(\circledslash f)^{\infty} : D \to C \oplus (B/A)}$$

[7] We follow usual functional programming practice writing the function definitions equationally, e.g. $\lceil (1_A)^{\triangleright}\rceil$ is the function $\lambda x \lambda k.(k\ x)$.

$$\begin{array}{rcl} \lceil(\obslash f)^{\triangleright}\rceil\ \langle\langle w,v\rangle,z\rangle & = & \lambda\delta.((\lceil f^{\triangleright}\rceil\ \langle w,\langle v,z\rangle\rangle)\ \delta) \\ \lfloor(\obslash f)^{\triangleright}\rfloor = \lceil((\obslash f)^{\infty})^{\triangleleft}\rceil\ \langle\gamma,\delta\rangle & = & \lambda z.(\delta\ \lambda\langle\beta,x\rangle.((\lceil(f^{\infty})^{\triangleleft}\rceil\ \lambda m.(m\ \langle\langle\gamma,\beta\rangle,x\rangle)),z)) \end{array} \tag{38}$$

Example. We contrast the cbv (left) and cbn (right) interpretations of a simple sentence 'somebody left' involving a generalized quantifier expression of type $(s \oslash s) \obslash np$. Literals are indexed to facilitate identification of the axiom matchings.

$$\dfrac{\dfrac{\dfrac{\dfrac{np_2 \vdash np_3 \quad \dfrac{\dfrac{s_4 \vdash s_0 \quad s_1 \vdash s_5}{(s_4 \oslash s_5) \vdash (s_0 \oslash s_1)}\oslash}{s_4 \vdash ((s_0 \oslash s_1) \oplus s_5)}\blacktriangleright'}{(np_3 \backslash s_4) \vdash (np_2 \backslash ((s_0 \oslash s_1) \oplus s_5))}\backslash}{(np_2 \otimes (np_3 \backslash s_4)) \vdash ((s_0 \oslash s_1) \oplus s_5)}\triangleleft'}{((s_0 \oslash s_1) \obslash (np_2 \otimes (np_3 \backslash s_4))) \vdash s_5}\blacktriangleleft}{(((s_0 \oslash s_1) \obslash np_2) \otimes (np_3 \backslash s_4)) \vdash s_5}\obslash \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{s_5 \vdash s_1 \quad s_0 \vdash s_4}{(s_1 \backslash s_0) \vdash (s_5 \backslash s_4)}\backslash}{(s_5 \otimes (s_1 \backslash s_0)) \vdash s_4}\triangleleft' \quad np_3 \vdash np_2}{((s_5 \otimes (s_1 \backslash s_0)) \oslash np_2) \vdash (s_4 \oslash np_3)}\oslash}{(s_5 \otimes (s_1 \backslash s_0)) \vdash ((s_4 \oslash np_3) \oplus np_2)}\blacktriangleright'}{s_5 \vdash (((s_4 \oslash np_3) \oplus np_2)/(s_1 \backslash s_0))}\triangleright}{s_5 \vdash ((s_4 \oslash np_3) \oplus (np_2/(s_1 \backslash s_0)))}\obslash^{\infty}$$

$$\text{(cbv)} \quad \lambda c.((\lceil\!\lceil \mathbf{left} \rceil\!\rceil\ \langle \pi^2 \lceil\!\lceil \mathbf{somebody} \rceil\!\rceil, \lambda z.(\pi^1 \lceil\!\lceil \mathbf{somebody} \rceil\!\rceil\ \langle z,c\rangle)\rangle)$$

$$\text{(cbn)} \quad \lambda c.((\lfloor\!\lfloor \mathbf{somebody} \rfloor\!\rfloor\ \lambda\langle q,y\rangle.(y\ \langle c, \lambda c'.(\lfloor\!\lfloor \mathbf{left} \rfloor\!\rfloor\ \langle c',q\rangle)\rangle)))$$

3 Linguistic Applications

In-depth discussion of the linguistic applications of **LG** is beyond the scope of this paper. We suffice with some very simple illustrations. Recall the two problems with Lambek calculus discussed in §1: non-local semantic construal and displacement. These problems find a natural solution in the symmetric Lambek-Grishin systems. As shown in (39), in both cases the solution starts from a lexical type assignment from which the usual Lambek type is derivable.

