Symmetric Categorial Grammar

Course Notes ESSLLI'07, Dublin

Raffaella Bernardi and Michael Moortgat

Contents

Preface

The following chapters constitute the reading materials for our advanced course on *Symmetric Categorial Grammar* at ESSLLI'07, Dublin. We give some bibliographic information below.

Chapter 1 We present a corrected version, prepared by Anna Chernilovskaya, of the English translation of Grishin's seminal paper from 1983. The translation originally appeared in V.M. Abrusci and C. Casadio (eds.) New Perspectives in Logic and Formal Linguistics, Bulzoni Editore, Rome, 2002. The corrected text is included here with permission of the editors and the author.

Chapter 2 Draft of an invited paper for WoLLIC'07, Rio de Janeiro, July 2–5, 2007. To appear under this title in D. Leivant and R. de Queiroz (eds.) Proceedings 14th Workshop on Logic, Language, Information and Computation. Springer LNCS 4576, pages 264–284.

Chapter 3 Draft of a contributed paper for WoLLIC'07, Rio de Janeiro, July 2–5, 2007. To appear under this title in D. Leivant and R. de Queiroz (eds.) Proceedings 14th Workshop on Logic, Language, Information and Computation. Springer LNCS 4576, pages 53–71.

Chapter 4 Draft of a paper accepted for MOL'07, UCLA, July 28–30, 2007. Final version to appear in M. Kracht, G. Penn and E. Stabler (eds.) Proceedings of the 10th Mathematics of Language Conference. UCLA Working Papers in Linguistics.

Chapter 5 A chapter from Anna Chernilovskaya's "Notes on the Lambek-Grishin calculus", prepared in the context of her diploma work (Moscow State University, June 2007).

Chapter 6 Draft of a paper accepted for FG'07, Dublin, August 4–5, 2007. Final version to appear in L. Kallmeyer et.al. (eds.) Proceedings of the 12th Conference on Formal Grammar.

June 2007

Raffaella Bernardi
Bolzano
`bernardi@inf.unibz.it`

Michael Moortgat
Utrecht
`moortgat@let.uu.nl`

1

On a Generalization of the Ajdukiewicz-Lambek System

Viacheslav Nikolaevich Grishin

Bibliographic note Original version "Об одном обобщении системы Айдукевича–Ламбека" in A.I. Mikhailov (ed.) *Studies in non-classical logics and formal systems*, Nauka, Moscow 1983, pages 315–334 (*Исследования по неклассическим логикам и формальным системам*, Наука, M., 1983). An English translation by D. Čubrić , edited by the author, appeared in V.M. Abrusci and C. Casadio (eds.) *New Perspectives in Logic and Formal Linguistics*. Proceedings 5th Roma workshop, Bulzoni Editore, Rome, 2002. The corrected version below was prepared by Anna Chernilovskaya.

In this article we consider ordered algebraic systems, generalizing partially ordered groups [3], and Heyting-Brouwer algebras [7].

In groups there is a division $a\backslash y = a^{-1}y$, satisfying the equivalence $ax = y \Leftrightarrow x = a\backslash y$. In Heyting algebras there is an implication $a \supset y$, satisfying “almost” the same equivalence

$$ax \leq y \Leftrightarrow x \leq a \supset y \tag{1.1}$$

but only having $\leq$ instead of $=$. Splitting the group operation in two operations $a \cdot x$ and $a \times x$ and for each of them writing (1.1) but with reversed inequality in the second case, we get (1.1) and also

$$a \times x \geq y \Leftrightarrow x \geq a \because y \ . \tag{1.2}$$

This is the idea of the generalization.

The question about a generalization of group-like and logical systems was brought up by Birkhoff in the case of lattice-ordered groups (ℓ -groups) and Boolean algebras [1]. Swamy [8] described a class of systems generalizing commutative ℓ-groups and Brouwer algebras.

Here we suggest a class of systems that is considerably broader than Swamy's class and includes unordered groups as well. Adding certain inequalities to the axioms of the algebraic systems considered, one gets not only ordered groups and Heyting-Brouwer algebras, but also some other systems which appear in the literature, for example Chang's MV-algebras [2] and L°-algebras [5]. Although the situation is more general, many notions and properties that are known for groups and Heyting algebras have their analogies here, from which one gets respective notions and properties in groups and Heyting algebras as a particular case.

The word "generalization" from the title refers to the axiomatization and not to the class of systems satisfying the axioms. This class is smaller than the Ajdukiewicz-Lambek class, and the axiomatization is, respectively, larger than in Lambek [6].

1.1 Axiomatization

1.1 We shall consider ordered algebraic systems

$$\mathbf{A} = (A; \quad \leq, \quad \cdot, \quad \backslash, \quad /, \quad \mathbf{1}, \quad \times, \quad \therefore, \quad \because, \quad \mathbf{I}),$$

satisfying the preorder axioms:

PO1. $x \leq x$; PO2. $x \leq y \;\&\; y \leq z \Longrightarrow x \leq z$

and the following axioms:

A1. Axioms of adjointness:

1. $a \cdot x \leq y \Leftrightarrow x \leq a\backslash y$ 2. $x \cdot a \leq y \Leftrightarrow x \leq y/a$
3. $y \leq a \times x \Leftrightarrow a \therefore y \leq x$ 4. $y \leq x \times a \Leftrightarrow y \because a \leq x$

A2. Axioms of same sort associativity:

1. $b \cdot (x \cdot a) \leq (b \cdot x) \cdot a$ 2. $(a \cdot x) \cdot b \leq a \cdot (x \cdot b)$
3. $b \times (x \times a) \leq (b \times x) \times a$ 4. $(a \times x) \times b \leq a \times (x \times b)$

A3. Axioms of mixed associativity:

1. $b \cdot (x \times a) \leq (b \cdot x) \times a$ 2. $(a \times x) \cdot b \leq a \times (x \cdot b)$

A4. Axioms of neutrality of the constants $\mathbf{1}$ and $\mathbf{I}$:

1. $x \leq \mathbf{1} \cdot x \leq x$ 2. $x \leq x \cdot \mathbf{1} \leq x$
3. $x \leq \mathbf{I} \times x \leq x$ 4. $x \leq x \times \mathbf{I} \leq x$

Here and from now on x, y, z, a, b are variables ranging over $\mathbf{A}$. Equality of two elements x and y from $\mathbf{A}$ is understood as a conjunction of two inequalities

$$x = y \rightleftharpoons (x \leq y) \;\&\; (y \leq x)$$

It is well known that from the axioms of adjointness one can derive axioms of monotonicity (congruence of operations with the order):

AM.
$$\begin{array}{lcll} x \leq x_1 \;\&\; y \leq y_1 & \Longrightarrow & x \cdot y \leq x_1 \cdot y_1 \quad \& \quad x \times y \leq x_1 \times y_1 & \& \\ & & x_1\backslash y \leq x\backslash y_1 \quad \& \quad y/x_1 \leq y_1/x & \& \\ & & x_1 \therefore y \leq x \therefore y_1 \quad \& \quad y \because x_1 \leq y_1 \because x & \end{array}$$

1.2 Ajdukiewicz-Lambek systems are systems of the type $(A;\ \leq,\ \cdot,\ \backslash,\ /)$, satisfying the preorder axioms PO1 and PO2 and the axioms A1.1, A1.2, A2.1, A2.2. System A1-A4 consists of two Ajdukiewicz-Lambek systems with neutral elements: one is $(A;\ \leq,\ \cdot,\ \backslash,\ /,\ \mathbf{1})$, and another is $(A;\ \geq,\ \times,\ \therefore,\ \because,\ \mathbf{I})$ which are interconnected with the A3 axioms.

1.3 Partially ordered groups in a signature: $\leq$ – a preorder, $\circ$ – a group composition, $\backslash$ – a left division, $/$ – a right division, $\mathbf{e}$ – a neutral element — can be defined with axioms PO1, PO2 together with the following axioms:

G1. $b \circ (x \circ a) \leq (b \circ x) \circ a \qquad (b \circ x) \circ a \leq b \circ (x \circ a)$

G2. $a \circ x \leq y \Leftrightarrow x \leq a \backslash y \qquad x \circ a \leq y \Leftrightarrow x \leq y/a$
$y \leq a \circ x \Leftrightarrow a \backslash y \leq x \qquad y \leq x \circ a \Leftrightarrow y/a \leq x$

G3. $x \leq \mathbf{e} \circ x \leq x \qquad x \leq x \circ \mathbf{e} \leq x$

If we define $x = y \rightleftharpoons (x \leq y)\ \&\ (y \leq x)$, then from G2 follows: $a \circ x = y \Leftrightarrow x = a \backslash y$, and $x \circ a = y \Leftrightarrow x = y/a$. Axioms A1-A4 are obtained from the group axioms G1-G3 by substituting in certain places the group composition for operations $\cdot$ and $\times$, the left division for left quasi-divisions $\backslash$ and $\therefore$, the right division for right quasi-divisions $/$ and $\because$, the neutral element for $\mathbf{1}$ and $\mathbf{I}$.

An ordered group can be considered as an example of a system satisfying A1-A4, if we define $a \cdot b = a \circ b$, $a \times b = a \circ b$, $a \therefore b = a \backslash b$, $b \because a = b/a$, $\mathbf{1} = \mathbf{I} = \mathbf{e}$, and $\circ$, $\backslash$, $/$ are unchanged.

1.4 Recall that a Heyting-Brouwer algebra is a Heyting (pseudoboolean) algebra with an additional binary operation of subtraction $y - a$, satisfying the equivalence $y \leq x \vee a \Leftrightarrow y - a \leq x$. Heyting-Brouwer algebras also satisfy the axioms A1-A4. Here we define $a \cdot b = a \wedge b$, $a \times b = a \vee b$, $a \backslash b = b/a = a \supset b$, $a \therefore b = b \because a = b - a$; $\mathbf{1}$ and $\mathbf{I}$ are supremum and infimum respectively.

1.5 The system A1-A4 has two kinds of duality.

First. Axioms are translated into axioms substituting $\leq$ for $\geq$ and every operation symbol ω for its dual ω° according to the table

ω	$\cdot$	$\times$	$\backslash$	$/$	$\therefore$	$\because$	$\mathbf{1}$	$\mathbf{I}$
ω°	$\times$	$\cdot$	$\therefore$	$\because$	$\backslash$	$/$	$\mathbf{I}$	$\mathbf{1}$

Second, also called the *principle of symmetry*. Axioms are translated into axioms by substituting every term t for the symmetrical term $t^\sim$ which is defined inductively:

$$x^\sim \rightleftharpoons x \qquad (x \text{ is a variable})$$
$$(t_1\, \omega\, t_2)^\sim \rightleftharpoons (t_2^\sim\, \omega^\sim\, t_1^\sim),$$

where $\omega^\sim$ is the symmetrical operation symbol as in the table

ω	$\cdot$	$\times$	$\backslash$	$/$	$\ddots$	$\iddots$	$\mathbf{1}$	$\mathbf{I}$
$\omega^{\sim}$	$\cdot$	$\times$	$/$	$\backslash$	$\iddots$	$\ddots$	$\mathbf{1}$	$\mathbf{I}$

Let us denote as t° the term obtained from t by substitution ω for ω°. Then $t^{\circ\sim} \doteq t^{\sim\circ}$, where $\doteq$ is graphical equality. Therefore, for every statement $\mathcal{A}$ derived from A1-A4, we automatically get three more valid statements: $\mathcal{A}^{\circ}$, $\mathcal{A}^{\sim}$, $\mathcal{A}^{\circ\sim}$.

1.2 Derivable formulas and equivalent axiomatizations

2.1 Since many derivations are performed similarly, it is convenient to introduce some "unified" notations.

With every binary operation $\omega : \mathbf{A} \times \mathbf{A} \to \mathbf{A}$ one naturally associates two operations of the type $\mathbf{A} \to \mathbf{A}^{\mathbf{A}}$. One, denoted as $(\omega ?)$, converts $a \in \mathbf{A}$ into the operation of the left shifting: $x \mapsto a\,\omega\,x$; another one, denoted as $(?\omega)$, transforms $a \in \mathbf{A}$ into the operation of the right shifting: $x \mapsto x\,\omega\,a$. Since in A1-A4 there are 6 binary operations, we get 12 operations $\mathbf{A} \to \mathbf{A}^{\mathbf{A}}$. Let

$$\Omega = \{\, \cdot ?, \quad \backslash ?, \quad ?\backslash, \quad /?, \quad ?/, \quad ?\cdot, \quad \times ?, \quad \ddots ?, \quad ?\ddots, \quad \iddots ?, \quad ?\iddots, \quad ?\times \,\}$$

be the set of these operations' symbols. Let λ, μ, ν denote elements of Ω. With $a^{\mu}x$ we denote the term $(a\,\omega\,x)$, if $\mu = (\omega ?)$, and the term $(x\,\omega\,a)$, if $\mu = (?\omega)$, where ω is a binary operation symbol from A1-A4.

2.2 For $\varepsilon \in \{0, 1\}$ we denote

$$x \le_{\varepsilon} y \rightleftharpoons \begin{cases} x \le y, & \text{if } \varepsilon = 0 \\ y \le x, & \text{if } \varepsilon = 1 \end{cases}$$

Also let

$$|\mu| = \begin{cases} 1, & \text{if } \mu = ?\backslash \quad \text{or } \mu = /? \quad \text{or } \mu = ?\ddots \quad \text{or } \mu = \iddots ? \\ 0, & \text{otherwise} \end{cases}$$

The following formula will be called the *monotonicity axiom for* μ:

AM_{μ}. $\qquad \forall\, x, y, a\, (x \le y \Longrightarrow a^{\mu}x \le_{|\mu|} a^{\mu}y)$

The monotonicity axiom for μ can be rewritten in a form:

$$\forall\, x, y, a\, (x \le_{|\mu|} y \Longrightarrow a^{\mu}x \le a^{\mu}y)$$

The monotonicity axioms are derivable from A1, and they state isotonicity or antitonicity of system operations in appropriate argument places.

2.3 Let us define a function $\perp : \Omega \to \Omega$ with a table

μ	$\cdot ?$	$?\backslash$	$?/$	$\times ?$	$?\ddots$	$?\iddots$
$\mu\perp$	$\backslash ?$	$/?$	$?\cdot$	$\ddots ?$	$\iddots ?$	$?\times$

and with an equality $\mu\perp\perp = \mu$.

We will call the *axiom of left adjunction for* μ the following formula

LA_μ . $\qquad \forall\, a,x,y\,(a^\mu x \le y \Leftrightarrow x \le_{|\mu|} a^{\mu\perp} y).$

The axiom of right adjunction for μ is

RA_μ . $\qquad \forall\, a,x,y\,(y \le a^\mu x \Leftrightarrow a^{\mu\perp} y \le_{|\mu|} x).$

If μ is any symbol from the set $\{\, ?\cdot,\quad \cdot?,\quad ?\because,\quad \because?,\quad ?\therefore,\quad \therefore?\,\}$, LA_μ follows from $\mathsf{A1}$. For the other symbols μ the formula RA_μ is provable. Therefore, the latter symbols are called *right* and the former *left*. Let us show, for example, RA_μ for $\mu = ?\backslash$.

$$y \le a^\mu x \Leftrightarrow y \le x\backslash a \Leftrightarrow x \cdot y \le a \Leftrightarrow x \le a/y \Leftrightarrow a^{\mu\perp} y \le_{|\mu|} x.$$

Remark. Words "left" and "right" come from the category theory. If an ordered set $\mathbf{C} = (A;\ \le)$ is considered as a category, and operation of shifting $x \mapsto a^\mu x$ as a functor $F\colon \mathbf{C} \to \mathbf{C}^{|\mu|}$, where

$$\mathbf{C}^{|\mu|} = \begin{cases} \mathbf{C} & , \quad \text{if } |\mu| = 0 \\ \mathbf{C}^{op} \text{ (dual category)} & , \quad \text{if } |\mu| = 1, \end{cases}$$

then F is left adjoint of the functor $G : \mathbf{C}^{|\mu|} \to \mathbf{C} = \mathbf{C}^{|\mu|+|\mu\perp|}$, where G is $x \mapsto a^{\mu\perp} x$.

2.4 Let

$$\mu^* \rightleftharpoons \begin{cases} ?\omega & , \quad \text{if } \mu = \omega? \\ \omega? & , \quad \text{if } \mu = ?\omega. \end{cases}$$

Obviously $a^{\mu^*} b \doteq b^\mu a$. It is also easy to see that if we start with $\mu = \cdot?$, the ordered sequence μ , $\mu\perp$, $\mu\perp^*$, $\mu\perp^*\perp$, $\mu\perp^*\perp^*$, $\mu\perp^*\perp^*\perp$ is the sequence of the first six elements of Ω . If $\mu = \times?$, the sequence above gives the remaining six elements of Ω .
Let us list all needed properties of the functions $*$, $\perp$, $|\cdot|$, which are direct consequences of their definitions:

(1) $\mu^{**} = \mu$, $\mu\perp\perp = \mu$
(2) $\mu^*\perp^* = \mu\perp^*\perp$
(3) $|\mu\perp| = |\mu|$, $|\mu\perp^*| = |\mu| + |\mu^*| + 1$. (Here the addition is modulo 2)

2.5 We shall consider the following six inequalities:

(1) $\forall\, a,b,c\,(a^{\mu} b^{\lambda} c \le b^{\lambda} a^{\mu} c)$

(2) $\forall\, a,b,c\,(b^{\lambda} a^{\mu\perp} c \le_{|\mu|} a^{\mu\perp} b^{\lambda} c)$

(3) $\forall\, a,b,c\,(a^{\lambda\perp} b^{\mu} c \le_{|\lambda|} b^{\mu} a^{\lambda\perp} c)$

(4) $\forall\, a,b,c\,((a^{\lambda\perp} b)^{\mu^*\perp} c \le_{|\mu^*|} b^{\mu^*\perp} a^{\lambda} c)$

(5) $\forall\, a,b,c\,(a^{\lambda^*\perp} b^{\mu} c \le_{|\lambda^*|} (b^{\mu\perp} a)^{\lambda^*\perp} c)$

(6) $\forall\, a,b,c\,((c^{\mu^*\perp} b)^{\mu} a \le (c^{\lambda^*\perp} a)^{\lambda} b)$

Theorem 1. The following implications hold:
(1.1) $|\lambda| = 0$ & LA_μ & $\mathsf{AM}_\lambda \Longrightarrow (1) \Leftrightarrow (2)$,

(1.2) $|\mu| = 0$ & RA_λ & $\mathsf{AM}_\mu \Longrightarrow (1) \Leftrightarrow (3)$,

(1.3) $|\lambda| = 0$ & LA_{μ^*} & $\mathsf{RA}_\lambda \Longrightarrow (3) \Leftrightarrow (4)$,

(1.4) $|\mu| = 0$ & RA_{λ^*} & $\mathsf{LA}_\mu \Longrightarrow (2) \Leftrightarrow (5)$,

(1.5) $|\mu| = |\lambda| = 0$ & LA_{μ^*} & RA_{λ^*} & AM_μ & AM_λ & AM_{μ^*} & $\mathsf{AM}_{\lambda^*} \Longrightarrow (1) \Leftrightarrow (6)$

Proof. For every formula $\mathcal{A}$ and every $\varepsilon \in \{0,1\}$, $(\mathcal{A})_\varepsilon$ stands for the formula in which $\leq$ is replaced by $\leq_\varepsilon$. If a formula $\mathcal{A}$ is one of the six inequalities having a certain number, for example (1), then $(1)_\varepsilon$ means $(\mathcal{A})_\varepsilon$. If A is the left adjunction axiom for μ , $(A)_\varepsilon$ is

$(\mathsf{LA}_\mu)_\varepsilon$. $$a^\mu x \leq_\varepsilon y \Leftrightarrow x \leq_{|\mu|+\varepsilon} a^{\mu\perp} y$$

and $(\mathsf{LA}_\mu)_{\varepsilon+1}$ is equivalent to $(\mathsf{RA}_\mu)_\varepsilon$. For the monotonicy axiom we have

$$(\mathsf{AM}_\mu)_\varepsilon \Leftrightarrow \mathsf{AM}_\mu \, .$$

We will prove $(1.1)_\varepsilon - (1.4)_\varepsilon$. First notice that

$$a^{\mu\perp} c \leq_{\varepsilon+|\mu|} a^{\mu\perp} c \;\&\; (\mathsf{LA}_\mu)_\varepsilon \Longrightarrow a^\mu a^{\mu\perp} c \leq_\varepsilon c$$

and therefore:

$$\mathsf{AM}_\lambda \;\&\; |\lambda| = 0 \;\&\; (\mathsf{LA}_\mu)_\varepsilon \Longrightarrow b^\lambda a^\mu a^{\mu\perp} c \leq_\varepsilon b^\lambda c$$

Using that, from $(1)_\varepsilon$ with $a^{\mu\perp} c$ replacing c one derives $a^\mu b^\lambda a^{\mu\perp} c \leq_\varepsilon b^{\lambda\perp} c$. This implies $(2)_\varepsilon$.

Having $\mu\perp$ instead of μ , $\varepsilon + |\mu| + 1$ instead of ε and leaving λ unchanged, previous proof gives $(2)_\varepsilon \Longrightarrow (1)_\varepsilon$. (This is possible because the replacements leave $(\mathsf{LA}_\mu)_\varepsilon$ unchanged.) This proves $(1.1)_\varepsilon$.

Substituting $\mu \mapsto \lambda$, $\lambda \mapsto \mu$, $\varepsilon \mapsto \varepsilon + 1$ in $(1.1)_\varepsilon$, we get $(1.2)_\varepsilon$.

Let us prove $(1.3)_\varepsilon$. First:

$$(\mathsf{LA}_{\mu^*})_\varepsilon \Longrightarrow (u^\mu a^{\lambda\perp} b \leq_\varepsilon c \Leftrightarrow u \leq_{\varepsilon+|\mu^*|} (a^{\lambda\perp} b)^{\mu^*\perp} c) \qquad (*)$$

(we used the graphical equality $u^\mu a^{\lambda\perp} b \doteq (a^{\lambda\perp} b)^{\mu^*} u$). Second:

$$(\mathsf{RA}_\lambda)_\varepsilon \;\&\; |\lambda| = 0 \;\&\; (\mathsf{LA}_{\mu^*})_\varepsilon \Longrightarrow (a^{\lambda\perp} u^\mu b \leq_\varepsilon c \Leftrightarrow b^{\mu^*} u \leq_\varepsilon a^\lambda c \Leftrightarrow u \leq_{\varepsilon+|\mu^*|} b^{\mu^*\perp} a^\lambda c) \qquad (**)$$

With $|\lambda| = 0$, from $(3)_\varepsilon$ it follows that the left part of the equivalence in (*) implies first formula from the equivalent ones in (**). Therefore, the right part of the equivalence in (*) implies the last equivalent formula in (**). Having $u = (a^{\lambda\perp} b)^{\mu^*\perp} c$, we get $(3)_\varepsilon \Longrightarrow (4)_\varepsilon$. Using $(4)_\varepsilon$, from the right part of the equivalence in (*) one derives the last equivalent formula in (**), and going in other direction, we get $(3)_\varepsilon$ (using $c = u^\mu a^{\lambda\perp} b$).

From $(1.3)_\varepsilon$, substituting $\mu \mapsto \lambda$, $\lambda \mapsto \mu$, $\varepsilon \mapsto \varepsilon + 1$, we get $(1.4)_\varepsilon$.