$$\begin{array}{lr} \text{someone} & (s \oslash s) \obslash np \vdash s/(np\backslash s) \\ \text{which} & (n\backslash n)/((s \oslash s) \oplus (s/np)) \vdash (n\backslash n)/(s/np) \end{array} \tag{39}$$

Non-local scope construal. An example would be a sentence of the type 'Alice suspects someone is cheating'. The sentence has two natural interpretations: there could be a particular player whom Alice suspects of cheating (for example, because she sees a card sticking out of his sleeve), or she could have the feeling that cheating is going on, without having a particular player in mind (for example, because she finds two aces of spades in her hand). A Lambek type assignment $s/(np\backslash s)$ for 'someone' is restricted to local construal in the embedded clause, i.e. the second interpretation. The assignment $(s \oslash s) \obslash np$ also allows construal at the main clause level as required for the first interpretation. In (40) one finds a summary derivation[8] for the reading where 'someone' has wide scope.

[8] Full proof for the endsequent under $\cdot^{\infty}$ is $\blacktriangleright'\ \blacktriangleright'\ \obslash^{\star}\ \blacktriangleright\ \obslash\ \blacktriangleright\ \obslash\ \triangleleft\ \triangleright'(\blacktriangleright'\ \blacktriangleleft'\ (\blacktriangleright'(1_s \oslash 1_{np}) \obslash (1_s \oslash 1_{np}))/1_s)$.

By means of the Grishin interactions, the $(s \oslash s)$ subformula moves to the top level leaving behind a np resource *in situ*; $(s \oslash s)$ then shifts to the succedent by means of the dual residuation principle, and establishes scope via the dual application law. Under the CPS interpretation discussed above, the derivation is associated with the term in (41). The reader is invited to consult [25] for a CPS interpretation of the lexical constants which associates this term with the reading $(\exists\ \lambda x.((\mathsf{suspects}\ (\mathsf{cheating}\ x))\ \mathsf{alice}))$ as required.

$$\dfrac{\dfrac{\dfrac{np \otimes (((np\backslash s)/s) \otimes (np \otimes (np\backslash s))) \vdash s \quad s \vdash (s \oslash s) \oplus s}{np \otimes (((np\backslash s)/s) \otimes (np \otimes (np\backslash s))) \vdash ((s \oslash s) \oplus s)}\,\text{trans}}{((s \oslash s) \obslash (np \otimes (((np\backslash s)/s) \otimes (np \otimes (np\backslash s))))) \vdash s}\,\text{drp}}{np \otimes (((np\backslash s)/s) \otimes (\underbrace{((s \oslash s) \obslash np)}_{\mathsf{someone}} \otimes (np\backslash s))) \vdash s}\,G2 \tag{40}$$

$$\lambda c.(\llbracket\mathsf{someone}\rrbracket\ \lambda\langle q, y\rangle.(y\ \langle c, \lambda c'.(\llbracket\mathsf{suspect}\rrbracket\ \langle \lambda c''.(\llbracket\mathsf{cheating}\rrbracket\ \langle c'', q\rangle), \langle c', \llbracket\mathsf{alice}\rrbracket\rangle\rangle)\rangle)) \tag{41}$$

Displacement. The second example deals with *wh* dependencies as in '(movie which) John (np) saw $((np\backslash s)/np = tv)$ on TV $((np\backslash s)\backslash(np\backslash s) = adv)$'. The shorthand derivation in (42) combines the Grishin principles of (12) and their converses. The (s/np) subformula is added to the antecedent via the dual residuation principle, and *lowered* to the target tv via applications of (Gn^{-1}). The tv context is then shifted to the succedent by means of the (dual) residuation principles, and the relative clause body with its np hypothesis in place is reconfigured by means of (Gn) and residuation shifting.