Let us prove (1.5). From (1), substituting $a \mapsto c^{\mu^*\perp} b$, $b \mapsto c^{\lambda^*\perp} a$, $c \mapsto c$ we have

$$(c^{\mu^*\perp} b)^\mu (c^{\lambda^*\perp} a)^\lambda c \leq (c^{\lambda^*\perp} a)^\lambda (c^{\mu^*\perp} b)^\mu c \qquad (***)$$

Since

$$\mathsf{RA}_{\lambda^*} \;\&\; \mathsf{PO1} \Longrightarrow a \leq c^{\lambda^*} c^{\lambda^*\perp} a$$

and

$$\mathsf{LA}_{\mu^*} \;\&\; \mathsf{PO1} \Longrightarrow c^{\mu^*} c^{\mu^*\perp} b \leq b$$

then substituting $(c^{\lambda^*\perp}a)^\lambda c \doteq c^{\lambda^*}c^{\lambda^*\perp}a$ for a on the left side of (***), and on the right side $(c^{\mu^*\perp}b)^\mu c \doteq c^{\mu^*}c^{\mu^*\perp}b$ for b, using AM_μ, AM_λ, and $|\mu| = |\lambda| = 0$, we get $(c^{\mu^*\perp}b)^\mu a \le (c^{\lambda^*\perp}a)^\lambda b$, which is the same as (6). From (6), substituting $c \mapsto c^{\lambda^*}b$, $b \mapsto c^{\mu^*}a$, $c \mapsto c$, we get $(c^{\mu^*\perp}c^{\mu^*}a)^\mu c^{\lambda^*}b \le (c^{\lambda^*\perp}c^{\lambda^*}b)^\lambda c^{\mu^*}a$, i.e $(c^{\lambda^*}b)^{\mu^*}c^{\mu^*\perp}c^{\mu^*}a \le (c^{\mu^*}a)^{\lambda^*}c^{\lambda^*\perp}c^{\lambda^*}b$. From this, using the inequalities $a \le_{\mu^*} c^{\mu^*\perp}c^{\mu^*}a$, $c^{\lambda^*\perp}c^{\lambda^*}b \le_{\lambda^*} b$ derived from LA_{μ^*} and RA_{λ^*}, and applying AM_{μ^*}, AM_{λ^*}, we come to (1).

2.6 *Corollary.* If μ is left, λ is right, and $|\mu| = |\lambda| = 0$, the axioms $\mathsf{A1}$ imply equivalence of the inequalities (1)–(6).

Proof. As it was noticed in 2.2 and 2.3, $\mathsf{A1} \Longrightarrow \mathsf{AM}_\mu$ for all $\mu \in \Omega$. Also $\mathsf{A1} \Longrightarrow \mathsf{LA}_\mu$ for the left μ and $\mathsf{A1} \Longrightarrow \mathsf{RA}_\mu$ for the right μ. Moreover, $*$ preserves the direction of adjointness for symbols in Ω. Therefore, the premises of the implications in Theorem 1 are satisfied.

2.7 We are going to see what the formulas (1)-(6) are for particular μ and λ. There are four left μ for which $|\mu| = 0$ (namely $\cdot ?$, $? \cdot$, $\ddots ?$, $? \iddots$), and four right λ for which $|\lambda| = 0$ ($\times ?$, $? \times$, $\backslash ?$, $?/$), therefore, there are 16 classes of equivalent formulas (1)-(6). For given μ and λ, let $K_{\mu\lambda}$ be the list of (1)-(6). Consider the following classes $K_{\mu\lambda}$.

$\mathsf{I_R} = K_{\mu\lambda}$, where $\mu = (?\cdot)$, $\lambda = (\times ?)$.

(1)	$(b \times c) \cdot a \le b \times (c \cdot a)$	(4)	$(a \ddots b)\backslash c \le b\backslash(a \times c)$
(2)	$a \times (c/b) \le (a \times c)/b$	(5)	$(c \cdot b) \iddots a \le c \iddots (a/b)$
(3)	$a \ddots (c \cdot b) \le (a \ddots c) \cdot b$	(6)	$a \cdot (c\backslash b) \le (a \iddots c) \times b$

$\mathsf{II_R} = K_{\mu\lambda}$, where $\mu = (?\cdot)$, $\lambda = (\backslash ?)$.

(1)	$(b\backslash c) \cdot a \le b\backslash(c \cdot a)$	(4)	$(a \cdot b)\backslash c \le b\backslash(a\backslash c)$
(2)	$a\backslash(c/b) \le (a\backslash c)/b$	(5)	$(a/b)/c \le a/(c \cdot b)$
(3)	$a \cdot (c \cdot b) \le (a \cdot c) \cdot b$	(6)	$a \cdot (c\backslash b) \le (c/a)\backslash b$

$\mathsf{III_L} = K_{\mu\lambda}$, where $\mu = (\ddots ?)$, $\lambda = (? \times)$.

(1)	$a \ddots (c \times b) \le (a \ddots c) \times b$	(4)	$b \iddots (c \times a) \le (b \iddots a) \iddots c$
(2)	$(a \times c) \times b \le a \times (c \times b)$	(5)	$a \ddots (b \ddots c) \le (b \times a) \ddots c$
(3)	$(a \ddots c) \iddots b \le a \ddots (c \iddots b)$	(6)	$(c \iddots b) \ddots a \le b \times (c \ddots a)$

$\mathsf{IV_R} = K_{\mu\lambda}$, where $\mu = (? \iddots)$, $\lambda = (\backslash ?)$.

(1)	$(b\backslash c) \iddots a \le b\backslash(c \iddots a)$	(4)	$(a\backslash c) \ddots b \le c \ddots (a \cdot b)$
(2)	$b\backslash(c \times a) \le (b\backslash c) \times a$	(5)	$(a \times b)/c \le a/(c \iddots b)$
(3)	$a \cdot (c \iddots b) \le (a \cdot c) \iddots b$	(6)	$a \iddots (b \ddots c) \le (c/a)\backslash b$

We will denote classes that are symmetrical to the above ones as $\mathsf{I_L}$, $\mathsf{II_L}$, $\mathsf{III_R}$, $\mathsf{IV_L}$. If $\mu = (\omega ?)$, let us define $\mu^\sim = (? \omega^\sim)$, where $\omega^\sim$ is an operation symmetrical to ω (see 1.5) (analogously, $\mu^\sim = (\omega^\sim ?)$, for $\mu = (?\omega)$. It is easy to see that $K^\sim_{\mu\lambda}$ – the class of formulas symmetrical to the formulas in $K_{\mu\lambda}$ – coincides with $K_{\mu^\sim\lambda^\sim}$. Therefore, $\mathsf{I_L} = K_{(?\cdot)^\sim(\times ?)^\sim} = K_{(\cdot ?)(? \times)}$, etc.

2.8 Since in each class I - III there is one axiom from $\mathsf{A2}$ or $\mathsf{A3}$, all formulas from these six classes are derivable from $\mathsf{A1}$ - $\mathsf{A3}$.

2.9 Let us find some other axiomatizations equivalent to A1 - A4 .

Theorem 2. The system A1 - A4 is equivalent to the following system defined with preorder axioms, monotonicity axioms for all operations, axioms of neutrality for $\mathbf{1}$ and $\mathbf{I}$, and axioms of the insertion-cancellation type

(IC)	$a \cdot (a\backslash x) \leq x$	$x \leq a\backslash(a \cdot x)$
	$(x/a) \cdot a \leq x$	$x \leq (x \cdot a)/a$
	$a \therefore (a \times x) \leq x$	$x \leq a \times (a \therefore x)$
	$(x \times a) : a \leq x$	$x \leq (x : a) \times x)$

The other axioms may vary with the choice of one inequality from each of the classes I - III and symmetrical to these classes.

Proof. It follows from 2.6, 2.8, and a well-known fact that LA_μ can be replaced with AM_μ, $\mathsf{AM}_{\mu\perp}$, and inequalities $a^\mu a^{\mu\perp} x \leq x$, $x \leq a^{\mu\perp} a^\mu x$.

2.10 Let us write an axiomatization obtained by weakening the axioms A1 . Namely, in A1.1 and A1.2 put $x = \mathbf{1}$, and in A1.3 and A1.4 put $x = \mathbf{I}$ and $y = \mathbf{1}$. The set of axioms obtained we will denote as $\mathsf{A1}^-$.

Theorem 3. The system A1 - A4 is equivalent to the conjunction of

$\mathsf{A1}^-$	1. $a \leq b \Leftrightarrow \mathbf{1} \leq a\backslash b$	2. $a \leq b \Leftrightarrow \mathbf{1} \leq b/a$
	3. $\mathbf{1} \leq a \Leftrightarrow a \therefore \mathbf{1} \leq \mathbf{I}$	4. $\mathbf{1} \leq a \Leftrightarrow \mathbf{1} : a \leq \mathbf{I}$

(instead of A1),

$\mathsf{A2}^+$	1. $(a \cdot b)\backslash c = b\backslash(a\backslash c)$	2. $c/(b \cdot a) = (c/a)/b$
	3. $(a \times b) \therefore c = b \therefore (a \therefore c)$	4. $c : (b \times a) = (c : a) : b$

(instead of A2), A3 , A4 , and

A5	1. $(a \therefore b)\backslash\mathbf{I} \leq b\backslash a$	2. $\mathbf{I}/(b : a) \leq a/b$
	3. $b \therefore a \leq (a\backslash b) \therefore \mathbf{1}$	4. $a \therefore b \leq \mathbf{1} : (b/a)$

Proof. 2.8 shows that all the formulas are consequences of A1 - A4 :

$\mathsf{II_R}.4$ & $\mathsf{II_L}.5$	$\Longrightarrow$	$\mathsf{A2}^+.1$	$\mathsf{II_L}.4$ & $\mathsf{II_L}.5$	$\Longrightarrow$	$\mathsf{A2}^+.2$
$\mathsf{III_L}.4$ & $\mathsf{III_R}.5$	$\Longrightarrow$	$\mathsf{A2}^+.3$	$\mathsf{III_R}.4$ & $\mathsf{III_L}.5$	$\Longrightarrow$	$\mathsf{A2}^+.4$
$\mathsf{I_R}.4$ (for $c = \mathbf{I}$)	$\Longrightarrow$	A5.1	$\mathsf{I_L}.4$ (for $c = \mathbf{I}$)	$\Longrightarrow$	A5.2
$\mathsf{I_R}.5$ (for $c = \mathbf{1}$)	$\Longrightarrow$	A5.3	$\mathsf{I_L}.5$ (for $c = \mathbf{1}$)	$\Longrightarrow$	A5.4

In the other direction, we prove A1.1 :

$$a \cdot x \leq y \overset{\mathsf{A1}^-.1}{\Longleftrightarrow} \mathbf{1} \leq (a \cdot x)\backslash y \overset{\mathsf{A2}^+.1}{\Longleftrightarrow} \mathbf{1} \leq x\backslash(a\backslash y) \overset{\mathsf{A1}^-.1}{\Longleftrightarrow} x \leq a\backslash b.$$

A1.3 follows from the equivalence $b \leq a \Leftrightarrow a \therefore b \leq \mathbf{I}$ proven below dualizing the previous proof.

$$b \leq a \overset{\mathsf{A1}^-.1}{\Longrightarrow} \mathbf{1} \leq b\backslash a \overset{\mathsf{A1}^-.3}{\Longrightarrow} (b\backslash a) \therefore \mathbf{1} \leq \mathbf{I} \overset{\mathsf{A5.3}}{\Longrightarrow} a \therefore b \leq \mathbf{I} \overset{\mathsf{A1}^-.1}{\Longrightarrow} \mathbf{1} \leq (a \therefore b)\backslash\mathbf{I} \overset{\mathsf{A5.1}}{\Longrightarrow} \mathbf{1} \leq b\backslash a \overset{\mathsf{A1}^-.1}{\Longrightarrow} b \leq a$$

Axioms A2 follow from $\mathsf{A2}^+$ by 2.6.

1.3 Negations and consequences of additional axioms

3.1 *Convention*: Instead of $\backslash$, $/$, $\times$, $\therefore$, $\because$, $\mathbf{I}$, we will write $\supset$, $\subset$, $+$, $\div$, $-$, $\mathbf{0}$ respectively, i.e.

$$\begin{array}{rclcrcl} a \supset b & \rightleftharpoons & a\backslash b & & b \subset a & \rightleftharpoons & b/a \\ a \div b & \rightleftharpoons & a \therefore b & & b - a & \rightleftharpoons & b \because a \\ a + b & \rightleftharpoons & a \times b & & \mathbf{0} & \rightleftharpoons & \mathbf{I} \end{array}$$

3.2 We introduce four negations:

$$\begin{array}{rclcrcl} \neg_L a & \rightleftharpoons & a \supset \mathbf{0} & & \neg_R a & \rightleftharpoons & \mathbf{0} \subset a \\ -_L a & \rightleftharpoons & a \div \mathbf{1} & & -_R a & \rightleftharpoons & \mathbf{1} - a \end{array}$$

3.3 The following theorem shows interaction between the negations, the constants $\mathbf{0}$, $\mathbf{1}$, and the system operations.

Theorem 4. A1-A4 implies:

1. $\neg_L a + b \le a \supset b \le -_R a + b$ $\qquad$ $b \cdot -_R a \ge b - a \ge b \cdot \neg_L a$
2. $\neg_L a \le -_R a$
3. $a \le \neg_R \neg_L a$ $\qquad$ $a \ge -_L -_R a$
4. $\neg_L \mathbf{1} = \mathbf{0} = -_R \mathbf{1}$ $\qquad$ $-_R \mathbf{0} = \mathbf{1} = \neg_L \mathbf{0}$
5. $\neg_L b + \neg_L a \le \neg_L(a \cdot b) \le -_R b + \neg_L a$ $\qquad$ $-_R a \cdot -_R b \ge -_R(b + a) \ge -_R a \cdot \neg_L b$
6. $\neg_L b \cdot \neg_L a \le \neg_L(a + b) \le -_R b \cdot -_R a$ $\qquad$ $-_R a + -_R b \ge -_R(b \cdot a) \ge \neg_L a + \neg_L b$
7. $\neg_R(b - a) \le a \subset b$ $\qquad$ $-_R(b \subset a) \ge a - b$
8. $\neg_R \neg_L \neg_R a = \neg_R a$ $\qquad$ $-_L -_R -_L a = -_L a$
9. $a \supset b \le \neg_L a \subset \neg_L b$ $\qquad$ $b - a \ge -_R b \div -_R a$
10. $\neg_L -_L a \cdot \neg_L -_L b \le \neg_L -_L(a \cdot b)$ $\qquad$ $-_L \neg_L a + -_L \neg_L b \ge -_L \neg_L(a + b)$
11. $\neg_L -_L(a \supset b) \le \neg_L -_L a \supset \neg_L -_L b$ $\qquad$ $-_R \neg_R(b - a) \ge -_R \neg_R b - -_R \neg_R a$
12. $\mathbf{1} \supset a = a$ $\qquad$ $a - \mathbf{0} = a$
13. $\mathbf{0} \supset a = 1 + a$ $\qquad$ $a - \mathbf{1} = a \cdot \mathbf{0}$

and inequalities symmetrical to the ones above.

Proof. We will prove only inequalities of the left column.

(1) follows from $\mathsf{I_L}.2$ with $c = \mathbf{I}$, and from $\mathsf{I_R}.6$ with $a = \mathbf{1}$.

(2) follows from (1) with $b = \mathbf{0}$.

(3) follows from $a \le \neg_R \neg_L a \Leftrightarrow a \cdot (a \supset \mathbf{0}) \le \mathbf{0}$.

(4) For this, substitute $a = \mathbf{1}$ in (2) and get $\neg_L \mathbf{1} \le \mathbf{1} - \mathbf{1} \le \mathbf{0}$. Also, $\mathbf{0} \le \mathbf{1} \supset \mathbf{0}$, because $\mathbf{1} \cdot \mathbf{0} \le \mathbf{0}$. Therefore, $\neg_L \mathbf{1} = \mathbf{0}$. Since $\mathbf{1} \le \mathbf{0} + \mathbf{1}$, then $\mathbf{1} - \mathbf{1} \le \mathbf{0}$. On the other hand, from the right column of (1) , using $b = a = \mathbf{1}$, we get $\mathbf{1} \cdot \neg_L \mathbf{1} \le \mathbf{1} - \mathbf{1}$, i.e. $\mathbf{0} \le \mathbf{1} - \mathbf{1}$. Therefore, $\mathbf{0} = -_R \mathbf{1}$.

(5) $\mathsf{II_R}.4$ (with $c = 0$) $\Longrightarrow \neg_L(a \cdot b) \le b \supset \neg_L a$. $\mathsf{II_L}.5$ (with $a = 0$) $\Longrightarrow b \supset \neg_L a \le \neg_L(a \cdot b)$. So, we have $\neg_L(a \cdot b) = b \supset \neg_L a$. Applying inequalities from (1) to $b \supset \neg_L a$, we get (5) .

(6) The left inequality follows from $(a + b) \cdot (\neg_L b \cdot \neg_L a) \le (a + (b \cdot (b \supset \mathbf{0}))) \cdot (a \supset \mathbf{0}) \le a \cdot (a \supset \mathbf{0}) \le \mathbf{0}$. Right inequality we get first increasing $\neg_L(a + b)$ to $-_R(a + b)$ (with the help of (2)), and then applying (5) (right column).

(7) follows from $\mathsf{I_L}.4$ when $c = \mathbf{0}$.

(8) Use (3) where $a \rightleftharpoons \neg_R a$, and an inequality symmetrical to (3) applying $\neg_R$ which reverses the order.
(9) Since $a \cdot ((a \supset b) \cdot \neg_L b) \leq b \cdot (b \supset \mathbf{0}) \leq \mathbf{0}$, one gets $(a \supset b) \cdot \neg_L b \leq \neg_L a$.
(10) (6) and an inequality symmetrical to (6) (using the antitonicity of $\neg_L$) give $\neg_L -_L a \cdot \neg_L -_L b \leq \neg_L(-_L b + -_L a) \leq \neg_L -_L(a \cdot b)$
(11) Replace in (10) b with $a \supset b$, and $\neg_L -_L(a \cdot (a \supset b))$ with bigger $\neg_L -_L b$.
(12) $\mathbf{1} \supset a \leq \mathbf{1} \cdot (\mathbf{1} \supset a) \leq a$ gives (12).
(13) follows from (1) and (4) with $a = \mathbf{0}$.

3.4. Let us see what happens when we add to the system A1-A4 some other axioms.

3.4.1. Consider the *laws of excluded middle*:

$\mathbf{1} \leq \neg_L a + a$ (the left law of excluded middle), $\mathbf{1} \leq a + \neg_R a$ (the right law of excluded middle).

The left law of excluded middle is equivalent to $-_R a \leq \neg_L a$, i.e. to the reverse of (2) from theorem 4. Therefore, adding the laws of excluded middle leads to the coincidence of negations $-_R a = \neg_L a$, which we denote by $\sim_L a$. Similarly, $\sim_R a = \neg_R a = -_L a$. Operations $\sim_L$ and $\sim_R$ we will call the *left* and *right complement* respectively. With these negations, we can express the implication and the subtraction (with the help of (1) taken from theorem 4):

$$\begin{aligned} a \supset b &= \sim_L a + b \\ b - a &= b \cdot \sim_L a \end{aligned} \tag{\dagger}$$

Inequalities (3), (5) and (6) from theorem 4 imply:

$$\sim_R \sim_L a = a \quad \text{(cancellation of the double oppositely directed complements)}$$

$$\left.\begin{aligned} \sim_L(a+b) &= \sim_L b \cdot \sim_L a \\ \sim_L(b \cdot a) &= \sim_L b + \sim_L a \end{aligned}\right\} \text{(de Morgan laws)}$$

Obviously, the symmetrical equalities also hold. These equalities make it possible to express all operations with the multiplications and the complements.

Furthermore, the inequalities from the groups $\mathsf{IV_L}$ and $\mathsf{IV_R}$ become derivable. Indeed, $\mathsf{IV_R}.2$ becomes a valid statement: $\sim_L b + (c + a) \leq (\sim_L b + c) + a$. By virtue of 2.6, one implies that all inequalities in $\mathsf{IV_R}$ are derivable. On the other hand, the left law of excluded middle follows from any formula of $\mathsf{IV_R}$. Substituting $c = \mathbf{0} = \mathbf{I}$ in $\mathsf{IV_R}.2$, we get $a \supset b \leq \sim_L a + b$, from where with $b = a$ we get the left law of excluded middle.

3.4.2. In the signature $(\leq,\ \cdot,\ +,\ \sim_L,\ \sim_R,\ \mathbf{0},\ \mathbf{1})$ the class of systems satisfying A1-A4 and the laws of excluded middle can be postulated with the following axioms:

$\tilde{\mathsf{A}}1.$

$$\begin{aligned} &a \cdot \sim_L a \leq \mathbf{0} && \sim_R a \cdot a \leq \mathbf{0} \\ &\mathbf{1} \leq a + \sim_R a && \mathbf{1} \leq \sim_L a + a \end{aligned}$$

together with A2, A3, A4 and the monotonicity axioms for $\cdot$ and $+$.

This follows from the fact that, having defined implications and subtractions in a system $\mathbf{B} = (B;\ \leq,\ +,\ \sim_L,\ \sim_R,\ \mathbf{0},\ \mathbf{1})$ using the equalities of (†) and their symmetrical variants, we get a system $\mathbf{B}^+$ satisfying A1-A4. The transitions $\mathbf{B} \mapsto \mathbf{B}^+$ and $\mathbf{A} \mapsto \tilde{\mathbf{A}}$ are mutually inverse. Here $\tilde{\mathbf{A}}$ denotes a system $\mathbf{A}$ without implications and subtractions, but with complements.

3.4.3. Let us call the following inequalities as the *axioms of complement*: $\mathbf{1} \le a + \neg_L a$, $\mathbf{1} \le \neg_R a + a$ (remember: $+$ is still not commutative). Adding these two axioms to A1-A4 makes all four negations equal. Indeed, the axioms imply $-_L a \le \neg_L a$ and $-_R a \le \neg_R a$. Therefore, $\neg_R a \le -_L a \le \neg_L a \le -_R a \le \neg_R a$. Let

$$\bar{a} \rightleftharpoons \sim_L a = \sim_R a$$

Then the axioms $\tilde{\mathsf{A}}1$ become:

$\bar{\mathsf{A}}1.$

$$a \cdot \bar{a} \le \mathbf{0} \qquad \bar{a} \cdot a \le \mathbf{0}$$
$$\mathbf{1} \le a + \bar{a} \qquad \mathbf{1} \le \bar{a} + a$$

3.4.4. If, in addition to the axioms of complement, we consider the following one: $x \cdot y = x + y$, then $\mathbf{0} = \mathbf{1}$, and the formulas $\bar{\mathsf{A}}1$ express that $\bar{a}$ is the inverse element. Therefore, we get the axioms of ordered groups.