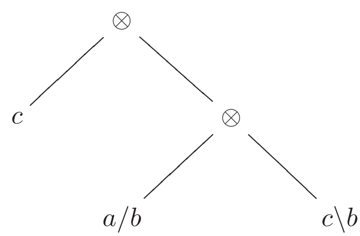
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{np \otimes ((tv \otimes np) \otimes adv) \vdash s \quad s \vdash (s \oslash s) \oplus s}{np \otimes ((tv \otimes np) \otimes adv) \vdash (s \oslash s) \oplus s}\,\text{trans}}{tv \vdash ((np\backslash(s \oslash s))/adv) \oplus (s/np)}\,Gn, \text{rp}}{np \otimes ((tv \oslash (s/np)) \otimes adv) \vdash s \oslash s}\,\text{rp,drp}}{(np \otimes (tv \otimes adv)) \oslash (s/np) \vdash s \oslash s}\,Gn^{-1}}{np \otimes (tv \otimes adv) \vdash (s \oslash s) \oplus (s/np)}\,\text{drp} \tag{42}$$

The derivation can be summarized in the derived rule of inference (†), which achieves the same effect as the extraction rule under modal control (‡). An attractive property of **LG** is that the expressivity resides entirely in the Grishin interaction principles: the composition operation $\otimes$ in itself (or the dual $\oplus$) allows no structural rules at all, which means that the **LG** notion of wellformedness is fully sensitive to linear order and constituent structure of the grammatical material.

$$\dfrac{\Gamma[\Delta \circ B] \Rightarrow C}{\Gamma[\Delta] \Rightarrow (C \oslash C) \oplus (C/B)}\,\dagger \qquad \dfrac{\Gamma[\Delta \circ B] \Rightarrow C}{\Gamma[\Delta] \Rightarrow C/\Diamond\Box B}\,\ddagger \tag{43}$$

Similarity. Rotations of the type $((A\backslash C)/B \sim A\backslash(C/B)$, $(C/B)/A \sim (C/A)/B$ make it possible to promote any embedded argument to a left or right peripheral position where it is visible for slash introduction. Since the original and the rotated types are in the $\sim$ relation, one can lexically assign their meet type according to the algorithm given in the previous section. Below we look at extraction and head adjunction from this perspective. We show how the stategy of assigning a meet type can be used to overcome the limitations of the Lambek calculus (both **NL** and **L**), and how overgeneration can be avoided by appropriate modal decoration.

Phenomena of head adjunction likewise give rise to dependencies for which the rotation invariant proves useful. In (44) below one finds a schematic representation of a crossed dependency in Dutch, as in the phrase '(dat Jan) boeken (c) wil (a/b) lezen ($c\backslash b$)' with the order object–modal auxiliary–transitive infinitive. One would like to derive type a (tensed verb phrase) for this structure. As with extraction, there is a double challenge: one wants to allow the transitive infinitive to communicate with its direct object across the intervening modal auxiliary; at the same time, one has to rule out the ungrammatical order $(a/b) \otimes (c \otimes (c\backslash b))$ which with the indicated types would make a derivable.



(44)

To bring our strategy into play, note that

$$c\backslash b \overset{\text{(lifting)}}{\sim} c\backslash((a/b)\backslash a) \overset{\text{(rotation)}}{\sim} (a/b)\backslash(c\backslash a)$$

For the original $c\backslash b$ and the rotated $(a/b)\backslash(c\backslash a)$ we have join C and meet D:

$$C = ((a/b)\backslash a) \oplus (c\backslash(a \obslash a))$$

$$D = ((c\backslash b)/C) \otimes (C \oslash (((a/b)\backslash(c\backslash a)) \obslash C))$$

To make the ungrammatical order modal auxiliary–object–transitive infinitive underivable, we can impose subtyping constraints using modal decoration. It is enough to change the type of the modal auxiliary to $a/\Diamond\Box b$, and modify D accordingly, marking the rotated argument:

$$D' = ((c\backslash b)/C) \otimes (C \oslash (((a/\Diamond\Box b)\backslash(c\backslash a)) \obslash C))$$

Recall that $\Diamond\Box A \vdash A$. The join type C in other words can remain as it was since

$$(a/\Diamond\Box b)\backslash(c\backslash a) \vdash ((a/b)\backslash a) \oplus (c\backslash(a \obslash a))$$

4 Conclusions

Natural languages exhibit mismatches between the articulation of compound expressions at the syntactic and at the semantic level; such mismatches seem to obstruct a direct compositional interpretation. In the face of this problem, two kinds of reaction have been prominent. The first kind is exemplified by Curry's [27] position in his contribution to the 1960 New York conference where also [2] was presented: Curry criticizes Lambek for incorporating syntactic considerations in his category concept, and retreats to a semantically biased view of categories. The complementary reaction is found in Chomskyan generative grammar, where precedence is given to syntax, and where the relevance of modeltheoretic semantics is questioned.

The work reviewed in this paper maintains the strong view that the type-forming operations are constants both in the syntactic and in the semantic dimension. Semantic uniformity is reconciled with structural diversity by means of structure-preserving interactions between the composition operation $\otimes$ and its residuals and a dual family $\oplus$; the symmetries among these families restore the strong Curry-Howard view on derivational semantics.

References

1. Grishin, V.: On a generalization of the Ajdukiewicz-Lambek system. In: Studies in Nonclassical Logics and Formal Systems. Nauka, Moscow (1983) 315–334 [English translation in Abrusci and Casadio (eds.) Proceedings 5th Roma Workshop, Bulzoni Editore, Roma (2002)]
2. Lambek, J.: On the calculus of syntactic types. In: Jakobson, R. (ed.) Structure of Language and Its Mathematical Aspects, pp. 166–178 (1961)
3. Došen, K.: A brief survey of frames for the Lambek calculus. Zeitschrift für mathematischen Logik und Grundlagen der Mathematik 38, 179–187 (1992)
4. Moortgat, M.: Multimodal linguistic inference. Language and Information 5(3–4), 349–385 (1996)
5. Bernardi, R.: Reasoning with polarities in categorial type logic. PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University (2002)
6. Vermaat, W.: The logic of variation. A cross-linguistic account of wh-question formation. PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University (2006)
7. Kurtonina, N., Moortgat, M.: Structural control. In: Blackburn, P., de Rijke, M. (eds.) Specifying Syntactic Structures, pp. 75–113. CSLI Publications, Stanford, CA (1997)
8. Moortgat, M.: Categorial type logics. In: van Benthem, J., ter Meulen, A. (eds.) Handbook of Logic and Language. Elsevier/MIT Press, 93–177 (1997)
9. de Groote, P.: The non-associative Lambek calculus with product in polynomial time. In: Murray, N.V. (ed.) TABLEAUX 1999. LNCS (LNAI), vol. 1617, pp. 128–139. Springer, Heidelberg (1999)
10. Pentus, M.: Lambek calculus is NP-complete. Theoretical Computer Science 357, 186–201 (2006)
11. Moot, R.: Proof Nets for Linguistic Analysis. PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University (2002)