3.4.5. Now consider the axiom $\mathbf{0} \le x \le \mathbf{1}$. It is equivalent to the equalities $\mathbf{0} \cdot a = \mathbf{0}$, $b + \mathbf{1} = \mathbf{1}$, and to the inequalities $a_1 \cdot a_2 \le a_i$, $a_i \le a_1 + a_2$, $i = 1,\ 2$. Indeed, A3.1 (for $b = \mathbf{0}$) $\Longrightarrow \mathbf{0} \le a$, similarly A3.1 (for $a = \mathbf{1}$) $\Longrightarrow b \le \mathbf{1}$.

Adding the axiom $\mathbf{0} \le x \le \mathbf{1}$ and idempotency axioms $x \cdot x = x$, $x + x = x$ to A1-A4 gives the axiomatization of Heyting-Brouwer algebras.

3.4.6. A1-A4 plus commutativity of $\cdot$ and $+$, plus $\mathbf{0} \le x \le \mathbf{1}$, and plus the laws of excluded middle form the axiomatization of L°-algebras considered in [5]. Notice that commutativity follows from $K_{\mu\lambda}$ for appropriate μ and λ. For example, we can obtain the inequality $x \cdot y \le y \cdot x$ from the formula (3) of $K_{\mu\lambda}$ assuming $\mu = (\cdot ?)$, $\lambda = (\supset ?)$, and $c = \mathbf{1}$.

3.4.7. We call equalities $x^q + x^q = x^q$ and $qx \cdot qy = qx$ as q*-idempotency*, where q is a positive integer and

$$x^q \rightleftharpoons \underbrace{x \cdot \dots \cdot x}_{q \text{ times}} \qquad qx \rightleftharpoons \underbrace{x + \ldots + x}_{q \text{ times}}$$

Adding q-idempotency to the axioms of L°-algebras, one gets the L°_q-algebras (see [5]). The case $q = 2$ gives rise to the class L°_2, which is adequate to three-valued Łukasiewicz logic. If we add the axioms of q-idempotency to the group axioms, they will transform into $x^q = \mathbf{1}$, so we get ordered periodical groups, or, simply, periodical groups, because in this case any order is just equality ($\mathbf{1} \le x \Longrightarrow x \le x^2 \le \ldots \le x^q = \mathbf{1}$).

3.4.8. Axioms A1-A4 plus Chang's axioms

$$(x - y) + y = (y - x) + x, \qquad y \cdot (y \supset x) = x \cdot (x \supset y)$$

(see [2]) imply that the order is a lattice. Disjunction and conjunction are defined as:

$$x \vee y \rightleftharpoons (x - y) + y \qquad x \wedge y \rightleftharpoons x \cdot (x \supset y)$$

Indeed, $(x - x) + x \le \mathbf{0} + x = x$, $x \le (x - y) + y$, $y \le (y - x) + x = (x - y) + y$. The operation $(x, y) \mapsto (x - y) + y$ is obviously monotonous in its first argument, and, from Chang's axioms, also in the second. Therefore, $(x \le c)\ \&\ (y \le c) \Longrightarrow (x - y) + y \le (c - c) + c \le c$, i.e. $(x - y) + y = \text{supremum}\{x, y\}$. We treat the conjunction analogously. The axioms of L°-algebras together with Chang's axioms give an axiomatization of Chang's MV-algebras.

3.5. In the table below "+" marks those axioms (properties) which it is enough to add to A1-A4 to obtain the class of algebraic systems from the appropriate row. Symbol "(+)" will denote the properties that hold in each system of the class in the appropriate row, whereas "–" symbolizes properties that do not hold at least in one system of the class.

Classes of algebras	Properties							
	1 axioms of complement	2 $x \cdot y = x + y$	3 commutativity	4 $0 \leq x$ $x \leq 1$	5 idempotency	6 2-idempotency	7 q-idempotency ($q \geq 3$)	8 Chang's axioms
1. Heyting-Brouwer algebras	–	–	(+)	+	+	(+)	(+)	(+)
2. Boolean algebras	+	–	(+)	+	+	(+)	(+)	(+)
3. Ordered groups	+	+	–	–	–	–	–	–
4. Periodical groups of exponent q	+	+	–	–	–	–	+	–
5. L°-algebras adequate to a logic without contraction rules	+	–	+	+	–	–	–	–
6. Variety generated by 3-elements Łukasiewicz algebra	+	–	+	+	–	+	(+)	(+)
7. L_q°-algebras	+	–	+	+	–	–	+	–
8. Chang's MV-algebras	+	–	+	+	–	–	–	+

For example, the axioms A1-A4 plus the properties 4 and 5 give Heyting-Brouwer algebras, satisfying the properties 3, 6, 7, 8. Properies 1 and 2 are not valid in the whole class.

Symbols "(+)" and "–" do not require any additional explanation, except for the following cases.

The interval [0,1] of real numbers considered as a Łukasiewicz algebra is a Chang MV-algebra. There q-idempotency (for any q) does not hold. This explains the "–" symbol in the following table cells: (5,5), (5,6), (5,7), (8,5), (8,6), (8,7).

For L°-algebras

$$\forall x\,(x^q + x^q = x^q) \Longrightarrow \forall x\,(x^p + x^p = x^p) \qquad \text{for } p \geq q.$$

Indeed, simple L_q°-algebras can be characterized with the condition $\forall x \neq \mathbf{1}\,(x^q = \mathbf{0})$ (see [5]). From this one gets $\forall x \neq \mathbf{1}\,(x^p = \mathbf{0})$ when $p \geq q$. Therefore, every simple L_q°-algebra is a simple L_p°-algebra when $p \geq q$. Furthermore, every L_q°-algebra can be represented as a subdirect product of simple L_q°-algebras ([5]). Thus, every L_q°-algebra is an L_p°-algebra ($p \geq q$) as well. That explains "(+)" in the (6,7)-cell.

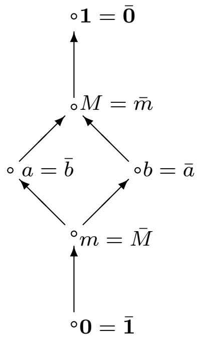
As it was shown in [4], there are only two simple L_2°-algebras: $\{\mathbf{0}, \mathbf{1}\}$ — two-element Boolean algebra, and $\{0, \frac{1}{2}, 1\}$ — three-element Łukasiewicz algebra. Therefore, the variety of L_2°-algebras is generated by the three-element Łukasiewicz algebra. Thus, in the cells (6,7) and (6,8) we have "(+)".

In the case of $q \geq 3$, the q-element Łukasiewicz algebra

$$\{0,\ \frac{1}{q-1},\ \frac{2}{q-1}, \ldots, \frac{q-2}{q-1},\ 1\},$$

which is an L_{q-1}°-algebra, does not satisfy 1- and 2-idempotency. Therefore, there are "–" symbols in (7,5) and (7,6).

The set $\{0,\ m,\ a,\ b,\ M,\ 1\}$ with operations $x \mapsto \bar{x}$, $(x,y) \mapsto x \cdot y$, $(x,y) \mapsto x + y$ defined below and ordered as shown



$$x \cdot y = \begin{cases} \mathbf{0}, & \text{if } x \leq \bar{y} \\ x, & \text{if } y = \mathbf{1} \\ y, & \text{if } x = \mathbf{1} \\ m, & \text{otherwise} \end{cases} \qquad x + y = \overline{(\bar{x} \cdot \bar{y})}$$

is a simple L_q°-algebra (for any $q \geq 3$) which satisfies $(a - b) + b = a \cdot \bar{b} = a^2 + b = m + b = \mathbf{0} \neq a \vee b$. Therefore, we have "–" in (5,8) and (7,8).

1.4 Deduction theorem and normal filters

4.1. Now we are going to find a calculus which would describe the class of formulas greater than $\mathbf{1}$ in a free system satisfying A1-A4. This free system is obtained in the following way.

Consider a set of formulas generated from a set of propositional letters (the generators of the system) using the operations of the A1-A4 system. If we consider the postulated inequalities in the system as axiom schemes, and the implications (equivalence is two implications) as rules of derivation, we get a *free calculus* which imposes an order on the set of formulas. So the axiom schemes are PO1, A2, A3, A4, the rules of derivation are PO2, A1. Inequalities of the sort $X \leq Y$, where X and Y are any formulas constructed as it was described above, we will state $X \longrightarrow Y$ for *sequents*. Thus, $X \leq Y$ holds in the *free system* iff the sequent $X \longrightarrow Y$ is derivable in the free calculus described. An arbitrary system satisfying A1-A4 can be obtained from the free system by adding a set of particular sequents as new axioms that can be used in a proof as many times as it is required.

4.2 We describe a calculus **J** generating the set of all elements of the free system greater than $\mathbf{1}$. The objects of the calculus are formulas in the sense specified in 4.1. Letters X, Y, Z denote arbitrary formulas. Notations $\supset$, $\subset$, $-$, $\div$, $+$, $\mathbf{0}$, $\neg_L$, $\neg_R$, $-_L$, $-_R$ are described in 3.1 and 3.2.

Axioms and axiom schemes of the calculus **J***.*

$L1.\quad \mathbf{1}$	
$L2.\ X \supset (\mathbf{1} \supset X)$	$(X \subset \mathbf{1}) \subset X$
$L2.\ ((X \cdot Y) \supset Z) \supset (Y \supset (X \supset Z))$	$(Z \subset (Y \cdot X)) \subset ((Z \subset X) \subset Y)$
$L4.\ (Y \supset (X \supset Z)) \supset ((X \cdot Y) \supset Z)$	$((Z \subset X) \subset Y) \subset (Z \subset (Y \cdot X))$
$L5.\ (Y \supset Z) \supset ((X \supset Y) \supset (X \supset Z))$	
$L6.\ ((X \supset Y) + Z) \supset (X \supset (Y + Z))$	$(Z + (Y \subset X)) \supset ((Z + Y) \subset X)$
$L7.\ \mathbf{1} \supset (\mathbf{0} + \mathbf{1})$	$\mathbf{1} \supset (\mathbf{1} + \mathbf{0})$
$L8.\ (X + \mathbf{0}) \supset X$	$(\mathbf{0} + X) \supset X$
$L9.\ (X - (Y + Z)) \supset ((X - Z) - Y)$	$((Z + Y) \div X) \supset (Y \div (Z \div X))$
$L10.\ ((X - Z) - Y) \supset (X - (Y + Z))$	$(Y \div (Z \div X)) \supset ((Z + Y) \div X)$
$L11.\ \neg_R(X - Y) \supset (Y \subset X)$	$\neg_L(Y \div X) \supset (X \supset Y)$
$L12.\ (Y - X) \supset -_R(X \subset Y)$	$(X \div Y) \supset -_L(Y - X)$

Rules of derivation of **J**

$$(\mathtt{R1})_{\mathtt{L}}\ \frac{X \qquad X \supset Y}{Y} \qquad\qquad (\mathtt{R1})_{\mathtt{R}}\ \frac{Y \subset X \qquad X}{Y}$$

$$(\mathtt{R2})_{\mathtt{L}}^{A}\ \frac{X \subset A}{A \supset X} \qquad\qquad (\mathtt{R2})_{\mathtt{R}}^{A}\ \frac{A \supset X}{X \subset A}$$

$$(\mathtt{R3})_{\mathtt{L}}\ \frac{X}{\neg_L -_L X} \qquad\qquad (\mathtt{R2})_{\mathtt{R}}\ \frac{X}{\neg_R -_R X}$$

4.3 Let S be a set of sequents, and S^{ϕ} a set of all formulas $A \supset B$ such that $A \longrightarrow B$ is in S.

Theorem 5. A sequent $X \longrightarrow Y$ is derivable from a set of sequents S (in the free calculus) iff a formula $X \supset Y$ is derivable in **J** from the set of formulas S^{ϕ}.

Proof. It is enough to show that if $X \longrightarrow Y$ is derivable from S in the calculus corresponding to the axiomatization in 2.10, then $X \supset Y$ is derivable from S^{ϕ} in the calculus **J**.

The axiom $X \longrightarrow X$ becomes the formula $X \supset X$ which can be derived in the following way:

$$
\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{L2}{X \supset (\mathbf{1} \supset X)} \qquad \cfrac{L4}{(X \supset (\mathbf{1} \supset X)) \supset (\mathbf{1} \cdot X \supset X)}}{\mathbf{1} \cdot X \supset X}\,(\mathtt{R1})_{\mathtt{L}}}{X \subset \mathbf{1} \cdot X}\,(\mathtt{R2})_{\mathtt{R}}^{X} \qquad L4}{(X \subset X) \subset \mathbf{1}}\,(\mathtt{R1})_{\mathtt{R}} \qquad \mathbf{1}}{\cfrac{X \subset X}{X \supset X}\,(\mathtt{R2})_{\mathtt{L}}^{X}}\,(\mathtt{R1})_{\mathtt{R}}
$$

Transitivity is derived using $L5$. The rules $\mathsf{A1}^{-}$ become rules derivable from $L1$, $L2$ and $\mathtt{R1}$-$\mathtt{R3}$. The axioms $\mathsf{A2}^{+}$ are translated into $L3$, $L4$, $L9$, $L10$. The axioms $\mathsf{A3}$ (in fact, formulas $\mathsf{I}_{\mathsf{R}}(2)$ and $\mathsf{I}_{\mathsf{L}}(2)$ from 2.7 are equivalent to $\mathsf{A3}$ in the presence of $\mathsf{A1}$ (see 2.6 and 2.10)) become $L6$. $\mathsf{A5}$ become $L11$, $L12$. The formula $\mathbf{1} \cdot X \supset X$ follows from $L2$ and $L4$. Analogously, one gets $X \subset X \cdot \mathbf{1}$ and then $X \cdot \mathbf{1} \supset X$ (using $\mathtt{R2}$). From the derivable formula $(\mathbf{1} \cdot X \subset X) \subset \mathbf{1}$ we get $\mathbf{1} \cdot X \subset X$, and from that $X \supset \mathbf{1} \cdot X$. The sequent $X \longrightarrow \mathbf{0} + X$ follows from $\mathbf{1} \longrightarrow \mathbf{0} + \mathbf{1}$ (indeed, $X \longrightarrow \mathbf{1} \cdot X \longrightarrow (\mathbf{0} + \mathbf{1}) \cdot X \longrightarrow \mathbf{0} + \mathbf{1} \cdot X \longrightarrow \mathbf{0} + X$).

4.4 Now consider the deduction theorem for **J**.

The rules

$$(\mathtt{R}^{*}\mathtt{2})_{\mathtt{L}}^{A}\ \frac{X}{A \supset (X \cdot A)} \qquad\qquad (\mathtt{R}^{*}\mathtt{2})_{\mathtt{R}}^{A}\ \frac{X}{(A \cdot X) \subset A}$$

are called the *rules of adjointness*. They are derivable in **J**. Indeed, the formula $((X \cdot A \subset A) \subset X) \subset (X \cdot A \subset X \cdot A)$ is the axiom $L4$. Therefore, the formula $(X \cdot A \subset A) \subset X$ is derivable in **J**. From X we can derive $(X \cdot A) \subset A$, and then using $(\mathtt{R2})_{\mathtt{L}}^{A}$, we get $A \supset (X \cdot A)$.

Let d be a proof (possibly from hypotheses). Let $d^{\mathbf{J}}$ be the sequence of hypotheses in d taken from left to right. For example, if d is obtained from d_1 and d_2 by $(\mathtt{R1})_{\mathtt{L}}$, then $d^{\mathbf{J}} = d_1^{\mathbf{J}} d_2^{\mathbf{J}}$. If d is just a formula A, $d^{\mathbf{J}} = A$ in the case A is not axiom, otherwise $d^{\mathbf{J}}$ is empty. We will write $\Gamma \overset{d}{\vdash} A$, if d is a proof in **J** of the formula A from hypotheses and $d^{\mathbf{J}} = \Gamma$. $\Gamma \vdash A$ means that there exists d such that $\Gamma \overset{d}{\vdash} A$. The result of application (step by step) of one-argument rules $R_1, R_2, \ldots, R_m$ to a formula H will be denoted as $R_m \ldots R_1 H$

The deduction theorem. Let $\Gamma H \Delta \overset{d}{\vdash} B$, and $R_1, \ldots, R_m$ be the sequence of $\mathtt{R2}$ or $\mathtt{R3}$ rules which are on the path below the given occurrence of H. Then there exists a proof d' such that $\Gamma\Delta \overset{d}{\vdash} (R'_m \ldots R'_1 H) \supset B$, where

$$R'_i = \begin{cases} (\mathtt{R^*2})^A_{\mathtt{L}}, & \text{if } R_i = (\mathtt{R2})^A_{\mathtt{L}} \\ (\mathtt{R^*2})^A_{\mathtt{R}}, & \text{if } R_i = (\mathtt{R2})^A_{\mathtt{R}} \\ R_i, & \text{if } R_i \text{ is } \mathtt{R3} \text{ rule} \end{cases}$$

Proof. If d is just one formula $B \rightleftharpoons H$, then $m = 0$, and the theorem is an obvious statement $\vdash B \supset B$. Let $\Gamma_1 H \Gamma_2 \overset{d_1}{\vdash} X$, and $\Delta \overset{d_2}{\vdash} X \supset B$, and the derivation d be obtained by $(\mathtt{R1})_{\mathtt{L}}$ from d_1 and d_2. Let $R_1, \ldots, R_m$ be the sequence of $\mathtt{R2}$ and $\mathtt{R3}$ rules taken for d_1 and a particular occurrence of H. By the induction hypothesis, $\Gamma_1\Gamma_2 \overset{d'_1}{\vdash} (R'_m \ldots R'_1 H) \supset X$. From d'_1 and d_2 we get d' in an obvious way.

Let $\Gamma \overset{d_1}{\vdash} X$, $\Delta_1 H \Delta_2 \overset{d_2}{\vdash} X \supset B$, and $R_1, \ldots, R_m$ be the sequence of the rules $\mathtt{R2}$ and $\mathtt{R3}$ for d_2 and H, and $H' \rightleftharpoons R'_m \ldots R'_1 H$. By the induction hypothesis, $\Delta_1\Delta_2 \overset{d'_2}{\vdash} H' \supset (X \supset B) \vdash X \cdot H' \supset B \vdash B \subset X \cdot H' \vdash (B \subset H') \subset X \vdash X \supset (B \subset H')$, and therefore $\Gamma\Delta_1\Delta_2 \overset{d'}{\vdash} B \subset H' \vdash H' \supset B$, for some d'.

Let $\Gamma_1 H \Gamma_2 \overset{d_1}{\vdash} B \subset A$, and $R_1, \ldots, R_m$ be the appropriate sequence of $\mathtt{R2}$ and $\mathtt{R3}$, $H' \rightleftharpoons R'_m \ldots R'_1 H$, and the proof d be obtained from d_1 by $(\mathtt{R2})^A_{\mathtt{L}}$. Again, the induction hypothesis gives $\Gamma_1\Gamma_2 \overset{d'_1}{\vdash} H' \supset (B \subset A)$, and from that $\Gamma_1\Gamma_2 \vdash H' \cdot A \supset B$. Since $\vdash A \cdot (A \supset H' \cdot A) \supset H' \cdot A$, we have $\Gamma_1\Gamma_2 \vdash (A \supset H' \cdot A) \supset (A \supset B)$, i.e. $\Gamma_1\Gamma_2 \vdash (\mathtt{R^*2})^A_{\mathtt{L}} H' \supset (A \supset B)$, which was necessary to prove.

Let $\Gamma_1 H \Gamma_2 \overset{d_1}{\vdash} B$, and d be obtained from d_1 applying $(\mathtt{R3})_{\mathtt{L}}$. Let $R_1, \ldots, R_m$ be the appropriate sequence of $\mathtt{R2}$ and $\mathtt{R3}$ for d and H. By the induction hypothesis, $\Gamma_1\Gamma_2 \overset{d'_1}{\vdash} (R'_m \ldots R'_1 H) \supset B \vdash \neg_L -_L((R'_m \ldots R'_1 H) \supset B)$. From (11) in theorem 4, we have $\Gamma_1\Gamma_2 \vdash ((\mathtt{R3})_{\mathtt{L}} R'_m \ldots R'_1 H) \supset \neg_L -_L B$.

Corollary. Suppose that a formula A is derivable from a set of formulas $S \cup H$. Then there exist formulas $H_1, \ldots, H_m$ derivable from H using $\mathtt{R^*2}$ and $\mathtt{R3}$, such that $H_1 \cdot \ldots \cdot H_m \supset A$ is derivable from the set S.

4.5 Now we introduce a notion generalizing the notion of the normal divisor in groups and the notion of the filter in Heyting-Brouwer algebras.

We will call a subset $\nabla \subset \mathbf{A}$ of an algebraic system satisfying $\mathsf{A1}$-$\mathsf{A4}$ a *normal filter* iff

1. $\mathbf{1} \in \nabla$
2. $x \in \nabla \;\&\; y \in \nabla \Longrightarrow x \cdot y \in \nabla$
3. $x \in \nabla \;\&\; x \leq y \Longrightarrow y \in \nabla$

4. $x \in \nabla \Longrightarrow a\backslash(x \cdot a) \in \nabla \,\&\, (a \cdot x)/a \in \nabla \quad \text{for } a \in A$
5. $x \in \nabla \Longrightarrow \neg_L -_L x \in \nabla \,\&\, \neg_R -_R x \in \nabla$

If $\mathbf{A} = (A;\ \leq_{\mathbf{A}},\ \cdot,\ \backslash,\ /,\ \mathbf{1},\ \times,\ \therefore,\ \because,\ \mathbf{I}\ )$ is an A1-A4 system, $MO(\mathbf{A})$ stands for the partially ordered (by inclusion) set of all preorders on $\mathbf{A}$ containing the preorder $\leq_{\mathbf{A}}$ and satisfying the monotonicity axioms AM (see 1.1). We will write $x \leq_\theta y$ iff $(x, y) \in \theta$ for $\theta \in MO(\mathbf{A})$. Let $NF(\mathbf{A})$ be the set of all normal filters on $\mathbf{A}$ ordered by inclusion.

Theorem 6. The sets $MO(\mathbf{A})$ and $NF(\mathbf{A})$ are complete lattices isomorphic to each other. The isomorphism is given by

$$\begin{array}{ll} MO(\mathbf{A}) \to NF(\mathbf{A}) & \qquad NF(\mathbf{A}) \to MO(\mathbf{A}) \\ \theta \mapsto \nabla_\theta = \{a \in \mathbf{A} : \mathbf{1} \leq_\theta a\} & \qquad \nabla \mapsto \theta_\nabla = \{(a,b) \in \mathbf{A} \times \mathbf{A} : a\backslash b \in \nabla\} \end{array}$$

Proof. By virtue of 2.9, every preorder from $MO(\mathbf{A})$ satisfies A1-A4. Therefore, the set ∇_θ is a normal filter.

We can consider the preorder $\leq_{\mathbf{A}}$ as being given by some set S of sequents added to the free system (see 4.1). Let us construct the set S^ϕ as in 4.3. Let ∇ be a normal filter. Applying (several times) the corollary after the deduction theorem, we get:

> B is derivable from $S^\phi \cup \Delta$ iff there exist formulas $H_1, \ldots, H_k$ derivable by (R*2) and (R3) from the formulas in ∇, such that $H_1 \cdot \ldots \cdot H_k \supset B$ is derivable from S^ϕ.

By theorem 5 from 4.3, $H_1 \cdot \ldots \cdot H_k \leq_{\mathbf{A}} B$. Since ∇ is a normal filter, $H_1 \cdot \ldots \cdot H_k \in \nabla$, and therefore, $B \in \nabla$. Thus, $(a, b) \in \theta_\nabla$ iff the sequent $a \longrightarrow b$ is derivable from the set $S \cup \{\mathbf{1} \longrightarrow x : x \in \nabla\}$. By virtue of 4.1, θ_∇ satisfies A1-A4, i.e. $\theta_\nabla \in MO(\mathbf{A})$.

Let us check that $\theta \mapsto \nabla_\theta$ and $\nabla \mapsto \theta_\nabla$ are mutually inverse. Let $\theta_0 \in MO(\mathbf{A})$ and $\theta_1 = \theta_{\nabla_{\theta_0}}$. Then

$$a \leq_{\theta_1} b \Leftrightarrow a\backslash b \in \nabla_{\theta_0} \Leftrightarrow \mathbf{1} \leq_{\theta_0} a\backslash b \Leftrightarrow a \leq_{\theta_0} b.$$

Let $\nabla_0 \in NF(\mathbf{A})$ and $\nabla_1 = \nabla_{\theta_{\nabla_0}}$. Then

$$a \in \nabla_1 \Leftrightarrow \mathbf{1} \leq_\theta a \Leftrightarrow \mathbf{1}\backslash a \in \nabla_0 \Leftrightarrow a \in \nabla_0.$$

4.6. Intersection of all normal filters containing a set M is called the *normal filter generated by* M. Notice that the normal filter generated by M comprises all $a \in A$ for which there are some $x_1, \ldots, x_n$ obtained from R*2 and R3 from some elements of M, such that $x_1 \cdot \ldots \cdot x_n \leq_{\mathbf{A}} a$.

4.7. If $\mathbf{A}$ is a (non-ordered) group, i.e. $\leq_{\mathbf{A}}$ is symmetrical and $x \cdot y = x \times y$, the set $H = \{x \in \nabla : x^{-1} \in \nabla\}$, where ∇ is a normal filter, is a normal divisor, and ∇ becomes a positive cone of some order on the factor group $\mathbf{A}/H$. If $\mathbf{A}$ is a periodical group, the definition 4.5 is the definition of a normal divisor. If $\mathbf{A}$ is a Heyting-Brouwer algebra, then 4.5 is the definition of the ⅂Γ filter in the sense of [7].

1.5 Bibliography

[1] G. Birkhoff. *Lattice theory*, volume 25 of *Colloquium Publications.* American Mathematical Society, Providence, Rhode Island, 1967.

[2] C.C. Chang. Algebraic analysis of many-valued logics. *Transactions of the American Mathematical Society*, 88(2):467–490, 1958.

[3] L. Fuchs. *Partially ordered algebraic systems*. Mir, Moscow, 1965. (In Russian).

[4] V.N. Grishin. A non-standard logic and its application to set theory. In *Studies in Formalized Languages and Non-classical Logics*, page 135171. Nauka, Moscow, 1974. (in Russian).

[5] V.N. Grishin. On the algebraic semantics of a logic without contraction rules. In *Studies in set theory and non-classical logics*, pages 247–264. Nauka, Moscow, 1976. (in Russian).

[6] Joachim Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.

[7] C. Rauszer. Semi-boolean algebras and their application to intuitionistic logic with dual connectives. *Fundamenta Mathematicae*, 83:219–249, 1974.

[8] K.L.N. Swamy. Dually residuated lattice ordered semigroups. *Mathematische Annalen*, 159(2):105–114, 1965.

2

Symmetries in natural language syntax and semantics: the Lambek-Grishin calculus

Michael Moortgat

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 [13]. 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.

Thanks 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.

2.1 Background

The basic Lambek calculus [18] 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 (2.1).

$$\text{RESIDUATED TRIPLE} \qquad A \rightarrow C/B \quad \text{iff} \quad A \otimes B \rightarrow C \quad \text{iff} \quad B \rightarrow A\backslash C \tag{2.1}$$

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 this interpretation are given in (2.2); one finds the basic soundness/completeness results in [9]. 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 /, \ here are seen as directional implications; elimination of these operations corresponds to function application, introduction to lambda abstraction.

$$\begin{array}{l} x \Vdash A \otimes B \text{ iff } \exists yz.R_{\otimes}xyz \text{ and } y \Vdash A \text{ and } z \Vdash B \\ y \Vdash C/B \text{ iff } \forall xz.(R_{\otimes}xyz \text{ and } z \Vdash B) \text{ implies } x \Vdash C \\ z \Vdash A\backslash C \text{ iff } \forall xy.(R_{\otimes}xyz \text{ and } y \Vdash A) \text{ implies } x \Vdash C \end{array} \tag{2.2}$$

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 (2.3) below; type transitions of this kind have played an important role in our understanding of natural language semantics.

$$A \rightarrow B/(A\backslash B) \qquad (B/(A\backslash B))\backslash B \rightarrow A\backslash B \tag{2.3}$$

It is ironic that precisely in the hypothetical reasoning component the Lambek grammars turn out to be deficient. As one sees in (2.3), hypothetical reasoning typically involves higher order types, where a slash occurs in a negative environment as in the schema (2.4) 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 \.

$$C/(A\backslash B) \qquad (M\ \lambda x^A.N^B)^C \tag{2.4}$$

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 \, 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.

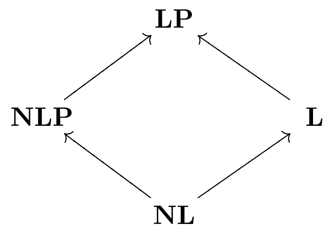


Figure 2.1: The Lambek hierarchy

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 2.1. 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.

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 [20]). Like their binary relatives, $\Diamond, \Box$ form a residuated pair, satisfying (2.5), and with interpretation (2.6).

$$\textsc{residuated pair} \qquad \Diamond A \rightarrow B \quad \text{iff} \quad A \rightarrow \Box B \tag{2.5}$$

$$\begin{array}{l} x \Vdash \Diamond A \text{ iff } \exists y\ (R_\Diamond xy \text{ and } y \Vdash A) \\ y \Vdash \Box A \text{ iff } \forall x\ (R_\Diamond xy \text{ implies } x \Vdash A) \end{array} \tag{2.6}$$

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 $\Box\Diamond A$.

$$\Diamond\Box A \rightarrow A \rightarrow \Box\Diamond A \tag{2.7}$$

Bernardi [2] makes good use of the patterns in (2.7) 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 (2.8) make left branches ($P1$, $P2$) or right branches ($P3$, $P4$) accessible for hypothetical reasoning.

$$\begin{array}{llll} (P1) & \Diamond A \otimes (B \otimes C) \rightarrow (\Diamond A \otimes B) \otimes C & (C \otimes B) \otimes \Diamond A \rightarrow C \otimes (B \otimes \Diamond A) & (P3) \\ (P2) & \Diamond A \otimes (B \otimes C) \rightarrow B \otimes (\Diamond A \otimes C) & (C \otimes B) \otimes \Diamond A \rightarrow (C \otimes \Diamond A) \otimes B & (P4) \end{array} \tag{2.8}$$

Vermaat [27] 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 2.1.1. 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 2.1.1 *Structural control (Kurtonina and Moortgat [15]). For logics $\mathcal{L}$, $\mathcal{L}^\uparrow$, $\mathcal{L}^\downarrow$ and translations $\cdot^\downarrow$ and $\cdot^\uparrow$ as defined above,*

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

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 [21] for an overview of this line of work.

2.2 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 [8], already the simplest extension with an associative regime is known to be NP complete [26]; 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 [24].

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 [13] 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]

[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'.

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

Figure 2.2: Residuated and Galois connected pairs and their duals.

$$\begin{array}{rcll} A, B & ::= & p \mid & \text{atoms: } s \text{ sentence, } np \text{ noun phrases, } \ldots \\ & & A \otimes B \mid B\backslash A \mid A/B \mid & \text{product, left vs right division} \\ & & A \oplus B \mid A \oslash B \mid B \oslash A & \text{coproduct, right vs left difference} \end{array} \tag{2.9}$$

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

$$\begin{array}{rccccc} \text{RESIDUATED TRIPLE} & A \rightarrow C/B & \text{iff} & A \otimes B \rightarrow C & \text{iff} & B \rightarrow A\backslash C \\ \text{DUAL RESIDUATED TRIPLE} & B \oslash C \rightarrow A & \text{iff} & C \rightarrow B \oplus A & \text{iff} & C \oslash A \rightarrow B \end{array} \tag{2.10}$$

Dunn's [10] framework of gaggle theory brings out the underlying algebraic structure in a particularly clear way. In Fig 2.2, 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 $* \in \{/, \otimes, \backslash, \oslash, \oplus, \oslash\}$, we write $-*$ $(*-)$ for the operation that suffixes (prefixes) a fixed type to its operand, for example: $A \rightarrow C/B$ iff $B \rightarrow A\backslash C$ instantiates the pattern $(/-, -\backslash)$.

The patterns of Fig 2.2 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 (2.11): $\bowtie$ is order-preserving, ∞ order-reversing: $A^{\bowtie} \rightarrow B^{\bowtie}$ iff $A \rightarrow B$ iff $B^{\infty} \rightarrow A^{\infty}$.

$$\bowtie \quad \frac{C/D \quad A \otimes B \quad B \oplus A \quad D \oslash C}{\overline{D\backslash C \quad B \otimes A \quad A \oplus B \quad C \oslash D}} \qquad \infty \quad \frac{C/B \quad A \otimes B \quad A\backslash C}{\overline{B \oslash C \quad B \oplus A \quad C \oslash A}} \tag{2.11}$$

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 (2.10). $\mathbf{LG}_{\emptyset}$ by itself does not offer us the kind of expressivity needed to address the problems discussed in §2.1. 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 (2.12) which we will collectively refer to as $\mathcal{G}^{\uparrow}$.[2]

[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 [13] such as [19, 12] omit $G2$ and $G4$. The full set of (2.12) is essential for the intended linguistic applications.

$$\begin{array}{llll} (G1) & (A \oslash B) \otimes C \to A \oslash (B \otimes C) & C \otimes (B \oslash A) \to (C \otimes B) \oslash A & (G3) \\ (G2) & C \otimes (A \oslash B) \to A \oslash (C \otimes B) & (B \oslash A) \otimes C \to (B \otimes C) \oslash A & (G4) \end{array} \tag{2.12}$$

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 (2.13).

$$\frac{A \oslash (B \otimes C) \to D}{(A \oslash B) \otimes C \to D}\, G1 \qquad \frac{(A \otimes B) \oslash C \to D}{A \otimes (B \oslash C) \to D}\, G3$$
$$\frac{B \oslash (A \otimes C) \to D}{A \otimes (B \oslash C) \to D}\, G2 \qquad \frac{(A \otimes C) \oslash B \to D}{(A \oslash B) \otimes C \to D}\, G4 \tag{2.13}$$

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 (2.12) under $\cdot^{\infty}$ are given in (2.14): 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 (2.14) are derivable from (2.12) — the derivation of $G1'$ from $G1$ is given as an example in (2.19).

$$\begin{array}{llll} (G1') & (C \oplus B)/A \to C \oplus (B/A) & A\backslash(B \oplus C) \to (A\backslash B) \oplus C & (G3') \\ (G2') & (B \oplus C)/A \to (B/A) \oplus C & A\backslash(C \oplus B) \to C \oplus (A\backslash B) & (G4') \end{array} \tag{2.14}$$

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 §2.3. 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 $\mathbf{LG}$.

2.2.1 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 (2.10), the Grishin axioms in rule form (2.16), and the monotonicity rules of (2.17).[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 (2.15).

[3]The axiomatization presented here is a close relative of Display Logic, see [12] 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.

$$\frac{f : A \otimes B \to C}{\triangleleft f : B \to A \backslash C} \quad \frac{f : C \to A \oplus B}{\blacktriangleleft f : A \obslash C \to B}$$
$$\frac{f : A \otimes B \to C}{\triangleright f : A \to C / B} \quad \frac{f : C \to A \oplus B}{\blacktriangleright f : C \oslash B \to A} \tag{2.15}$$

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.

$$\frac{f : A \obslash (B \otimes C) \to D}{\urcorner f : (A \obslash B) \otimes C \to D} \quad \frac{f : (A \otimes B) \oslash C \to D}{\ulcorner f : A \otimes (B \oslash C) \to D}$$
$$\frac{f : B \obslash (A \otimes C) \to D}{\lrcorner f : A \otimes (B \obslash C) \to D} \quad \frac{f : (A \otimes C) \oslash B \to D}{\llcorner f : (A \oslash B) \otimes C \to D} \tag{2.16}$$

Finally, (2.17) 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 combination monotonicity plus residuation effectively *absorbs* cut. Admissibility of cut is established in Appendix 2.5, extending the earlier result of [22] to the case of symmetric **LG**.

$$\frac{f : A \to B \quad g : C \to D}{f \otimes g : A \otimes C \to B \otimes D} \quad \frac{f : A \to B \quad g : C \to D}{f \oplus g : A \oplus C \to B \oplus D}$$
$$\frac{f : A \to B \quad g : C \to D}{f / g : A / D \to B / C} \quad \frac{f : A \to B \quad g : C \to D}{f \oslash g : A \oslash D \to B \oslash C} \tag{2.17}$$
$$\frac{f : A \to B \quad g : C \to D}{g \backslash f : D \backslash A \to C \backslash B} \quad \frac{f : A \to B \quad g : C \to D}{g \obslash f : D \obslash A \to C \obslash B}$$

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

$$\infty \quad \frac{h/g \quad f \otimes g \quad f \backslash h}{g \obslash h \quad g \oplus f \quad h \oslash f} \qquad \infty \quad \frac{\triangleleft f \quad \triangleright f \quad \triangleleft' f \quad \triangleright' f}{\blacktriangleright f \quad \blacktriangleleft f \quad \blacktriangleright' f \quad \blacktriangleleft' f} \tag{2.18}$$

As an example, consider (2.19). 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 $\bowtie \infty$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(c \oplus b)/a \to (c \oplus b)/a}{((c \oplus b)/a) \otimes a \to c \oplus b}\triangleright'}{c \obslash (((c \oplus b)/a) \otimes a) \to b}\blacktriangleleft}{(c \obslash ((c \oplus b)/a)) \otimes a \to b}\urcorner}{c \obslash ((c \oplus b)/a) \to b/a}\triangleright}{(c \oplus b)/a \to c \oplus (b/a)}\blacktriangleleft' \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{a \obslash (b \otimes c) \to a \obslash (b \otimes c)}{b \otimes c \to a \oplus ((a \obslash (b \otimes c))}\blacktriangleleft'}{b \to (a \oplus ((a \obslash (b \otimes c)))/c}\triangleright}{b \to a \oplus ((a \obslash (b \otimes c))/c)}\urcorner^{\infty}}{a \obslash b \to (a \obslash (b \otimes c))/c}\blacktriangleleft}{(a \obslash b) \otimes c \to a \obslash (b \otimes c)}\triangleright' \tag{2.19}$$

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

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 [7]. Capelletti [4] 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 [4] can be extended to include the Grishin interaction principles remains to be investigated.

2.2.2 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 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
$\mathrm{Frm}(\oslash, \oplus, \oslash)$	1	$\bowtie$

(2.20)

One easily checks that the Grishin postulates (2.12) 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 (2.21) 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

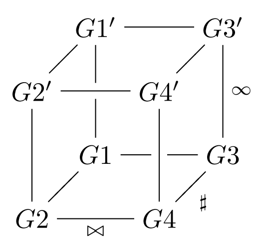
(2.21)

Similarly, from ∞ we obtain three further order-reversing symmetries: $\natural = \bowtie \infty$, and the mixed forms $\dagger$ and $\ddagger$ in (2.22) below.

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

(2.22)

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.



(2.23)

The lower plane is the square (2.12) which we saw is characterized by D_2. The transformation ∞ reflects (2.12) to the upper plane (2.14). 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

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

CALCULUS	INTERPRETATION
NL	free quasigroup (Foret [11])
L	free group (Pentus [25])
LP	free Abelian group (Pentus [25])
LG	free Abelian group (Moortgat and Pentus [23])

Table 2.1: Models for type equivalence

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 $G2'$.

2.2.3 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 [17] as the reflexive, transitive, symmetric closure of the derivability relation: $A \sim B$ iff there exists a sequence $C_1 \dots C_n$ $(1 \leq n)$ 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 [17] are given in (2.24). It is shown in [11] that these solutions are in fact adequate for the pure logic of residuation, i.e. the non-associative calculus **NL**.

$$\mathbf{NL}: \quad D = (A/((C/C)\backslash C)) \otimes ((C/C)\backslash B), \quad C = (A \otimes (D\backslash D))/(B\backslash (D \otimes (D\backslash D))) \tag{2.24}$$

For associative **L**, [25] has the shorter solution in (2.25). 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 (2.26).

$$\mathbf{L}: \quad D = (A/C) \otimes (C \otimes (C\backslash B)), \quad C = (D/A)\backslash (D/(B\backslash D)) \tag{2.25}$$

$$\mathbf{LG}: \quad D = (A/C) \otimes (C \oslash (B \obslash C)), \quad C = (A \oslash D) \oplus (D/(B\backslash D)) \tag{2.26}$$

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 2.1 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 (2.27).

$$[\![p]\!] = p,\ [\![A \otimes B]\!] = [\![A]\!] \cdot [\![B]\!],\ [\![A/B]\!] = [\![A]\!] \cdot [\![B]\!]^{-1},\ [\![B\backslash A]\!] = [\![B]\!]^{-1} \cdot [\![A]\!] \tag{2.27}$$

We see in Table 2.1 that for the systems in the Lambek hierarchy, expressivity for $\sim$ is inversely proportional to structural discrimination: the structural rules of associativity and commutativity destroy sensitivity for constituent structure and word order. The result below shows that **LG**

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

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 2.2.1 *(Moortgat and Pentus [23]) 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 [23] for **LG** is a variant of that in (2.27); for **LG** one adds an extra element $\star$ to keep track of the *operator count*, defined as in (2.28). 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.

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

$$\begin{array}{rclrcl}
|A \otimes B|_\otimes & = & |A|_\otimes + |B|_\otimes + 1 & |A \otimes B|_\oplus & = & |A|_\oplus + |B|_\oplus \\
|A \oplus B|_\otimes & = & |A|_\otimes + |B|_\otimes & |A \oplus B|_\oplus & = & |A|_\oplus + |B|_\oplus + 1 \\
|A \,/\, B|_\otimes & = & |A|_\otimes - |B|_\otimes - 1 & |A \,/\, B|_\oplus & = & |A|_\oplus - |B|_\oplus \\
|B \,\backslash\, A|_\otimes & = & |A|_\otimes - |B|_\otimes - 1 & |B \,\backslash\, A|_\oplus & = & |A|_\oplus - |B|_\oplus \\
|A \oslash B|_\otimes & = & |A|_\otimes - |B|_\otimes & |A \oslash B|_\oplus & = & |A|_\oplus - |B|_\oplus - 1 \\
|B \obslash A|_\otimes & = & |A|_\otimes - |B|_\otimes & |B \obslash A|_\oplus & = & |A|_\oplus - |B|_\oplus - 1
\end{array} \tag{2.28}$$

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 §2.3. Below we reproduce a step in the proof of Thm 2.2.1 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 $X \rightarrow B\backslash A$ and $X \rightarrow A/B$, which by residuation means

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

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

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

which by Grishin mixed associativity or commutativity follows from

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

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.

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

2.2.4 Relational semantics

Let us turn now to the frame semantics for **LG**. We have seen in (2.2) 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.

$$\begin{array}{l} x \Vdash A \oplus B \text{ iff } \forall yz.R_\oplus xyz \text{ implies } (y \Vdash A \text{ or } z \Vdash B) \\ y \Vdash C \oslash B \text{ iff } \exists xz.R_\oplus xyz \text{ and } z \nVdash B \text{ and } x \Vdash C \\ z \Vdash A \obslash C \text{ iff } \exists xy.R_\oplus xyz \text{ and } y \nVdash A \text{ and } x \Vdash C \end{array} \tag{2.29}$$

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 [14] 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 $\cdot \widehat{\otimes} \cdot$, $\cdot \widehat{\obslash} \cdot$ defined in (2.30).

$$\begin{array}{rcl} X \widehat{\otimes} Y & = & \{C \mid \exists A, B \ (A \in X \text{ and } B \in Y \text{ and } A \otimes B \to C)\} \\ X \widehat{\obslash} Y & = & \{B \mid \exists A, C \ (A \notin X \text{ and } C \in Y \text{ and } A \obslash C \to B\}, \text{ alternatively} \\ & = & \{B \mid \exists A, C \ (A \notin X \text{ and } C \in Y \text{ and } C \to A \oplus B\} \end{array} \tag{2.30}$$

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

$$(\dagger) \quad \lfloor A \otimes B \rfloor = \lfloor A \rfloor \widehat{\otimes} \lfloor B \rfloor \qquad (\ddagger) \quad \lfloor A \obslash C \rfloor = \lceil A \rceil^\sim \widehat{\obslash} \lfloor C \rfloor \tag{2.31}$$

The equations of (2.31) 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.

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 $Y \widehat{\otimes} Z \subseteq X$

$R^c_\oplus XYZ$ iff $Y \widehat{\obslash} X \subseteq Z$

$V^c(p) = \{X \in W^c \mid p \in X\}$

[6] Allwein and Dunn [1] develop a richer theory of Kripke models for a hierarchy of substructural logics, accommodating both the lattice operations and (co)product and (co)implications.

Theorem 2.2.2 *Soundness and completeness (Kurtonina and Moortgat [16]).*

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

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 (2.32) (where $R^{(-2)}xyz = Rzyx$).

$$\forall xyzwv \ \ (R_{\otimes}xyz \ \wedge \ R_{\oplus}^{(-2)}ywv) \Rightarrow \exists t \ (R_{\oplus}^{(-2)}xwt \ \wedge \ R_{\otimes}tvz) \tag{2.32}$$

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 §2.3.