12. Dunn, J.M.: Gaggle theory: An abstraction of galois connections and residuation with applications to negation, implication, and various logical operations. In: van Eijck, J. (ed.) JELIA 1990. LNCS, vol. 478, pp. 31–51. Springer, Heidelberg (1991)
13. Lambek, J.: From categorial to bilinear logic. In: Schröder-Heister, K.D.P. (ed.) Substructural Logics, Oxford University Press, Oxford (1993)
14. Goré, R.: Substructural logics on display. Logic Journal of IGPL 6(3), 451–504 (1997)
15. Moortgat, M., Oehrle, R.: Proof nets for the grammatical base logic. In: Abrusci, M., Casadio, C. (eds.) Dynamic perspectives in logic and linguistics, Roma, Bulzoni pp. 131–143 (1999)
16. De Groote, P., Lamarche, F.: Classical non-associative Lambek calculus. Studia Logica 71, 335–388 (2002)
17. Capelletti, M.: Parsing with structure-preserving categorial grammars. PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University (2007)
18. Lambek, J.: The mathematics of sentence structure. American Mathematical Monthly 65, 154–170 (1958)
19. Foret, A.: On the computation of joins for non-associative Lambek categorial grammars. In: Levy, J., Kohlhase, M., Niehren, J., Villaret, M. (eds.) Proceedings of the 17th International Workshop on Unification, UNIF'03 Valencia, pp. 25–38 (2003)
20. Pentus, M.: The conjoinability relation in Lambek calculus and linear logic. ILLC Prepublication Series ML–93–03, Institute for Logic, Language and Computation, University of Amsterdam (1993)
21. Moortgat, M., Pentus, M.: Type similarity for the Lambek-Grishin calculus. AI preprint series, Utrecht University (2007) (Submitted to the 12th Formal Grammar Conference, Dublin)
22. Kurtonina, N.: Frames and Labels. A Modal Analysis of Categorial Inference. PhD thesis, OTS Utrecht, ILLC Amsterdam (1995)
23. Allwein, G., Dunn, J.M.: Kripke models for Linear Logic. Journal of Symbolic Logic 58(2), 514–545 (1993)
24. Kurtonina, N., Moortgat, M.: Relational semantics for the Lambek-Grishin calculus. Technical report, AI Preprint Series, Utrecht (2007) (Submitted to MOL'10, UCLA, Los Angeles)
25. Bernardi, R., Moortgat, M.: Continuation semantics for symmetric categorial grammar. In: Leivant, D., de Queiros, R. (eds.) Proceedings 14th Workshop on Logic, Language, Information and Computation (WoLLIC'07) 2007, LNCS, vol. 4576, Springer, Heidelberg (this volume)
26. Curien, P., Herbelin, H.: Duality of computation. In: International Conference on Functional Programming (ICFP'00), 2000, pp. 233–243 [2005: corrected version]
27. Curry, H.B.: Some logical aspects of grammatical structure. In: Jacobson, R. (ed.) Structure of Language and its Mathematical Aspects. Volume XII of Proceedings of the Symposia in Applied Mathematics. American Mathematical Society, pp. 56–68 (1961)

A Admissibility of Cut

A cut-elimination algorithm for the $\mathbf{Frm}(/, \otimes, \backslash)$ fragment (i.e. **NL**) is presented in Moortgat and Oehrle [15]. Induction is on the degree of a cut inference, measured as the number of type-forming operations in the factors involved ($|A|+|B|+|C|$, where

B is the cut formula, A the left premise lhs, C the right premise rhs). Targets for elimination are 'uppermost' cut inferences: instances of cut which are themselves derived without the aid of the cut rule. One shows that such an instance of cut can be replaced by one or more cuts of lower degree; the basis of the induction being the case where one of the cut premises is an instance of the axiom schema. The elimination process is iterated until the derivation is cut-free.

For the transformations one distinguishes principal cuts from permutation cases. Principal cuts, in the combinator presentation of §2.1, are cases where the cut formula in the two premises is introduced by the monotonicity rules. Such cuts are replaced by cuts on the four subformulas involved, with a reduction of the complexity degree. For the non-principal cases, one shows that the application of cut can be pushed upwards, again reducing complexity. The extension of the procedure of [15] to $\mathbf{Frm}(\oslash, \oplus, \obslash)$ is immediate via arrow reversal. In Fig 3 and Fig 4 we give the $\oplus$ and $\oslash$ cases with the corresponding equations on the proof terms; the $\obslash$ case is symmetric. This covers the minimal symmetric system $\mathbf{LG}_{\emptyset}$. For full $\mathbf{LG}$ (the extension of $\mathbf{LG}_{\emptyset}$ with the $\mathcal{G}^{\uparrow}$ interaction principles of (12)), what remains to be shown is that applications of cut never have to be immediately preceded by applications of the Grishin interaction rules. In Fig 5 is an instance of cut immediately preceded by $\olessthan$. We have unfolded the left

$$\dfrac{h : D \to A \oplus B \qquad \dfrac{\dfrac{f : A \to A' \quad g : B \to B'}{A \oplus B \to A' \oplus B'}}{\dfrac{\vdots}{A \oplus B \to C}}}{D \to C} \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{h : D \to A \oplus B}{D \oslash B \to A}\blacktriangleright \quad f : A \to A'}{D \oslash B \to A'}}{D \to A' \oplus B}\blacktriangleright'}{A' \obslash D \to B}\blacktriangleleft \qquad g : B \to B'}{A' \obslash D \to B'}}{D \to A' \oplus B'}\blacktriangleleft'}{\dfrac{\vdots}{D \to C}}$$

Fig. 3. $k[f \oplus g] \circ h = k[\blacktriangleleft'(g \circ \blacktriangleleft \blacktriangleright' (f \circ \blacktriangleright h))]$

$$\dfrac{\dfrac{\dfrac{f : C' \to C \quad g : B \to B'}{C' \oslash B' \to C \oslash B}}{\dfrac{\vdots}{D \to C \oslash B}} \qquad h : C \oslash B \to A}{D \to A} \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{f : C' \to C \quad \dfrac{h : C \oslash B \to A}{C \to A \oplus B}\blacktriangleright'}{C' \to A \oplus B}}{A \obslash C' \to B}\blacktriangleleft \qquad g : B \to B'}{A \obslash C' \to B'}}{C' \to A \oplus B'}\blacktriangleleft'}{C' \oslash B' \to A}\blacktriangleright}{\dfrac{\vdots}{D \to A}}$$

Fig. 4. $h \circ k[f \oslash g] = k[\blacktriangleright \blacktriangleleft' (g \circ \blacktriangleleft(\blacktriangleright' h \circ f))]$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{g : A \to A' \quad f : B' \to B}{A' \obslash B' \to A \obslash B}}{\vdots}}{k_1[g \obslash f] : E' \to A \obslash B} \qquad h : C' \to C}{E' \otimes C' \to (A \obslash B) \otimes C}}{\vdots}}{k_1'[k_1[g \obslash f] \otimes h] : E \to (A \obslash B) \otimes C} \qquad \dfrac{k_2 : A \obslash (B \otimes C) \to D}{\olessthan\, k_2 : (A \obslash B) \otimes C \to D}}{\olessthan\, k_2 \circ k_1'[k_1[g \obslash f] \otimes h] : E \to D}$$

Fig. 5. An instance of $G1$ feeding cut

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{h : C' \to C \qquad \dfrac{\dfrac{\dfrac{f : B' \to B \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{k_2 : A \obslash (B \otimes C) \to D}{B \otimes C \to A \oplus D}\,\blacktriangleleft'}{(B \otimes C) \oslash D \to A}\,\blacktriangleright \qquad g : A \to A'}{(B \otimes C) \oslash D \to A'}}{B \otimes C \to A' \oplus D}\,\blacktriangleright'}{B \to (A' \oplus D)/C}\,\triangleright}{B' \to (A' \oplus D)/C}}{B' \otimes C \to A' \oplus D}\,\triangleright'}{C \to B' \backslash (A' \oplus D)}\,\triangleleft}{C' \to B' \backslash (A' \oplus D)}}{B' \otimes C' \to A' \oplus D}\,\triangleleft'}{A' \obslash (B' \otimes C') \to D}\,\blacktriangleleft}{(A' \obslash B') \otimes C' \to D}\,\olessthan}{A' \obslash B' \to D/C'}\,\triangleright}{\vdots}}{\dfrac{E' \to D/C'}{E' \otimes C' \to D}\,\triangleright'}}{\vdots}}{E \to D}$$

$$\olessthan\, k_2 \circ k_1'[k_1[g \obslash f] \otimes h] = k_1'[\triangleright' k_1[\triangleright \olessthan \blacktriangleleft \triangleleft' (\triangleleft \triangleright' (\triangleright \blacktriangleright' (g \circ \blacktriangleright \blacktriangleleft' k_2) \circ f) \circ h)]]$$

Fig. 6. Permutability of Grishin and cut

cut premise so as to unveal the applications of the $\obslash$ and $\otimes$ monotonicity rules within their contexts. This derivation can be rewritten as in Fig 6 where the cut on $(A \obslash B) \otimes C$ is replaced by cuts on the factors A, B and C, followed by the Grishin inference $\olessthan$.