2.2.5 Computational semantics

The second type of semantics we want to consider is the Curry-Howard interpretation of **LG** derivations. In Bernardi and Moortgat [3], one finds a continuation semantics based on (a directional refinement of) Curien and Herbelin's [5] 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 [3] 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 §2.2.1 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 (2.33), 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).

$$\begin{array}{rclcrcl} \lceil A\backslash B\rceil & = & R^{\lceil A\rceil\times R^{\lceil B\rceil}} & & \lceil B\oslash A\rceil & = & \lceil B\rceil\times R^{\lceil A\rceil}; \\ \lceil B/A\rceil & = & R^{R^{\lceil B\rceil}\times\lceil A\rceil} & & \lceil A\obslash B\rceil & = & R^{\lceil A\rceil}\times\lceil B\rceil. \end{array} \tag{2.33}$$

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 $f : A \rightarrow B$. Assuming countably infinite sets of variables x_i and covariables α_i, we inductively associate each arrow $f : A \rightarrow B$ 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 (2.34).

$$\lceil f^{\triangleright}\rceil : \lceil A\rceil \mapsto R^{R^{\lceil B\rceil}} \qquad \text{and} \qquad \lceil f^{\triangleleft}\rceil : R^{\lceil B\rceil} \mapsto R^{\lceil A\rceil} \tag{2.34}$$

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 (2.35).[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.

$$\lceil (1_A)^{\triangleright}\rceil\ x = \lambda k.(k\ x) \qquad \lceil (1_A)^{\triangleleft}\rceil\ \alpha = \lambda x.(\alpha\ x) \tag{2.35}$$

Monotonicity Given $f : A \to B$ and $g : C \to D$ we have the monotonicity rules of (2.36). We use curly brackets to distinguish the meta-application of the function definition from the actual target language lambda term computed.

$$\begin{array}{rcl}
\lceil (g\backslash f)^{\triangleright}\rceil\ y &=& \lambda k.(k\ \lambda\langle x,\beta\rangle.(\{\lceil g^{\triangleright}\rceil\ x\}\ \lambda m.(y\ \langle m, \{\lceil f^{\triangleleft}\rceil\ \beta\}\rangle)))\\
\lceil (f/g)^{\triangleright}\rceil\ x &=& \lambda k.(k\ \lambda\langle \alpha,y\rangle.(\{\lceil g^{\triangleright}\rceil\ y\}\ \lambda m.(x\ \langle \{\lceil f^{\triangleleft}\rceil\ \alpha\}, m\rangle)))\\
\lceil (f \oslash g)^{\triangleleft}\rceil\ \alpha &=& \lambda\langle x,\delta\rangle.(\{\lceil f^{\triangleright}\rceil\ x\}\ \lambda y.(\alpha\ \langle y, \{\lceil g^{\triangleleft}\rceil\ \delta\}\rangle))\\
\lceil (g \obslash f)^{\triangleleft}\rceil\ \beta &=& \lambda\langle \delta,x\rangle.(\{\lceil f^{\triangleright}\rceil\ x\}\ \lambda y.(\beta\ \langle \{\lceil g^{\triangleleft}\rceil\ \delta\}, y\rangle))
\end{array} \tag{2.36}$$

Residuation For the typing of the arrows f, see (2.15).

$$\begin{array}{lll|lll}
\lceil (\triangleleft f)^{\triangleright}\rceil & y & = \lambda k.(k\ \lambda\langle x,\gamma\rangle.(\{\lceil f^{\triangleright}\rceil\ \langle x,y\rangle\}\ \gamma)) & \lceil (\blacktriangleright f)^{\triangleleft}\rceil & \alpha & = \lambda\langle z,\beta\rangle.(\{f^{\triangleleft}\ \langle \alpha,\beta\rangle\}\ z)\\
\lceil (\triangleright f)^{\triangleright}\rceil & x & = \lambda k.(k\ \lambda\langle \gamma,y\rangle.(\{\lceil f^{\triangleright}\rceil\ \langle x,y\rangle\}\ \gamma)) & \lceil (\blacktriangleleft f)^{\triangleleft}\rceil & \beta & = \lambda\langle \alpha,z\rangle.(\{f^{\triangleleft}\ \langle \alpha,\beta\rangle\}\ z)\\
\lceil (\triangleleft' f)^{\triangleright}\rceil & \langle x,y\rangle & = \lambda\gamma.(\{\lceil f^{\triangleright}\rceil\ y\}\ \lambda n.(n\ \langle x,\gamma\rangle)) & \lceil (\blacktriangleright' f)^{\triangleleft}\rceil & \langle \alpha,\beta\rangle & = \lambda z.(\{\lceil f^{\triangleleft}\rceil\ \alpha\}\ \langle z,\beta\rangle)\\
\lceil (\triangleright' f)^{\triangleright}\rceil & \langle x,y\rangle & = \lambda\gamma.(\{\lceil f^{\triangleright}\rceil\ x\}\ \lambda m.(m\ \langle \gamma,y\rangle)) & \lceil (\blacktriangleleft' f)^{\triangleleft}\rceil & \langle \alpha,\beta\rangle & = \lambda z.(\{\lceil f^{\triangleleft}\rceil\ \beta\}\ \langle \alpha,z\rangle)
\end{array} \tag{2.37}$$

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.

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

$$\begin{array}{rcl}
\lceil (\urcorner f)^{\triangleright}\rceil\quad \langle\langle w,v\rangle, z\rangle &=& \lambda\delta.((\lceil f^{\triangleright}\rceil\ \langle w, \langle v,z\rangle\rangle)\ \delta)\\
\lfloor (\urcorner f)^{\triangleright}\rfloor = \lceil ((\urcorner f)^{\infty})^{\triangleleft}\rceil\quad \langle \gamma,\delta\rangle &=& \lambda z.(\delta\ \lambda\langle \beta,x\rangle.((\lceil (f^{\infty})^{\triangleleft}\rceil\ \lambda m.(m\ \langle\langle \gamma,\beta\rangle, x\rangle)), z))
\end{array} \tag{2.38}$$

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.

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

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

$$
\begin{array}{ll}
(\text{cbv}) & \lambda c.((\lceil\!\lceil \mathbf{left} \rceil\!\rceil \; \langle \pi^2 \lceil\!\lceil \mathbf{somebody} \rceil\!\rceil, \lambda z.(\pi^1 \lceil\!\lceil \mathbf{somebody} \rceil\!\rceil \; \langle z, c \rangle)\rangle)) \\
(\text{cbn}) & \lambda c.((\lfloor\!\lfloor \mathbf{somebody} \rfloor\!\rfloor \; \lambda \langle q, y \rangle.(y \; \langle c, \lambda c'.(\lfloor\!\lfloor \mathbf{left} \rfloor\!\rfloor \; \langle c', q \rangle)\rangle)))
\end{array}
$$

2.3 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 §2.1: non-local semantic construal and displacement. These problems find a natural solution in the symmetric Lambek-Grishin systems. As shown in (2.39), in both cases the solution starts from a lexical type assignment from which the usual Lambek type is derivable.

$$
\begin{array}{lrcl}
\text{someone} & (s \oslash s) \obslash np & \to & s/(np \backslash s) \\
\text{which} & (n \backslash n)/((s \oslash s) \oplus (s/np)) & \to & (n \backslash n)/(s/np)
\end{array}
\tag{2.39}
$$

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 (2.40) one finds a summary derivation[8] for the reading where 'someone' has wide scope. 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 (2.41). The reader is invited to consult [3] 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.

$$
\dfrac{\dfrac{\dfrac{np \otimes (((np \backslash s)/s) \otimes (np \otimes (np \backslash s))) \to s \quad s \to (s \oslash s) \oplus s}{np \otimes (((np \backslash s)/s) \otimes (np \otimes (np \backslash s))) \to ((s \oslash s) \oplus s)}\;\text{trans}}{((s \oslash s) \obslash (np \otimes (((np \backslash s)/s) \otimes (np \otimes (np \backslash s))))) \to s}\;\text{drp}}{np \otimes (((np \backslash s)/s) \otimes (\underbrace{((s \oslash s) \obslash np)}_{\textsf{someone}} \otimes (np \backslash s))) \to s}\;G2
\tag{2.40}
$$

$$
\lambda c.(\lfloor\!\lfloor \mathsf{someone} \rfloor\!\rfloor \; \lambda \langle q, y \rangle.(y \; \langle c, \lambda c'.(\lfloor\!\lfloor \mathsf{suspect} \rfloor\!\rfloor \; \langle \lambda c''.(\lfloor\!\lfloor \mathsf{cheating} \rfloor\!\rfloor \; \langle c'', q \rangle), \langle c', \lfloor\!\lfloor \mathsf{alice} \rfloor\!\rfloor \rangle\rangle)\rangle)))
\tag{2.41}
$$

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

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 (2.42) combines the Grishin principles of (2.12) 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.

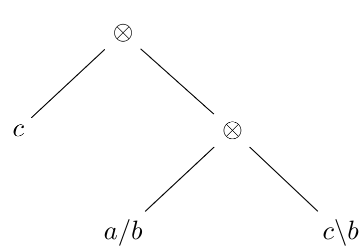
$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{np \otimes ((tv \otimes np) \otimes adv) \to s \quad s \to (s \oslash s) \oplus s}{np \otimes ((tv \otimes np) \otimes adv) \to (s \oslash s) \oplus s}\ \text{trans}}{tv \to ((np\backslash(s \oslash s))/adv) \oplus (s/np)}\ Gn,\text{rp}}{np \otimes ((tv \oslash (s/np)) \otimes adv) \to s \oslash s}\ \text{rp,drp}}{(np \otimes (tv \otimes adv)) \oslash (s/np) \to s \oslash s}\ Gn^{-1}}{np \otimes (tv \otimes adv) \to (s \oslash s) \oplus (s/np)}\ \text{drp} \qquad (2.42)
$$

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.

$$
\dfrac{\Gamma[\Delta \circ B] \Rightarrow C}{\Gamma[\Delta] \Rightarrow (C \oslash C) \oplus (C/B)}\ \dagger \qquad \dfrac{\Gamma[\Delta \circ B] \Rightarrow C}{\Gamma[\Delta] \Rightarrow C/\Diamond\Box B}\ \ddagger \qquad (2.43)
$$

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 (2.44) 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.



(2.44)

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:

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

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

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:

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

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

$$(a/\Diamond\Box b)\backslash(c\backslash a) \to ((a/b)\backslash a) \oplus (c\backslash(a \oslash a))$$

2.4 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 [6] position in his contribution to the 1960 New York conference where also [18] 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.

2.5 Appendix: Admissibility of cut

A cut-elimination algorithm for the **Frm**$(/, \otimes, \backslash)$ fragment (i.e. **NL**) is presented in [22]. 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 §2.2.1, 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

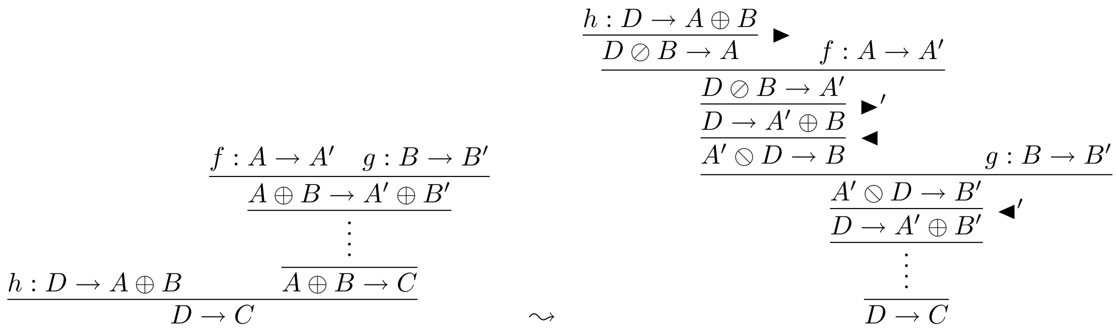


Figure 2.3: $k[f \oplus g] \circ h = k[\blacktriangleleft'(g \circ \blacktriangleleft \blacktriangleright' (f \circ \blacktriangleright h))]$

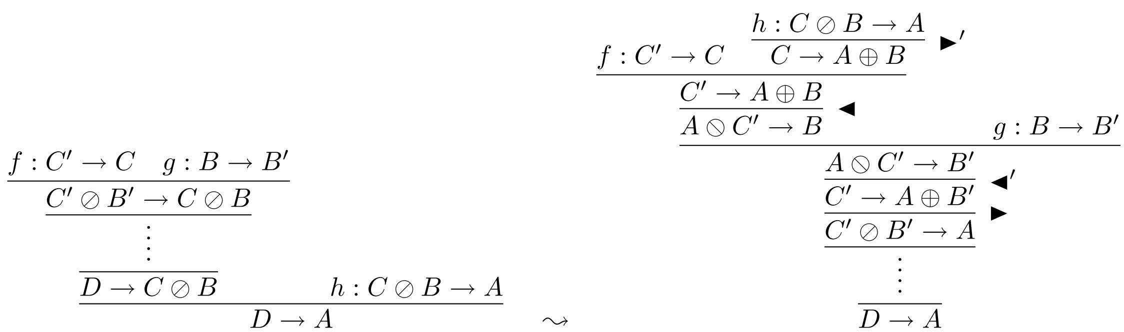


Figure 2.4: $h \circ k[f \oslash g] = k[\blacktriangleright \blacktriangleleft' (g \circ \blacktriangleleft(\blacktriangleright' h \circ f))]$

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 [22] to $\mathbf{Frm}(\oslash, \oplus, \obslash)$ is immediate via arrow reversal. In Fig 2.3 and Fig 2.4 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 (2.12)), 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 2.5 is an instance of cut immediately preceded by $\urcorner$. We have unfolded the left 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 2.6 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 $\urcorner$.

2.6 Bibliography

[1] Gerard Allwein and J. Michael Dunn. Kripke models for Linear Logic. *Journal of Symbolic Logic*, 58(2):514–545, 1993.

[2] Raffaella Bernardi. *Reasoning with polarities in categorial type logic*. PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University, 2002.

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\begin{array}{c}\dfrac{g : A \to A' \quad f : B' \to B}{A' \oslash B' \to A \oslash B}\\ \vdots\end{array}}{k_1[g \oslash f] : E' \to A \oslash B} \quad h : C' \to C}{E' \otimes C' \to (A \oslash B) \otimes C}\\ \vdots\end{array}}{k_1'[k_1[g \oslash f] \otimes h] : E \to (A \oslash B) \otimes C} \quad \dfrac{k_2 : A \oslash (B \otimes C) \to D}{\urcorner\, k_2 : (A \oslash B) \otimes C \to D}}{\urcorner\, k_2 \circ k_1'[k_1[g \oslash f] \otimes h] : E \to D}$$

Figure 2.5: An instance of $G1$ feeding cut.

$$\begin{array}{c}\dfrac{\begin{array}{c}\dfrac{\dfrac{\dfrac{\dfrac{h : C' \to C \quad \dfrac{\dfrac{\dfrac{f : B' \to B \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{k_2 : A \oslash (B \otimes C) \to D}{B \otimes C \to A \oplus D}\blacktriangleleft'}{(B \otimes C) \oslash D \to A}\blacktriangleright \quad g : A \to A'}{(B \otimes C) \oslash D \to A'}}{B \otimes C \to A' \oplus D}\blacktriangleright'}{B \to (A' \oplus D)/C}\triangleright}{B' \to (A' \oplus D)/C}}{B' \otimes C \to A' \oplus D}\triangleright'}{C \to B' \backslash (A' \oplus D)}\triangleleft}{C' \to B' \backslash (A' \oplus D)}}{B' \otimes C' \to A' \oplus D}\triangleleft'}{A' \oslash (B' \otimes C') \to D}\blacktriangleleft}{(A' \oslash B') \otimes C' \to D}\urcorner}{A' \oslash B' \to D/C'}\triangleright \\ \vdots \end{array}}{\dfrac{E' \to D/C'}{E' \otimes C' \to D}\triangleright'} \\ \vdots \\ \hline E \to D\end{array}$$

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

Figure 2.6: Permutability of Grishin and cut

[3] Raffaella Bernardi and Michael Moortgat. Continuation semantics for symmetric categorial grammar. In Daniel Leivant and Ruy de Queiros, editors, *Proceedings 14th Workshop on Logic, Language, Information and Computation (WoLLIC'07)*, LNCS 4576. Springer, 2007.

[4] Matteo Capelletti. *Parsing with structure-preserving categorial grammars*. PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University, 2007.

[5] P. Curien and H. Herbelin. Duality of computation. In *International Conference on Functional Programming (ICFP'00)*, pages 233–243, 2000. [2005: corrected version].

[6] Haskell B. Curry. Some logical aspects of grammatical structure. In Roman Jacobson, editor, *Structure of Language and its Mathematical Aspects*, volume XII of *Proceedings of the Symposia in Applied Mathematics*, pages 56–68. American Mathematical Society, 1961.

[7] P. De Groote and F. Lamarche. Classical non-associative Lambek calculus. *Studia Logica*, 71:335–388, 2002.

[8] Philippe de Groote. The non-associative Lambek calculus with product in polynomial time. In N. V. Murray, editor, *Automated Reasoning With Analytic Tableaux and Related Methods*, volume 1617 of *Lecture Notes in Artificial Intelligence*, pages 128–139. Springer, 1999.

[9] Kosta Došen. A brief survey of frames for the Lambek calculus. *Zeitschrift für mathematischen Logik und Grundlagen der Mathematik*, 38:179–187, 1992.

[10] J. Michael Dunn. Gaggle theory: An abstraction of galois connections and residuation with applications to negation, implication, and various logical operations. In Jan van Eijck, editor, *Logics in AI, European Workshop, JELIA '90, Amsterdam, The Netherlands, September 10-14, 1990, Proceedings*, volume 478 of *Lecture Notes in Computer Science*, pages 31–51. Springer, 1990.

[11] Annie Foret. On the computation of joins for non-associative Lambek categorial grammars. In Jordi Levy, Michael Kohlhase, Joachim Niehren, and Mateu Villaret, editors, *Proceedings of the 17th International Workshop on Unification, UNIF'03 Valencia*, pages 25–38, 2003.

[12] Rajeev Goré. Substructural logics on display. *Logic Journal of IGPL*, 6(3):451–504, 1997.

[13] V.N. Grishin. On a generalization of the Ajdukiewicz-Lambek system. In A.I. Mikhailov, editor, *Studies in Nonclassical Logics and Formal Systems*, pages 315–334. Nauka, Moscow, 1983. [English translation in Abrusci and Casadio (eds.) Proceedings 5th Roma Workshop, Bulzoni Editore, Roma, 2002].

[14] Natasha Kurtonina. *Frames and Labels. A Modal Analysis of Categorial Inference*. PhD thesis, OTS Utrecht, ILLC Amsterdam, 1995.

[15] Natasha Kurtonina and Michael Moortgat. Structural control. In Patrick Blackburn and Maarten de Rijke, editors, *Specifying Syntactic Structures*, pages 75–113. CSLI Publications, Stanford, 1997.

[16] Natasha Kurtonina and Michael Moortgat. Relational semantics for the Lambek-Grishin calculus. Technical report, AI Preprint Series, Utrecht, 2007. (Submitted to MOL'10, UCLA, Los Angeles).

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

[18] J. Lambek. On the calculus of syntactic types. In R. Jakobson, editor, *Structure of Language and Its Mathematical Aspects*, pages 166–178. American Mathematical Society, 1961.

[19] J. Lambek. From categorial to bilinear logic. In K. Došen P. Schröder-Heister, editor, *Substructural Logics*, pages 207–237. Oxford University Press, 1993.

[20] Michael Moortgat. Multimodal linguistic inference. *Journal of Logic, Language and Information*, 5(3–4):349–385, 1996.

[21] Michael Moortgat. Categorial type logics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, chapter 2, pages 93–177. Elsevier/MIT Press, 1997.

[22] Michael Moortgat and Richard Oehrle. Proof nets for the grammatical base logic. In Michele Abrusci and Claudia Casadio, editors, *Dynamic perspectives in logic and linguistics*, pages 131–143, Roma, 1999. Bulzoni.

[23] Michael Moortgat and Mati Pentus. Type similarity for the Lambek-Grishin calculus. In *Proceedings 12th Conference on Formal Grammar*, Dublin, 2007.

[24] Richard Moot. *Proof Nets for Linguistic Analysis*. PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University, 2002.

[25] Mati Pentus. 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.

[26] Mati Pentus. Lambek calculus is NP-complete. *Theoretical Computer Science*, 357:186–201, 2006.

[27] Willemijn Vermaat. *The logic of variation. A cross-linguistic account of wh-question formation.* PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University, 2006.

3

Continuation semantics for symmetric categorial grammar

Raffaella Bernardi and Michael Moortgat

Abstract Categorial grammars in the tradition of Lambek [12, 13] are asymmetric: sequent statements are of the form $\Gamma \Rightarrow A$, where the succedent is a single formula A, the antecedent a structured configuration of formulas $A_1, \ldots, A_n$. The absence of structural context in the succedent makes the analysis of a number of phenomena in natural language semantics problematic. A case in point is scope construal: the different possibilities to build an interpretation for sentences containing generalized quantifiers and related expressions. In this paper, we explore a symmetric version of categorial grammar based on work by Grishin [10]. In addition to the Lambek product, left and right division, we consider a dual family of type-forming operations: coproduct, left and right difference. Communication between the two families is established by means of structure-preserving distributivity principles. We call the resulting system **LG**. We present a Curry-Howard interpretation for $\mathbf{LG}(/, \backslash, \oslash, \oslash)$ derivations. Our starting point is Curien and Herbelin's sequent system for $\lambda\mu$ calculus [5] which capitalizes on the duality between logical implication (i.e. the Lambek divisions under the formulas-as-types perspective) and the difference operation. Importing this system into categorial grammar requires two adaptations: we restrict to the subsystem where linearity conditions are in effect, and we refine the interpretation to take the left-right symmetry and absence of associativity/commutativity into account. We discuss the continuation-passing-style (CPS) translation, comparing the call-by-value and call-by-name evaluation regimes. We show that in the latter (but not in the former) the types of **LG** are associated with appropriate denotational domains to enable a proper treatment of scope construal.

Thanks We thank Chris Barker and Ken Shan for sharing their view on continuation semantics with us at an earlier presentation of Lambek-Grishin calculus at the workshop Proof Theory at the Syntax-Semantics Interface (LSA Institute, Harvard/MIT, July 2005). Special thanks to Peter Selinger, for helpful discussion on the duality between call-by-value and call-by-name during GEOCAL'06 (Marseille-Luminy, February 2006) and to Philippe de Groote for bringing Curien and Herbelin's work to our attention. All errors remain our own.

3.1 Background

Lambek-style categorial grammars offer an attractive computational perspective on the principle of compositionality: under the Curry-Howard interpretation, derivations are associated with instructions for meaning assembly. In natural language semantics, scope construal of generalized quantifier expressions presents an ideal testing ground to bring out the merits of this approach. Scope construal exemplifies a class of phenomena known as *in situ* binding. An *in situ* binder syntactically occupies the position of a phrase of type A; semantically, it binds an A-type variable in that position within a context of type B, producing a value of type C as a result. The inference pattern of (3.1) (from [15]) schematically captures this behaviour in the format of a sequent rule. The challenge is to solve the equation for the type alias $q(A,B,C)$ in terms of the primitive type-forming operations.

$$\frac{\Delta[x : A] \Rightarrow N : B \qquad\qquad \Gamma[y : C] \Rightarrow M : D}{\Gamma[\Delta[z : q(A,B,C)]] \Rightarrow M[y := (z\ \lambda x.N)] : D}\ . \tag{3.1}$$

It is a poignant irony that precisely in the area of scope construal, the performance of the original Lambek calculus (whether in its associative or non-associative incarnation) is disappointing. For a sentence-level generalized quantifier (GQ) phrase, we have $A = np$, $B = C = s$ in (3.1). The type-forming operations available to define $q(np,s,s)$ are the left and right slashes. A first problem is the lack of *type uniformity*. Given standard modeltheoretic assumptions about the interpretation of the type language, an assignment $s/(np\backslash s)$ to a GQ phrase is associated with an appropriate domain of interpretation (a set of sets of individuals), but with such a type a GQ is syntactically restricted to subject positions: for phrase-internal GQ occurrences, context-dependent extra lexical type assignments have to be postulated. Second, this lexical ambiguity strategy breaks down as soon as one considers *non-local* scope construal, where the distance between the GQ occurrence and the sentential domain where it establishes its scope can be unbounded.

The solutions that have been proposed in the type-logical literature we consider suboptimal. The type-shifting approach of Hendriks [11] and the multimodal accounts based on wrapping operations of Morrill and co-workers [17, 18] each break the isomorphic relation between derivations and terms that is at the heart of the Curry-Howard interpretation. Hendriks introduces a one-to-many dichotomy between syntactic and semantic derivations. Morrill makes the opposite choice: a multiplicity of syntactically distinct implicational operations which collapse at the semantic level.

The approach we develop in the sections below sticks to the *minimal* categorial logic: the pure logic of residuation. We overcome the expressive limitations of the Lambek calculi by lifting the single succedent formula restriction and move to a symmetric system where the Lambek connectives (product, left and right division) coexist with a dual family (coproduct, right and left difference). The communication between these two families is expressed in terms of Grishin's [10] distributivity principles. Figure 3.1 schematically presents the outline of the paper. In §3.2 we present **LG** in algebraic format and discuss the symmetries that govern the vocabulary of type-forming operations. In §3.3 we present a 'classical' term language for the **LG** type system, and we discuss how a term τ of type A is obtained as the Curry-Howard image of an **LG** sequent derivation π. In §3.4 we then study the CPS interpretation of types and terms, comparing the dual call-by-value $\lceil\cdot\rceil$ and call-by-name $\lfloor\cdot\rfloor$ regimes. Under the CPS interpretation, the classical terms for **LG** derivations are transformed into terms of the simply typed lambda calculus — the terms that code proofs in positive intuitionistic logic. The $\lambda_{\rightarrow}$ terms thus obtained adequately reflect NL meaning composition, and (unlike the terms for Multiplicative Linear Logic or its categorial equivalent **LP**) they are obtained in a structure-preserving way. In §3.5 we illustrate

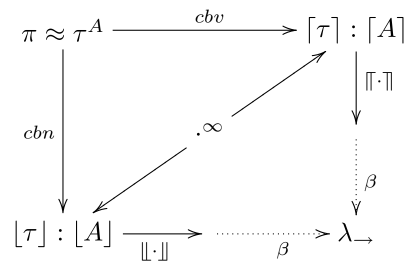


Figure 3.1: Outline of the paper

the approach with a discussion of scope construal. We investigate under what conditions the lexical constants of the original Lambek semantics can be lifted to the call-by-value $\lceil\!\lceil\cdot\rceil\!\rceil$ and/or call-by-name $\lfloor\!\lfloor\cdot\rfloor\!\rfloor$ level, and study how the $\lambda_{\rightarrow}$ terms one obtains after this transformation and β normalisation encode the different possibilities for scope construal. In the concluding section §3.6, we point to some directions for future work.

Relation to previous work Lambek [14] was the first paper to bring Grishin's work under the attention of a wider public. Lambek's bilinear systems are both stronger and weaker than what we propose here: they have hard-wired associativity for $\otimes, \oplus$, which means that control over constituent structure is lost; in addition, only half of the Grishin laws are taken into account ($G1$, $G3$ in Figure 3.2), an omission that precludes the account of non-peripheral scope construal presented here. De Groote [6] introduced $\lambda\mu$ calculus and continuations into the linguistic discussion of scope construal; Barker and Shan, in a series of papers ([2, 3, 20] among others), have been advocating this approach for a variety of semantic phenomena. We discuss the relation of our proposal to theirs in §3.6. Duality between the call-by-value and call-by-name evaluation strategies has been obtained in [19, 5, 21], among others. Our starting point is the Curien/Herbelin system because, in contrast to the other cited works, it has implication and difference as primitive operations.

3.2 The Lambek-Grishin Calculus

The calculus that we will use in this paper is presented in Figure 3.2. We refer to this system as **LG**. In (3.2), one finds the extended vocabulary of **LG**: the familiar Lambek operators $\otimes, \backslash, /$ are complemented with a family $\oplus, \oslash, \obslash$. In verbal communication, we pronounce $B\backslash A$ 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'. As usual under the formulas-as-types perspective, we can view the expressions of (3.2) as *types* with $\otimes, \backslash, /, \oplus, \oslash, \obslash$ as type-forming operations, or as logical *formulas*, with $\otimes, \backslash, /, \oplus, \oslash, \obslash$ as connectives.

A, B	$::=$	$p \mid$	atoms: s sentence, np noun phrase, ...
		$A \otimes B \mid B\backslash A \mid A/B \mid$	product, left vs right division (TYPES) tensor, left vs right implication (FORMULAS)
		$A \oplus B \mid A \oslash B \mid B \obslash A$	coproduct, right vs left difference (TYPES) cotensor, left vs right coimplication (FORMULAS)

(3.2)

$$
\begin{array}{lc}
\text{(pre-order)} & A \leq A \qquad \dfrac{A \leq B \quad B \leq C}{A \leq C} \\[2ex]
\text{(residuation)} & A \leq C/B \quad \text{iff} \quad A \otimes B \leq C \quad \text{iff} \quad B \leq A\backslash C \\[1ex]
\text{(dual residuation)} & B \obslash C \leq A \quad \text{iff} \quad C \leq B \oplus A \quad \text{iff} \quad C \oslash A \leq B \\[1ex]
\text{(interaction)} &
\begin{array}{llll}
(G1) & (A \obslash B) \otimes C \leq A \obslash (B \otimes C) & C \otimes (B \oslash A) \leq (C \otimes B) \oslash A & (G3) \\
(G2) & C \otimes (A \obslash B) \leq A \obslash (C \otimes B) & (B \oslash A) \otimes C \leq (B \otimes C) \oslash A & (G4)
\end{array}
\end{array}
$$

Figure 3.2: **LG**: symmetric Lambek calculus with Grishin interaction principles

The **LG** type system exhibits rich symmetries, discussed in full in [16]. In the present paper, two kinds of mirror symmetry will play an important role. The first $\cdot^{\bowtie}$ is internal to the $\otimes$ and $\oplus$ families and preserves the $\leq$ order; the second $\cdot^{\infty}$ relates the $\otimes$ and $\oplus$ families and is reverses the $\leq$ order. We have $p^{\bowtie} = p = p^{\infty}$ and the (bidirectional) translation tables of (3.3).

$$
\bowtie \quad \begin{array}{cccc} C/D & A \otimes B & B \oplus A & D \obslash C \\ \hline\hline D\backslash C & B \otimes A & A \oplus B & C \oslash D \end{array}
\qquad\qquad
\infty \quad \begin{array}{ccc} C/B & A \otimes B & A\backslash C \\ \hline\hline B \obslash C & B \oplus A & C \oslash A \end{array}
\tag{3.3}
$$

Algebraically, the Lambek operators $/, \otimes, \backslash$ form a residuated triple; likewise, the $\obslash, \oplus, \oslash$ family forms a dual residuated triple. The *minimal* symmetric categorial grammar consists of just the preorder axioms (reflexivity and transitivity of $\leq$) together with these (dual) residuation principles.[1]

Grishin's interaction principles The minimal symmetric system is of limited use if one wants to address the linguistic problems discussed in the introduction. In a system with just the (dual) residuation principles, for every theorem of the (non-associative) Lambek calculus, one also has its image under $\cdot^{\infty}$: $A \leq B$ iff $B^{\infty} \leq A^{\infty}$. Interaction between the $\otimes$ and the $\oplus$ family, however, is limited to gluing together theorems of the two families via cut. This limited interaction means that a formula from the $\oplus$ family which is trapped in a $\otimes$ context (or vice versa) will be inaccessible for logical manipulation.

The interaction principles proposed in [10] address this situation. Consider first $G1$ and $G2$ in Fig 3.2. On the lefthand side of the inequality, a coimplication $A \obslash B$ is hidden as the first or second coordinate of a product. The postulates invert the dominance relation between $\otimes$ and $\obslash$, raising the subformula A to a position where it can be shifted to the righthand side by means of the dual residuation principle. $G3$ and $G4$ are the images of $G1$ and $G2$ under $\cdot^{\bowtie}$. Similarly, a left or right implication trapped within a $\oplus$ context can be liberated by means of the $\cdot^{\infty}$ images in (3.4). Combined with transitivity, the Grishin postulates take the form of inference rules ($G1$: from $A \obslash (B \otimes C) \leq D$ conclude $(A \obslash B) \otimes C \leq D$, etc.)

$$
\begin{array}{llll}
(G3)^{\infty} & A\backslash(B \oplus C) \leq (A\backslash B) \oplus C & (C \oplus B)/A \leq C \oplus (B/A) & (G1)^{\infty}; \\
(G4)^{\infty} & A\backslash(C \oplus B) \leq C \oplus (A\backslash B) & (B \oplus C)/A \leq (B/A) \oplus C & (G2)^{\infty}.
\end{array}
\tag{3.4}
$$

[1] For a comprehensive overview (from a Display Logic perspective) of the substructural space of which **LG** is an inhabitant, the reader can consult [9]. De Groote and Lamarche [7] present sequent calculus and proof nets for a negation-tensor-par formulation of Classical Non-associative Lambek Calculus (CNL). **LG** is the subsystem of CNL given by the polarities of the operators in (3.2). CNL does not contain Grishin interaction principles.

The Grishin laws manifest themselves in many forms. The key observation for their usefulness in the analysis of scope construal lies in the fact that $(B \oslash C) \otimes A \leq C/(A\backslash B)$ is a theorem of **LG**. This means that a Lambek type $s/(np\backslash s)$ is derivable from a $(s \oslash s) \otimes np$ type; what can be done with the former can also be done with the latter, but the coimplication type also has non-local capabilities thanks to the Grishin interactions.

Apart from the interaction principles $G1$–$G4$ (and their duals) which will be at the heart of our analysis of scope construal, Grishin considers other options for generalizing Lambek calculus. The reader is referred to [16] for a discussion of these options and their potential linguistic uses. Also in [16] one finds a decision procedure for **LG** based on the monotonicity laws for the type-forming operations, together with the residuation principles and the Grishin principles in rule form.

3.3 Proofs and Terms

The term language we will use for **LG** derivations is a directional version of Curien/Herbelin's classical $\overline{\lambda}\mu\widetilde{\mu}$ terms which takes the $\cdot^{\bowtie}$ symmetry into account. The term language distinguishes terms, coterms (contexts) and commands. We give the syntax of the term language in (3.5). For terms, we use x, M, N; for coterms (contexts) α, K, L; commands are cuts $M * L$ between a term M and a coterm L. We overload the notation, writing $x \setminus M$ versus $M \,/\, x$ for the left and right abstraction constructs; similarly for coabstraction. As discussed in §3.1, the familiar lambda abstraction of $\lambda_{\rightarrow}$ will be reinstalled as a result of the CPS transformation on the terms of (3.5).

	$x \in \mathsf{Term}^A$	if	$x \in \mathsf{Var}^A$
(L ABSTRACTION)	$x\backslash M \in \mathsf{Term}^{B\backslash A}$	if	$x \in \mathsf{Var}^B, M \in \mathsf{Term}^A$
(R ABSTRACTION)	$M/x \in \mathsf{Term}^{A/B}$	if	$x \in \mathsf{Var}^B, M \in \mathsf{Term}^A$
(L COAPPLICATION)	$K \prec M \in \mathsf{Term}^{B \otimes A}$	if	$K \in \mathsf{CoTerm}^B, M \in \mathsf{Term}^A$
(R COAPPLICATION)	$M \succ K \in \mathsf{Term}^{A \oslash B}$	if	$K \in \mathsf{CoTerm}^B, M \in \mathsf{Term}^A$
(R SHIFT)	$\mu\alpha.(x * K) \in \mathsf{Term}^B$	if	$\alpha \in \mathsf{CoVar}^B, x \in \mathsf{Var}^A, K \in \mathsf{CoTerm}^A$
	$\alpha \in \mathsf{CoTerm}^A$	if	$\alpha \in \mathsf{CoVar}^A$
(L APPLICATION)	$M \ltimes K \in \mathsf{CoTerm}^{B\backslash A}$	if	$K \in \mathsf{CoTerm}^A, M \in \mathsf{Term}^B$
(R APPLICATION)	$K \rtimes M \in \mathsf{CoTerm}^{A/B}$	if	$K \in \mathsf{CoTerm}^A, M \in \mathsf{Term}^B$
(L COABSTR)	$\alpha \otimes K \in \mathsf{CoTerm}^{B \otimes A}$	if	$\alpha \in \mathsf{CoVar}^B, K \in \mathsf{CoTerm}^A$
(R COABSTR)	$K \oslash \alpha \in \mathsf{CoTerm}^{A \oslash B}$	if	$\alpha \in \mathsf{CoVar}^B, K \in \mathsf{CoTerm}^A$
(L SHIFT)	$\widetilde{\mu}x.(M * \alpha) \in \mathsf{CoTerm}^A$	if	$x \in \mathsf{Var}^A, M \in \mathsf{Term}^B, \alpha \in \mathsf{CoVar}^B$

(3.5)

As in the case of the Lambek calculus, for **LG** we are interested in the resource-sensitive sublanguage. This means that the (co)abstraction and (co)application cases are subject to a linearity condition: the (co)variable bound in a (co)abstraction occurs free exactly once in the body; in the (co)application case the sets of free (co)variables of the term and coterm involved are disjoint. Our use of cut is restricted to patterns $x * K$ $(M * \alpha)$ in the shift right (left) construct, where μ $(\widetilde{\mu})$ obeys the single-bind restriction.

The dualities we discussed for the type system extend to the term language: (3.6) and (3.7).

The latter acts on the directional constructs; identity otherwise.

$$\begin{array}{rclrcl} x^\infty & = & \alpha & \alpha^\infty & = & x; \\ (x \setminus M)^\infty & = & M^\infty \oslash \alpha & (K \oslash \alpha)^\infty & = & x \setminus K^\infty; \\ (M \,/\, x)^\infty & = & \alpha \obslash M^\infty & (\alpha \obslash K)^\infty & = & K^\infty \,/\, x; \\ (M \succ K)^\infty & = & K^\infty \ltimes M^\infty & (M \ltimes K)^\infty & = & K^\infty \succ M^\infty; \\ (K \prec M)^\infty & = & M^\infty \rtimes K^\infty & (K \rtimes M)^\infty & = & M^\infty \prec K^\infty; \\ (\mu\beta.(x * K))^\infty & = & \widetilde{\mu} y.(K^\infty * \alpha) & (\widetilde{\mu} y.(M * \alpha))^\infty & = & \mu\beta.(x * M^\infty). \end{array} \tag{3.6}$$

$$\begin{array}{rr} (M \succ K)^{\bowtie} = K^{\bowtie} \prec M^{\bowtie} & (M \ltimes K)^{\bowtie} = K^{\bowtie} \rtimes M^{\bowtie}; \\ (x \setminus M)^{\bowtie} = M^{\bowtie} \,/\, x & (K \oslash \alpha)^{\bowtie} = \alpha \obslash K^{\bowtie}. \end{array} \tag{3.7}$$

3.3.1 LG Sequent Calculus

In Lambek calculus, sequents are statements $\Gamma \Rightarrow B$, where Γ is a binary tree with formulas $A_1,\ldots,A_n$ at the yield, and B is a single formula. The structure-building operation which puts together the antecedent tree is the counterpart of the $\otimes$ logical operation. In **LG**, sequents $\Gamma \Rightarrow \Delta$ can have structures both in the antecedent and in the succedent. The sequent interpunction (which we write $\cdot \circ \cdot$) is the structural counterpart of $\otimes$ in the antecedent, and of $\oplus$ in the succedent. Notice that in the absence of associativity, $\circ$ is a binary operation.

In the rules below, we decorate **LG** derivations with the terms of (3.5). We distinguish sequents $\Gamma \xrightarrow{M} \Delta[B]$ and cosequents $\Gamma[A] \xrightarrow{K} \Delta$ with proof term M and coterm K respectively. A sequent (cosequent) has precisely one active succedent (antecedent) formula. The active formula is unlabeled. The passive antecedent (succedent) formulas are labeled with distinct variables x_i (covariables α_i).

For the axiomatic case, we distinguish two versions, depending on whether the succedent or the antecedent is the active formula. The rules ($\leftharpoondown$) and ($\rightleftharpoons$) make it possible to shift the focus from antecedent to succedent or vice versa. These rules are in fact restricted cuts, where one of the premises is axiomatic (Axiom or Co-Axiom).

$$\frac{}{x : A \xrightarrow{x} A}\ \text{Ax} \qquad \frac{}{A \xrightarrow{\alpha} \alpha : A}\ \text{Co-Ax}\ . \tag{3.8}$$

$$\frac{\Gamma[A] \xrightarrow{K} \Delta[\alpha : B]}{\Gamma[x : A] \xrightarrow{\mu\alpha.(x * K)} \Delta[B]}\ (\leftharpoondown) \qquad \frac{\Gamma[x : A] \xrightarrow{M} \Delta[B]}{\Gamma[A] \xrightarrow{\widetilde{\mu} x.(M * \alpha)} \Delta[\alpha : B]}\ (\rightharpoonup)\ . \tag{3.9}$$

Let us now consider the sequent left and right rules for the connectives. We restrict attention to the (co)implication fragment, i.e. we only cater for $\otimes$ and $\oplus$ in their 'structural' form $\circ$ as antecedent resp. succedent punctuation. The rules of use for the (co)implications are given in (3.10): these are two-premise rules, introducing an implication (coimplication) in the antecedent (succedent). Notice that we find the $\cdot^{\bowtie}$ and $\cdot^{\infty}$ symmetries here at the level of the inference rules, with $\cdot^{\infty}$ ($\cdot^{\bowtie}$) relating pairs of rules in the horizontal (vertical) dimension.

$$\frac{B \xrightarrow{K} \Delta \quad \Delta' \xrightarrow{M} \Gamma[A]}{\Delta' \xrightarrow{M \succ K} \Gamma[(A \oslash B) \circ \Delta]} \ (\oslash R) \qquad \frac{\Delta \xrightarrow{M} B \quad \Gamma[A] \xrightarrow{K} \Delta'}{\Gamma[\Delta \circ (B\backslash A)] \xrightarrow{M \ltimes K} \Delta'} \ (\backslash L)$$

$$\begin{array}{ccc} (\oslash R) & \leftarrow \cdot^{\infty} \rightarrow & (\backslash L) \\ \updownarrow \cdot^{\bowtie} & & \updownarrow \cdot^{\bowtie} \\ (\obslash R) & \leftarrow \cdot^{\infty} \rightarrow & (/L) \end{array} \tag{3.10}$$

$$\frac{B \xrightarrow{K} \Delta \quad \Delta' \xrightarrow{M} \Gamma[A]}{\Delta' \xrightarrow{K \prec M} \Gamma[\Delta \circ (B \obslash A)]} \ (\obslash R) \qquad \frac{\Delta \xrightarrow{M} B \quad \Gamma[A] \xrightarrow{K} \Delta'}{\Gamma[(A/B) \circ \Delta] \xrightarrow{K \rtimes M} \Delta'} \ (/L)$$

The rules of proof for the (co)implications are given in (3.11): these are one-premise rules, introducing an implication (coimplication) in the succedent (antecedent).

$$\frac{x : B \circ \Gamma \xrightarrow{M} \Delta[A]}{\Gamma \xrightarrow{x \backslash M} \Delta[B\backslash A]} \ (\backslash R) \qquad \frac{\Gamma[A] \xrightarrow{K} \Delta \circ \alpha : B}{\Gamma[A \oslash B] \xrightarrow{K \oslash \alpha} \Delta} \ (\oslash L)$$

$$\begin{array}{ccc} (\backslash R) & \leftarrow \cdot^{\infty} \rightarrow & (\oslash L) \\ \updownarrow \cdot^{\bowtie} & & \updownarrow \cdot^{\bowtie} \\ (/R) & \leftarrow \cdot^{\infty} \rightarrow & (\obslash L) \end{array} \tag{3.11}$$

$$\frac{\Gamma \circ x : B \xrightarrow{M} \Delta[A]}{\Gamma \xrightarrow{M / x} \Delta[A/B]} \ (/R) \qquad \frac{\Gamma[A] \xrightarrow{K} \alpha : B \circ \Delta}{\Gamma[B \obslash A] \xrightarrow{\alpha \obslash K} \Delta} \ (\obslash L)$$

Observe that to prove the soundness of the coimplication (implication) rules of proof from the algebraic presentation, one uses the Grishin interaction principles to move the B subformula upwards through the $\otimes$ context ($\oplus$ context), and then shifts it to the succedent (antecedent) part via the residuation principles. The Grishin interaction principles, in other words, are *absorbed* in these rules of proof. We illustrate this in (3.12) for $(\backslash R)$, writing $\Gamma^\bullet$ (Δ°) for the formula equivalent of an antecedent (succedent) structure. The vertical dots abbreviate a succession of Grishin interaction steps.

$$\begin{array}{c} \dfrac{B \otimes \Gamma^\bullet \leq \Delta^\circ[A]}{\Gamma^\bullet \leq B\backslash \Delta^\circ[A]} \\ \vdots \\ \Gamma^\bullet \leq \Delta^\circ[B\backslash A] \end{array} \tag{3.12}$$

As indicated in §3.2, we will use the formula $(B \oslash C) \obslash A$ to do the work of the *in situ* binder schema $q(A, B, C)$. (Alternatively, we could have used its $\cdot^{\bowtie}$ dual $A \oslash (C \obslash B)$.) The (qL) and (qR) rules of Fig 3.3 have the status of derived inference rules. We will use them in §3.5 to present proofs and terms in a more compact format. In §3.7 we give a worked-out derivation of $(B \oslash C) \obslash A \Rightarrow C/(A\backslash B)$, together with further abbreviatory conventions. The reader may want

$$\dfrac{C \xrightarrow{L} \Delta \quad \Gamma[x:A] \xrightarrow{N} B}{\Gamma[(B \oslash C) \obslash A] \xrightarrow{\texttt{bind}(x,N,L)} \Delta}\ (qL) \quad \triangleq \quad \dfrac{\dfrac{\dfrac{C \xrightarrow{L} \Delta \qquad \Gamma[x:A] \xrightarrow{N} B}{\Gamma[x:A] \xrightarrow{N \succ L} (B \oslash C) \circ \Delta}\ (\oslash R)}{\Gamma[A] \xrightarrow{\widetilde{\mu}x.((N \succ L) * \gamma)} \gamma : (B \oslash C) \circ \Delta}\ \rightleftharpoons}{\Gamma[(B \oslash C) \obslash A] \xrightarrow{\gamma \obslash (\widetilde{\mu}x.((N \succ L) * \gamma))} \Delta}\ (\obslash L)$$

$$\dfrac{B \xrightarrow{K} \Delta' \circ \beta : C \quad \Gamma \xrightarrow{M} \Delta[A]}{\Gamma \xrightarrow{\texttt{cobind}(M,K,\beta)} \Delta[\Delta' \circ ((B \oslash C) \obslash A)]}\ (qR) \quad \triangleq \quad \dfrac{\dfrac{B \xrightarrow{K} \Delta' \circ \beta : C}{B \oslash C \xrightarrow{K \oslash \beta} \Delta'}\ (\oslash L) \qquad \Gamma \xrightarrow{M} \Delta[A]}{\Gamma \xrightarrow{(K \oslash \beta) \prec M} \Delta[\Delta' \circ ((B \oslash C) \obslash A)]}\ (\obslash R)$$

Figure 3.3: Derived inference rules for $(B \oslash C) \obslash A \approx q(A, B, C)$.

to check that a cut of (qR) against (qL) can be rewritten with cuts on the subformulae A, B, C, as required: $\texttt{cobind}(M^A, K^B, \beta^C) * \texttt{bind}(x^A, N^B, L^C) \longrightarrow_\beta M * \widetilde{\mu}x.(\mu\beta.(N * K) * L)$. One should keep in mind that (qL) and (qR) are short-cuts, i.e. ways of abbreviating a sequence of n inference steps as a one-step inference. For some theorems of **LG**, one cannot take a short-cut: their derivation requires the individual inference rules for the connectives involved. The valid type transition $(B \oslash C) \obslash A \Rightarrow ((D\backslash B) \oslash (D\backslash C)) \obslash A$ is an example.

3.4 Interpretation: Continuation Semantics

We turn now to an interpretation for **LG** derivations in the continuation-passing-style (CPS). In the semantics of programming languages, CPS interpretation has been a fruitful strategy to make explicit (and open to manipulation) aspects of computation that remain implicit in a direct interpretation. In the direct interpretation, a function simply returns a value. Under the CPS interpretation, functions are provided with an extra argument for the continuation of the computation. This explicit continuation argument is then passed on when functions combine with each other. Key concepts, then, are “computation”, “continuation” and “value” and the way they relate to each other for different evaluation strategies.

Curien and Herbelin [5] develop the CPS interpretation for a classical system with an implication and a difference operation; call-by-value (cbv) $\lceil\cdot\rceil$ and call-by-name (cbn) $\lfloor\cdot\rfloor$ regimes are related by the duality between the implication and difference operations. For **LG** we refine the Curien/Herbelin continuation semantics to accommodate the left/right symmetry. We first consider the effect of the CPS interpretation on the level of *types*, comparing a call-by-value (cbv) and a call-by-name (cbn) regime; then we define the CPS interpretation on the level of the *terms* of (3.5).

Types: call-by-value The target type language has a distinguished type R of responses, products and functions; all functions have range R. For each type A of the source language, the target language has values $V_A = \lceil A \rceil$, continuations $K_A = R^{V_A}$ (functions from V_A to R)

and computations $C_A = R^{K_A}$ (functions from K_A to R).[2] Notice that given the canonical isomorphism $A \times B \to C \cong A \to (B \to C)$, one can also think of a $V_{A\backslash B}$ as a function from A values to B computations. For p atomic, $\lceil p \rceil = p$. In (3.13), the $\lceil\cdot\rceil$ translations for the (co)implications are related in the vertical dimension by left/right symmetry $\cdot^{\bowtie}$ and in the horizontal dimension by arrow reversal $\cdot^{\infty}$: right difference is dual to left division, left difference dual to right division.

$$\begin{array}{rclcrcl} \lceil A\backslash B \rceil & = & R^{\lceil A \rceil \times R^{\lceil B \rceil}} & & \lceil B \oslash A \rceil & = & \lceil B \rceil \times R^{\lceil A \rceil}; \\ \lceil B/A \rceil & = & R^{R^{\lceil B \rceil} \times \lceil A \rceil} & & \lceil A \obslash B \rceil & = & R^{\lceil A \rceil} \times \lceil B \rceil. \end{array} \tag{3.13}$$

Types: call-by-name Under the call-by-name regime, for each type A of the source language, the target language has continuations $K_A = \lfloor A \rfloor$ and computations $C_A = R^{K_A}$. The call-by-name interpretation $\lfloor\cdot\rfloor$ is obtained as the composition of the $\cdot^{\infty}$ duality map and the $\lceil\cdot\rceil$ interpretation: $\lfloor A \rfloor \triangleq \lceil A^{\infty} \rceil$. For atoms, $\lfloor p \rfloor = \lceil p^{\infty} \rceil = p$. For the (co)implications, compare the cbv intepretation (left) with the cbn interpretation (right) in (3.14).

$$\begin{array}{rclcrcl} \lceil A\backslash B \rceil & = & R^{\lceil A \rceil \times R^{\lceil B \rceil}} & & R^{\lfloor A \rfloor \times R^{\lfloor B \rfloor}} & = & \lfloor B \oslash A \rfloor; \\ \lceil B \oslash A \rceil & = & \lceil B \rceil \times R^{\lceil A \rceil} & & \lfloor B \rfloor \times R^{\lfloor A \rfloor} & = & \lfloor A\backslash B \rfloor; \\ \lceil B/A \rceil & = & R^{R^{\lceil B \rceil} \times \lceil A \rceil} & & R^{R^{\lfloor B \rfloor} \times \lfloor A \rfloor} & = & \lfloor A \obslash B \rfloor; \\ \lceil A \obslash B \rceil & = & R^{\lceil A \rceil} \times \lceil B \rceil & & R^{\lfloor A \rfloor} \times \lfloor B \rfloor & = & \lfloor B/A \rfloor. \end{array} \tag{3.14}$$

Notice that for the call-by-name regime, the starting point is the level of continuations, not values as under call-by-value. Let's take the definition of $B \oslash A$ by means of example. For call-by-value, one starts from $\lceil B \oslash A \rceil$ (i.e., $V_{B \oslash A}$) that is a pair $V_B \times K_A$; hence its continuation is $K_{B \oslash A} = (V_B \times K_A) \to R$ and its computation is $C_{B \oslash A} = ((V_B \times K_A) \to R) \to R$. On the other hand, the call-by-name interpretation starts at the level of continuations: $\lfloor B \oslash A \rfloor = (K_A \times C_B) \to R$ and from this the computation is obtained as usual, viz. $C_{B \oslash A} = ((K_A \times C_B) \to R) \to R$, hence obtaining a higher order function than the one computed under the call-by-value strategy. This difference will play an important role in the linguistic application of the two strategies.

Terms: cbv versus cbn Given the different CPS *types* for left and right (co)implications, we can now turn to their interpretation at the *term* level. In (3.15), we give the cbv interpretation of terms, in (3.16) of coterms. We repeat the typing information from (3.5) to assist the reader. The call-by-name regime is the composition of call-by-value and arrow reversal: $\lfloor\cdot\rfloor \triangleq \lceil \cdot^{\infty} \rceil$. This CPS interpretation of terms is set up in such a way that for sequents with yield $A_1, \ldots, A_n \Rightarrow B$, the cbv interpretation represents the process of obtaining a B *computation* from $A_1, \ldots, A_n$ *values*; the cbn interpretation takes $A_1, \ldots, A_n$ *computations* to a B computation. See Propositions 8.1 and 8.3 of [5].

[2] In the schemas (3.13) and (3.14) we use exponent notation for function spaces, for comparison with [5]. In the text, we usually shift to the typographically more convenient arrow notation, compare A^B versus $B \to A$.

$$
\begin{array}{rrcll}
A & \lceil x \rceil & = & \lambda k.k\ x & x : A \\
B\backslash A & \lceil x \backslash M \rceil & = & \lambda k.(k\ \lambda\langle x, \beta\rangle.\lceil M \rceil\ \beta) & x : B,\ M : A \\
A/B & \lceil M / x \rceil & = & \lambda k.(k\ \lambda\langle \beta, x\rangle.\lceil M \rceil\ \beta) & x : B,\ M : A \\
B \oslash A & \lceil M \succ K \rceil & = & \lambda k.(\lceil M \rceil\ \lambda y.(k\ \langle y, \lceil K \rceil\rangle)) & M : A,\ K : B \\
A \oslash B & \lceil K \prec M \rceil & = & \lambda k.(\lceil M \rceil\ \lambda y.(k\ \langle \lceil K \rceil, y\rangle)) & M : A,\ K : B \\
B & \lceil \mu\alpha.(x * K) \rceil & = & \lambda\alpha.(\lceil K \rceil\ x) & \alpha : B,\ x, K : A
\end{array}
\tag{3.15}
$$

$$
\begin{array}{rrcll}
A & \lceil \alpha \rceil & = & \alpha & \alpha : A \\
B \oslash A & \lceil \alpha \oslash K \rceil & = & \lambda\langle \alpha, x\rangle.(\lceil K \rceil\ x) & \alpha : B,\ K : A \\
A \oslash B & \lceil K \oslash \alpha \rceil & = & \lambda\langle x, \alpha\rangle.(\lceil K \rceil\ x) & \alpha : B,\ K : A \\
B\backslash A & \lceil M \ltimes K \rceil & = & \lambda k.(\lceil M \rceil\ \lambda x.(k\ \langle x, \lceil K \rceil\rangle)) & M : B,\ K : A \\
A/B & \lceil K \rtimes M \rceil & = & \lambda k.(\lceil M \rceil\ \lambda x.(k\ \langle \lceil K \rceil, x\rangle)) & M : B,\ K : A \\
A & \lceil \widetilde{\mu} x.(M * \alpha) \rceil & = & \lambda x.(\lceil M \rceil\ \alpha) & x : A,\ \alpha, M : B
\end{array}
\tag{3.16}
$$

3.5 Application: Scope Construal

In this section we turn to the linguistic application. Our aim is twofold. First we show that a type assignment $(s \oslash s) \oslash np$ for generalized quantifier phrases solves the problems with $s/(np\backslash s)$ mentioned in §3.1: the type $(s \oslash s) \oslash np$ uniformly appears in positions that can be occupied by ordinary noun phrases, and it gives rise to ambiguities of scope construal (local and non-local) in constructions with multiple GQ and/or multiple choices for the scope domain. Second, we relate the CPS interpretation to the original interpretation for Lambek derivations by defining translations $\lceil\!\lceil \cdot \rceil\!\rceil, \lfloor\!\lfloor \cdot \rfloor\!\rfloor$ lifting the lexical constants from the type they have in the original Lambek semantics to the type required by the cbv or cbn level. To realize this second aim, we assume that the result type R is the type of truth values. For the rest our modeltheoretic assumptions are standard. The domain of interpretation for np values is E (the set of individuals), for s values it is $\{0, 1\}$ (the set of truth values). Out of E and $\{0, 1\}$ one then constructs complex domains in terms of function spaces and Cartesian products. In the case of the original Lambek semantics (or Montague grammar) these domains of interpretation are obtained indirectly, via a mapping between syntactic and semantic types where $np' = e$, $s' = t$ and $(A\backslash B) = (B/A) = A' \rightarrow B'$, with $\text{Dom}_e = E$ and $\text{Dom}_t = \{0, 1\}$.

We start with a fully worked-out example of a 'S goes to NP VP' combination, 'Alice left'. The official sequent derivation is given in (3.17). Consulting the dictionary of Table 3.1, we fill in the lexical items alice and left of type np and $np\backslash s$ respectively for the x and y parameters of the proof term. Call the resulting term M. We now compare the CPS transformation of M under the cbv and cbn execution regimes as defined in (3.15) and (3.16).

$$
\cfrac{\cfrac{x : np \xrightarrow{x} np \quad s \xrightarrow{\alpha} \alpha : s}{x : np \circ np\backslash s \xrightarrow{x \ltimes \alpha} \alpha : s}\ (\backslash L)}{x : np \circ y : np\backslash s \xrightarrow{\mu\alpha.(y * (x \ltimes \alpha))} s}\ (\leftrightharpoons)
\tag{3.17}
$$

Consider first cbv, on the left in Figure 3.4. Recall that under the cbv regime a sequent with yield $A_1, \ldots, A_n \Rightarrow B$ maps the A_i values to a B computation. A value of type $np\backslash s$ ($V_{np\backslash s}$), as we see in Table 3.1, is a function taking a pair of an np value and an s continuation to the result type R, i.e. $V_{np} \times K_s \rightarrow R$. Combining V_{np} and $V_{np\backslash s}$ we obtain an s computation,

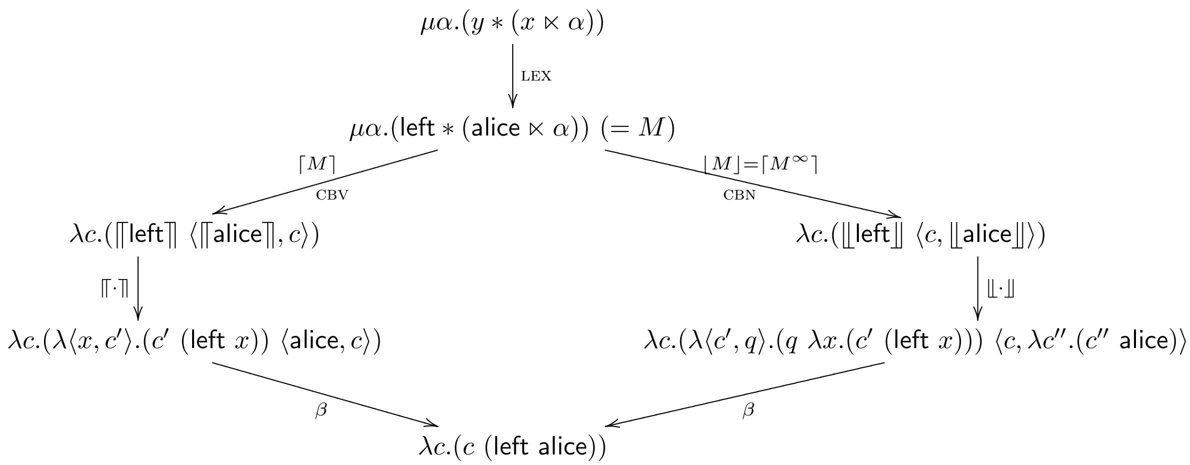


Figure 3.4: 'Alice left': cbv versus cbn

Table 3.1: Lexical entries. $\lceil A \rceil$ is a value of type A; $\lfloor A \rfloor$ is a continuation of type A.

WORD	TYPE	ALIAS	$\lceil\cdot\rceil$ CBV	$\lfloor\cdot\rfloor$ CBN
alice, lewis	np		$\lceil np \rceil$	$\lfloor np \rfloor$
left	$np\backslash s$	iv	$R^{\lceil np \rceil \times R^{\lceil s \rceil}}$	$\lfloor s \rfloor \times R^{\lfloor np \rfloor}$
teases	$(np\backslash s)/np$	tv	$R^{R^{\lceil iv \rceil} \times \lceil np \rceil}$	$R^{\lfloor np \rfloor} \times \lfloor iv \rfloor$
thinks	$(np\backslash s)/s$	tvs	$R^{R^{\lceil iv \rceil} \times \lceil s \rceil}$	$R^{\lfloor s \rfloor} \times \lfloor iv \rfloor$
somebody	$s/(np\backslash s)$	su	$R^{R^{\lceil s \rceil} \times \lceil iv \rceil}$	$R^{\lfloor iv \rfloor} \times \lfloor s \rfloor$

Table 3.2: Lifting lexical constants: cbv regime

$$\begin{aligned}
[\![\mathsf{alice}]\!] &= \mathsf{alice}\\
[\![\mathsf{lewis}]\!] &= \mathsf{lewis}\\
[\![\mathsf{left}]\!] &= \lambda\langle x, c\rangle.(c\ (\mathsf{left}\ x))\\
[\![\mathsf{teases}]\!] &= \lambda\langle v, y\rangle.(v\ \lambda\langle x, c\rangle.(c\ ((\mathsf{teases}\ y)\ x)))\\
[\![\mathsf{somebody}]\!] &= \lambda\langle c, v\rangle.(\exists\ \lambda x.(v\ \langle x, c\rangle))
\end{aligned}$$

i.e. $(s \to R) \to R$, by giving the $V_{np\backslash s}$ function the pair it requires, and abstracting over the K_s component. This is what we see in $\lceil M \rceil = \lambda c.([\![\mathsf{left}]\!]\ \langle [\![\mathsf{alice}]\!], c\rangle)$. The next step is to relate the cbv CPS interpretation to the original semantics of the Lambek calculus. In the original semantics, 'Alice' and 'left' would be interpreted in terms of constants of type e and $e \to t$ respectively. The mapping $[\![\cdot]\!]$ in Table 3.2 produces terms of type V_{np} and $V_{np\backslash s}$ from constants alice and left of type e and $e \to t$. Substituting these in $\lceil M \rceil$ (and β conversion) gives $\lambda c.(c\ (\mathsf{left\ alice}))$. Combined with the trivial s continuation (the identity function on $\{0,1\}$) one obtains (left alice).

On the right in Figure 3.4 we have the cbn interpretation. Under the cbn regime, a sequent with yield $A_1, \ldots, A_n \Rightarrow B$ maps A_i *computations* to a B computation. We obtain the cbn interpretation by duality: $\lfloor M \rfloor = \lceil M^\infty \rceil = \lceil \widetilde{\mu} x.((x \succ \mathsf{alice}) * \mathsf{left}) \rceil$, i.e. we read off the cbn interpretation from the mirror image derivation $(np \circ np\backslash s \Rightarrow s)^\infty$, which is $s \Rightarrow s \oslash np \circ np$. For the lexical items involved, Table 3.1 gives the *continuation* types $\lfloor A \rfloor$ corresponding to the

Table 3.3: Lifting lexical constants: cbn regime

$$\begin{aligned}
\lfloor\!\lfloor \mathsf{alice} \rfloor\!\rfloor &= \lambda c.(c\ \mathsf{alice})\\
\lfloor\!\lfloor \mathsf{lewis} \rfloor\!\rfloor &= \lambda c.(c\ \mathsf{lewis})\\
\lfloor\!\lfloor \mathsf{left} \rfloor\!\rfloor &= \lambda\langle c,q\rangle.(q\ \lambda x.(c\ (\mathsf{left}\ x)))\\
\lfloor\!\lfloor \mathsf{teases} \rfloor\!\rfloor &= \lambda\langle q,\langle c,q'\rangle\rangle.(q'\ \lambda x.(q\ \lambda y.(c\ ((\mathsf{teases}\ y)\ x))))\\
\lfloor\!\lfloor \mathsf{somebody} \rfloor\!\rfloor &= \lambda\langle v,c\rangle.(\exists\ \lambda x.(v\ \langle c,\lambda c'.(c'\ x)\rangle))
\end{aligned}$$

source language types A. To obtain the required types for A *computations*, we take functions from these continuations into R. Specifically, the verb $\lfloor\!\lfloor \mathsf{left} \rfloor\!\rfloor$ in this case is interpreted as a function taking a pair $K_s \times C_{np}$ into R; the noun phrase argument $\lfloor\!\lfloor \mathsf{alice} \rfloor\!\rfloor$ also is of type C_{np}, i.e. $(V_{np} \to R) \to R$. Combining these produces a result of type C_s again by abstracting over the K_s component of the pair given to the verb: $\lfloor M \rfloor = \lambda c.(\lfloor\!\lfloor \mathsf{left} \rfloor\!\rfloor\ \langle c, \lfloor\!\lfloor \mathsf{alice} \rfloor\!\rfloor\rangle)$. The mapping $\lfloor\!\lfloor \cdot \rfloor\!\rfloor$ in Table 3.3 produces terms of type $\lfloor np \rfloor \to R$ and $\lfloor np\backslash s \rfloor \to R$ (i.e. computations) from constants alice and left of type e and $e \to t$. Substituting these in $\lfloor M \rfloor$ (and β conversion) gives the same result as what we had under the cbv regime.

The reader is invited to go throught the same steps for 'Alice teases Lewis'. The term for the derivation is $\mu\alpha.(\mathsf{teases} * ((\mathsf{alice} \ltimes \alpha) \rtimes \mathsf{lewis}))$ $(= M)$, with the CPS interpretations in (3.18). The variable v is of type $K_{np\backslash s}$, a verb phrase continuation. Consulting the cbv and cbn dictionaries of Tables 3.2 and 3.3, we can substitute the required lambda terms for the lexical constants. After this substitution and β reduction, the cbv and cbn interpretations converge on $\lambda c.(c\ ((\mathsf{teases}\ \mathsf{lewis})\ \mathsf{alice}))$.

$$\begin{aligned}
\lceil M \rceil &= \lambda c.(\lceil\!\lceil \mathsf{teases} \rceil\!\rceil\ \langle \lambda v.(v\ \langle \lceil\!\lceil \mathsf{alice} \rceil\!\rceil, c\rangle), \lceil\!\lceil \mathsf{lewis} \rceil\!\rceil\rangle);\\
\lfloor M \rfloor &= \lambda c.(\lfloor\!\lfloor \mathsf{teases} \rfloor\!\rfloor\ \langle \lfloor\!\lfloor \mathsf{lewis} \rfloor\!\rfloor, \langle c, \lfloor\!\lfloor \mathsf{alice} \rfloor\!\rfloor\rangle\rangle).
\end{aligned} \tag{3.18}$$

In Fig 3.5 we highlight the type structure of the CPS interpretations for this sentence, showing that (i) call-by-value produces terms consisting of function applications of values to pairs of values and continuations (left tree), whereas (ii) call-by-name produces terms consisting of the application of computation to pairs of computations and continuation types. The observed difference will be relevant for the interpretation of generalized quantifiers expressions to which we now turn.

Scope construal: simple subject GQ In Table 3.1, one finds the CPS image under cbv and cbn of a Lambek-style $s/(np\backslash s)$ type assignment for a GQ expression such as 'somebody'. The corresponding lexical recipes for the cbv and cbn regimes are given in Tables 3.2 and 3.3, respectively. We leave it as an exercise for the reader to work through a derivation with these types/terms and to verify that a type assignment $s/(np\backslash s)$ is restricted to subject position and to local scope, as we saw in §3.1. Let us turn then to derivations with the type assignment we have proposed: $(s \oslash s) \obackslash np$ (alias: gq) — a type assignment that will accommodate both local and non-local scope construals. In (3.19) we compute the term for the derivation of 'somebody left', using the abbreviatory conventions discussed in the Appendix (e.g., the $\boxed{1}$ in step 2. stands for the proof term computed at step 1.); in (3.20) its CPS transformation under cbv and cbn. $(z : V_s,\ q : C_{np},\ y : K_{s \oslash s})$

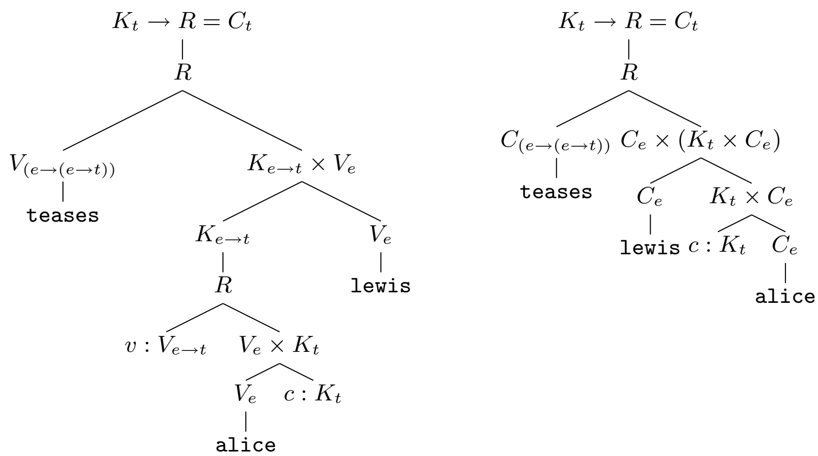
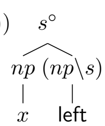
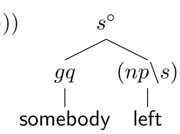


Figure 3.5: Type schemas for cbv (left) versus cbn (right).

1. $\mu\alpha.(\mathsf{left}*(x \ltimes \alpha))$



2. $\mu\beta.(\mathsf{somebody}*\mathtt{bind}(x, \boxed{1}, \beta)))$



(3.19)

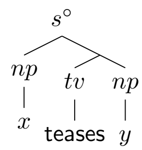
$$\boxed{2} = N \quad \begin{array}{rcl} \lceil N \rceil & = & \lambda c.((\lceil\!\lceil \mathsf{left} \rceil\!\rceil\ \langle \pi^2 \lceil\!\lceil \mathsf{somebody} \rceil\!\rceil, \lambda z.(\pi^1 \lceil\!\lceil \mathsf{somebody} \rceil\!\rceil\ \langle z, c \rangle)\rangle); \\ \lfloor N \rfloor & = & \lambda c.((\lfloor\!\lfloor \mathsf{somebody} \rfloor\!\rfloor\ \lambda\langle q, y\rangle.(y\ \langle c, \lambda c'.(\lfloor\!\lfloor \mathsf{left} \rfloor\!\rfloor\ \langle c', q \rangle)\rangle))). \end{array} \tag{3.20}$$

The difference between cbv and cbn regimes observed above with respect to $\lceil\!\lceil \cdot \rceil\!\rceil, \lfloor\!\lfloor \cdot \rfloor\!\rfloor$ in the term in (3.18) turns out to be of particular interest here. In (3.21) we give V_{gq} and K_{gq} for the cbv and cbn interpretations respectively. For cbv, this is a pair of an $s \oslash s$ continuation and an np value. This np value would have to be realised as the variable bound by the (logical) constant $\exists$ (of type $(e \to t) \to t$) in the $\lceil\!\lceil \cdot \rceil\!\rceil$ translation. Such a binding relation cannot be established from the $K_{s \oslash s}$ component of the pair. For cbn, the situation is different: $\lfloor gq \rfloor$ has the required structure to specify the lifted lexical recipe of (3.22). $\mathcal{Q}$ is a function taking a pair of a np computation and an $(s \oslash s)$ continuation to the result type R. In the body of the term, we apply $\mathcal{Q}$ to the C_{np} term $\lambda k.(k\ x)$, where x is the e type variable bound by $\exists$, and to the closed term $\lambda\langle c, p\rangle.(p\ c)$ obtained by applying the C_s variable p to the K_s variable c. In combination with $\lfloor\!\lfloor \mathsf{left} \rfloor\!\rfloor$ given above, $\lfloor N \rfloor$ simplifies to $\lambda c.(\exists\ \lambda x.(c\ (\mathsf{left}\ x)))$ as required. From here on, we stay with the cbn regime.

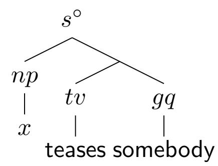
$$\lceil gq \rceil \ = \ R^{\lceil s \rceil \times R^{\lceil s \rceil}} \times \lceil np \rceil \qquad \lfloor gq \rfloor \ = \ R^{R^{\lfloor np \rfloor} \times R^{\lfloor s \rfloor \times R^{\lfloor s \rfloor}}}. \tag{3.21}$$

$$\lfloor\!\lfloor \mathsf{somebody} \rfloor\!\rfloor \ = \ \lambda\mathcal{Q}.(\exists\ \lambda x.(\mathcal{Q}\ \langle \lambda k.(k\ x), \lambda\langle c, p\rangle.(p\ c)\rangle)). \tag{3.22}$$

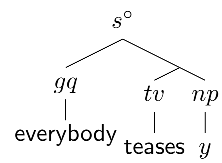
Ambiguous scope construal Compare 'Alice teases Lewis' with 'everybody teases somebody' involving two GQ expressions. Starting from $\boxed{1}$ $\mu\alpha.(\mathsf{teases} * ((x \ltimes \alpha) \rtimes y))$, the derivation can be completed in two ways, depending on whether we first bind the object variable y, then the subject variable x, or vice versa. On the left the subject wide scope reading, on the right the object wide scope reading.

1.


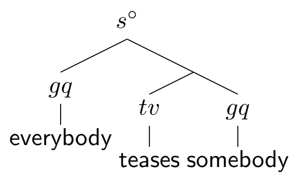
$\mu\alpha.(\mathsf{teases} * ((x \ltimes \alpha) \rtimes y))$

2a.


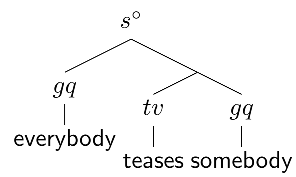
$\mu\beta.(\mathsf{somebody} * \mathtt{bind}(y, \boxed{1}, \beta))$

2b.


$\mu\gamma.(\mathsf{everybody} * \mathtt{bind}(x, \boxed{1}, \gamma))$

3a.


$\mu\gamma.(\mathsf{everybody} * \mathtt{bind}(x, \boxed{2a}, \gamma)))$

3b.


$\mu\beta.(\mathsf{somebody} * \mathtt{bind}(y, \boxed{2b}, \beta)))$

By applying the method described with previous examples, one obtains the $\forall/\exists$ reading for (3a) and the $\exists/\forall$ reading for (3b). First, the terms are interpreted by means of the definitions in (3.15) and (3.16) obtaining the following results:

$$\begin{aligned}\lfloor 3a \rfloor &= \lambda c.(\llbracket \mathsf{evrb.} \rrbracket\ \lambda\langle q, y\rangle.(\llbracket \mathsf{smb.} \rrbracket\ \lambda\langle p, z\rangle.(y\ \langle c, \lambda c'.(z\ \langle c', \lambda c''.(\llbracket \mathsf{teases} \rrbracket\ \langle p, \langle c'', q\rangle\rangle)\rangle)\rangle)));\\ \lfloor 3b \rfloor &= \lambda c.(\llbracket \mathsf{smb.} \rrbracket\ \lambda\langle p, z\rangle.(\llbracket \mathsf{evrb.} \rrbracket\ \lambda\langle q, y\rangle.(z\ \langle c, \lambda c'.(y\ \langle c', \lambda c''.(\llbracket \mathsf{teases} \rrbracket\ \langle p, \langle c'', q\rangle\rangle)\rangle)\rangle))).\end{aligned} \tag{3.23}$$

where the variables p, q of type C_{np} are for the object and subject respectively. The y, z variables are $s \oslash s$ continuations, the (primed) c are s continuations. Then, the $\llbracket \cdot \rrbracket$ translation is applied; the readings reduce to $\lambda c.(\forall\ \lambda x.(\exists\ \lambda y.(c\ ((\mathsf{teases}\ y)\ x))))$ and $\lambda c.(\exists\ \lambda y.(\forall\ \lambda x.(c\ ((\mathsf{teases}\ y)\ x))))$, respectively.

Local versus non-local scope Consider the two readings for the sentence 'Alice thinks somebody left'. The ambiguity arises here from the fact that in this context the GQ can non-deterministically select the embedded or the main clause s as its scope domain. We give the terms for these two derivations (local (a) versus non-local (b) scope) in (3.24), reusing the components $\boxed{1}$ and $\boxed{2}$ of (3.19), and the cbn interpretations of these terms in (3.25). These can be further reduced to (3.26) via a lexical recipe $\llbracket \mathsf{thinks} \rrbracket = \lambda\langle p, \langle c, q\rangle\rangle.(q\ \lambda x.(c\ ((\mathsf{thinks}\ (p\ \lambda c.c))\ x)))$ expressed in terms of a constant thinks of type $t \to (e \to t)$.

$$\begin{aligned}&a. \quad \mu\gamma.(\mathsf{thinks} * ((\mathsf{alice} \ltimes \gamma) \rtimes \boxed{2}));\\ &b. \quad \mu\gamma.(\mathsf{somebody} * \mathtt{bind}(x, \mu\gamma'.(\mathsf{thinks} * ((\mathsf{alice} \ltimes \gamma') \rtimes \boxed{1})), \gamma)).\end{aligned} \tag{3.24}$$

$$\begin{array}{ll}\lfloor a \rfloor = & \lambda c.(\llbracket \mathsf{thinks} \rrbracket\ \langle \lambda c'.(\llbracket \mathsf{somebody} \rrbracket\ \lambda\langle q, y\rangle.(y\ \langle c', \lambda c''.(\llbracket \mathsf{left} \rrbracket\ \langle c'', q\rangle)\rangle)), \langle c, \llbracket \mathsf{alice} \rrbracket\rangle\rangle); \\ \lfloor b \rfloor = & \lambda c.(\llbracket \mathsf{somebody} \rrbracket\ \lambda\langle q, y\rangle.(y\ \langle c, \lambda c'.(\llbracket \mathsf{thinks} \rrbracket\ \langle \lambda c''.(\llbracket \mathsf{left} \rrbracket\ \langle c'', q\rangle), \langle c', \llbracket \mathsf{alice} \rrbracket\rangle\rangle)\rangle)).\end{array} \tag{3.25}$$

$$\begin{array}{ll}\lfloor a \rfloor \rightsquigarrow_\beta & \lambda c.(c\ ((\mathsf{thinks}\ (\exists\ \mathsf{left}))\ \mathsf{alice})); \\ \lfloor b \rfloor \rightsquigarrow_\beta & \lambda c.(\exists\ \lambda y.(c\ ((\mathsf{thinks}\ (\mathsf{left}\ y))\ \mathsf{alice}))).\end{array} \tag{3.26}$$

3.6 Conclusions, Further Directions

In this paper we have moved from asymmetric Lambek calculus with a single succedent formula to the symmetric Lambek-Grishin calculus, where both the antecedent and the succedent are formula structures, configured in terms of $\otimes$ and $\oplus$ respectively, and where the $\otimes$ and $\oplus$ environments can interact in a structure-preserving way. This move makes it possible to import into the field of natural language semantics the powerful tools of $\lambda\mu$-calculus. The main attraction of the proposed continuation semantics, in our view, lies in the fact that **LG** allows us to fully exploit the duality between Lambek's directional $/, \backslash$ implications and the corresponding directional $\obslash, \oslash$ difference operations, at the level of syntax and at the level of semantics. We thus restore Curry's original idea of an *isomorphism* between proofs and terms, rather than the weaker homomorphic view of standard Lambek (or Montagovian) semantics.

Our approach differs in a number of respects from the related work cited in §3.1. Abstracting away from the directionality issue, de Groote's original application of $\lambda\mu$ calculus to scope construal syntactically types generalized quantifier phrases as *np* with meaning representation $\mu\alpha^{K_e}(\exists\ \alpha)$. As a result, a sentence with multiple GQ phrases is associated with a *unique* parse/term; the multiple readings for that term are obtained as a result of the non-confluence of $\lambda\mu$ calculus, which is considered as a feature, not a bug. Our approach in contrast is true to the principle that multiple readings can only arise as the images of distinct proofs, given the Curry-Howard isomorphism between proofs and terms. Barker [1, 2] uses a simplified continuation semantics, which lifts types A to $(A \rightarrow R) \rightarrow R$ 'externally', without applying the CPS transformation to the internal structure of complex types. This breaks the symmetry which is at the heart of our dual treatment of $/, \backslash$ vs $\obslash, \oslash$. The structural-rule account of scope flexibility in [3, 20] suffers from commutativity problems.

The approach described here, like Hendriks's type-shifting approach, creates all combinatorial possibilities for scope construal. However, it is well known that, depending on the choice of particular lexical items, many of these construals will in fact be unavailable. Bernardi [4] uses the control modalities $\diamondsuit, \Box$ to calibrate the scopal behaviour of particular classes of GQ expressions. Adding $\diamondsuit, \Box$ and a pair of dually residuated modalities to **LG** is straightforward. In a follow-up paper, we plan to study the continuation semantics of these operations, relating them to the `shift` and `reset` constructs one finds in the theory of functional programming languages and that have been considered in [2, 20].

Finally, the interpretation given here construes scopal ambiguities in a *static* setting. In a recent paper, de Groote [8] develops a continuation-based approach towards *dynamic* interpretation. A natural topic for further research would be to investigate how to incorporate this dynamic perspective in our setting, and how to extend the approach of [8] with the difference operations and the concomitant Grishin interaction principles.

3.7 Appendix: Shorthand Format for Sequent Derivations

As an example of the **LG** term assignment, (3.27) gives the derivation showing how one obtains a Lambek-style GQ type $C/(A\backslash B)$ from a $(B \oslash C) \obslash A$ source.

$$
\dfrac{\dfrac{\dfrac{\dfrac{C \xrightarrow{\alpha} \alpha : C \quad \dfrac{\dfrac{z : A \xrightarrow{z} A \quad B \xrightarrow{\gamma} \gamma : B}{z : A \circ A\backslash B \xrightarrow{z \ltimes \gamma} \gamma : B}\,(\backslash L)}{z : A \circ y : A\backslash B \xrightarrow{\mu\gamma.(y * (z \ltimes \gamma))} B}\,(\leftrightharpoons)}{z : A \circ y : A\backslash B \xrightarrow{(\mu\gamma.(y * (z \ltimes \gamma))) \succ \alpha} B \oslash C \circ \alpha : C}\,(\oslash R)}{A \circ y : A\backslash B \xrightarrow{\widetilde{\mu} z.((\mu\gamma.(y * (z \ltimes \gamma))) \succ \alpha) * \beta)} \beta : B \oslash C \circ \alpha : C}\,(\rightleftharpoons)}{(B \oslash C) \obslash A \circ y : A\backslash B \xrightarrow{\beta \obslash (\widetilde{\mu} z.((\mu\gamma.(y * (z \ltimes \gamma))) \succ \alpha) * \beta))} \alpha : C}\,(\obslash L)}{\dfrac{x : (B \oslash C) \obslash A \circ y : A\backslash B \xrightarrow{\mu\alpha.(x * (\beta \obslash (\widetilde{\mu} z.((\mu\gamma.(y * (z \ltimes \gamma))) \succ \alpha) * \beta)))} C}{x : (B \oslash C) \obslash A \xrightarrow{(\mu\alpha.(x * (\beta \obslash (\widetilde{\mu} z.((\mu\gamma.(y * (z \ltimes \gamma))) \succ \alpha) * \beta)))) \,/\, y} C/(A\backslash B)}\,(/R)}\,(\leftrightharpoons) \tag{3.27}
$$

This example shows that except for the most simple derivations, the sequent tree format soon becomes unwieldy. Below we introduce a more user-friendly line format, which graphically highlights the tree structure of the antecedent and succedent parts. In the line format, each row has (a) a line number, (b) the (co)term for the currently active formula, and the antecedent (c) and succedent (d) structures in tree format. The cursor singles out the currently active formula. It takes the form $\cdot^{\bullet}$ in the antecedent, and $\cdot^{\circ}$ in the succedent. With $\boxed{n}$ we refer to the (co)term at line n. Compare (3.27) with Figure 3.6, the derivation for a sentence 'Somebody left' in line format.

We reduce the length of a derivation further using the (qL) and (qR) rules of inference discussed in the main text and *folding*: a sequence of n $(/, \backslash R)$ (respectively $(\oslash, \obslash L)$) one-premise inferences is folded into a one-step one-premise inference; a $(\leftrightharpoons)$ step (or $(\rightleftharpoons)$ respectively) followed by a sequence of n $(/, \backslash L)$ (respectively $(\oslash, \obslash R)$) two-premise inferences is folded into a one-step $n+1$ premise inference; an n premise inference with m non-lexical axiom premises is contracted to an $n - m$ premise inference. Where the succedent (antecedent) tree is just a point, we write the highlighted formula as the root of the antecedent (succedent) tree. The result of applying these conventions for the example sentence 'somebody left' is (3.19) in the main text.

3.8 Bibliography

[1] C. Barker. Continuations and the nature of quantification. *Natural language semantics*, 2002.

[2] C. Barker. Continuations in natural language. In Hayo Thielecke, editor, *CW'04: Proceedings of the 4th ACM SIGPLAN continuations workshop*, pages 1–11. Tech. Rep. CSR-04-1, School of Computer Science, University of Birmingham, 2004.

[3] C. Barker and C. Shan. Types as graphs: Continuations in type logical grammar. *Journal of Logic, Language and Information*, 2005.

[4] R. Bernardi. *Reasoning with Polarity in Categorial Type Logic*. PhD thesis, Utrecht Institute of Linguistics OTS, 2002.

[5] P. Curien and H. Herbelin. Duality of computation. In *International Conference on Functional Programming (ICFP'00)*, pages 233–243, 2000. [2005: corrected version].

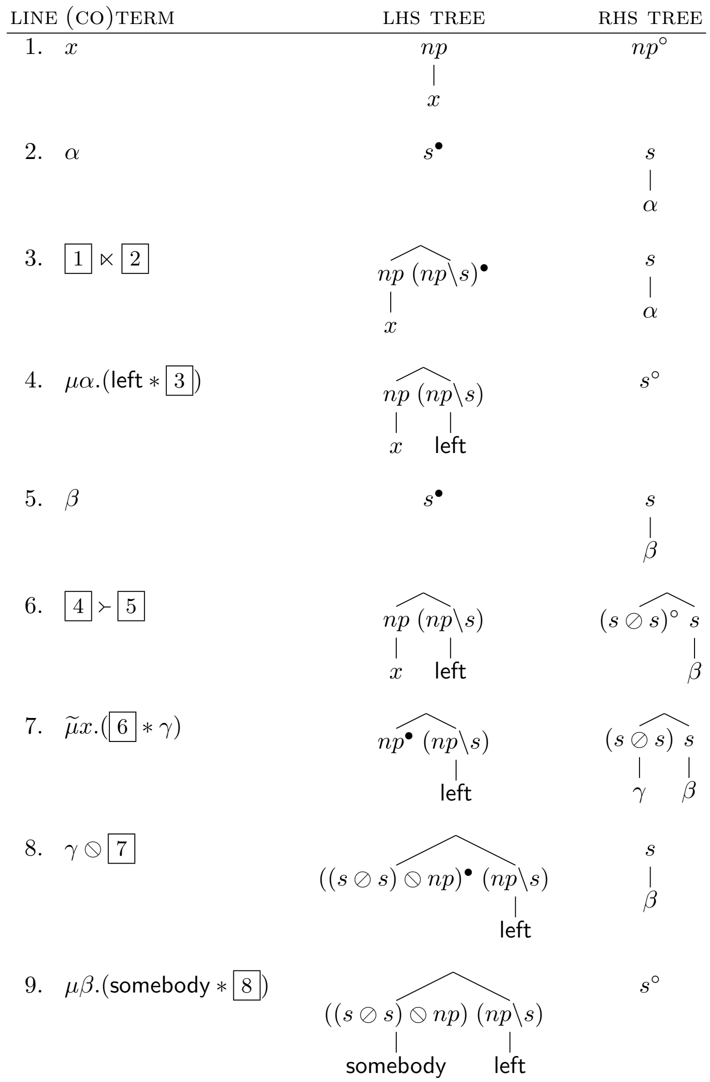


Figure 3.6: Tree-style derivation

[6] P. De Groote. Type raising, continuations, and classical logic. In M. Stokhof R. van Rooy, editor, *Proceedings of the Thirteenth Amsterdam Colloquium*, pages 97–101. ILLC, Universiteit van Amsterdam, 2001.

[7] P. De Groote and F. Lamarche. Classical non-associative Lambek calculus. *Studia Logica*, 71:335–388, 2002.

[8] Philippe de Groote. Towards a Montagovian account of dynamics. In *Proceedings SALT 16*. CLC Publications, 2006.

[9] Rajeev Goré. Substructural logics on display. *Logic Journal of IGPL*, 6(3):451–504, 1997.

[10] V.N. Grishin. On a generalization of the Ajdukiewicz-Lambek system. In A.I. Mikhailov, editor, *Studies in Nonclassical Logics and Formal Systems*, pages 315–334. Nauka, Moscow, 1983. [English translation in Abrusci and Casadio (eds.) Proceedings 5th Roma Workshop, Bulzoni Editore, Roma, 2002].

[11] H. Hendriks. *Studied flexibility. Categories and types in syntax and semantics.* PhD thesis, ILLC, Amsterdam University, 1993.

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

[13] J. Lambek. On the calculus of syntactic types. In R. Jakobson, editor, *Structure of Language and Its Mathematical Aspects*, pages 166–178. American Mathematical Society, 1961.

[14] J. Lambek. From categorial to bilinear logic. In K. Došen P. Schröder-Heister, editor, *Substructural Logics*, pages 207–237. Oxford University Press, 1993.

[15] M. Moortgat. Generalized quantifiers and discontinuous type constructors. In Harry Bunt and Arthur van Horck, editors, *Discontinuous Constituency*, pages 181–207. De Gruyter, 1996.

[16] Michael Moortgat. Symmetries in natural language syntax and semantics: the Lambek-Grishin calculus. In Daniel Leivant and Ruy de Queiros, editors, *Proceedings 14th Workshop on Logic, Language, Information and Computation (WoLLIC'07)*, LNCS 4576. Springer, 2007.

[17] G. Morrill. Discontinuity in categorial grammar. *Linguistics and Philosophy.*, 18:175–219, 1995.

[18] G. Morrill, M. Fadda, and Valentin. Nondeterministic discontinuous Lambek calculus. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS7)*, Tilburg, 2007.

[19] P. Selinger. Control categories and duality: on the categorical semantics of the lambda-mu calculus. *Math. Struct. in Comp. Science*, 11:207–260, 2001.

[20] C. Shan. *Linguistic side effects.* PhD thesis, Harvard University, 2005.

[21] P. Wadler. Call-by-value is dual to call-by-name. In *ICFP*, Uppsala, Sweden, August 2003.

4

Relational semantics for the Lambek-Grishin calculus

Natasha Kurtonina and Michael Moortgat

Abstract We study ternary relational semantics for symmetric versions of the Lambek calculus with interaction principles due to Grishin (1983). We obtain completeness on the basis of a Henkin-style weak filter construction.

Thanks We thank Anna Chernilovskaya and the anonymous MOL'07 referees for helpful comments on an earlier version of this paper.

4.1 Background, motivation

The categorial calculi proposed by Lambek and their current typelogical extensions respect an "intuitionistic" restriction: in a Gentzen presentation, Lambek sequents are of the form $\Gamma \Rightarrow B$, where B is a single formula, and Γ is a tree structure with formulas $A_1, \ldots, A_n$ at the yield. Depending on the particular calculus one works with, the antecedent structure can degenerate into a list or a multiset of formulas. The intuitionistic restriction is a serious expressive limitation when it comes to using the Lambek framework in the analysis of natural language syntax and semantics. Core phenomena such as displacement or scope construal are beyond the reach of the basic Lambek calculus; to deal with such phenomena, various extensions have been proposed based on structural rules, which can be introduced implicitly or explicitly, and with global or modally-controlled application regimes. The price one pays for such extensions is high: whereas the basic Lambek calculus has a polynomial recognition problem [3], already the simplest extension with an associative regime for $\otimes$ is known to be NP complete as shown in [8].

In a remarkable paper written in 1983, V.N. Grishin [4] has proposed a different strategy for generalizing the Lambek calculi. The starting point for Grishin's approach is a symmetric extension of the Lambek calculus: in addition to the familiar operators $\otimes, \backslash, /$ (product, left and right division), one also considers a dual family $\oplus, \oslash, \obslash$: coproduct, right and left difference.[1] The resulting vocabulary is given in (4.1).

[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'. We follow [6] in using the notation $\oplus$ for the coproduct, which is a multiplicative operation.

$$\begin{array}{lll} A, B \quad ::= & p \mid & \text{atoms: } s \text{ sentence, } np \text{ noun phrases, } \ldots \\ & A \otimes B \mid B \backslash A \mid A / B \mid & \text{product, left vs right division} \\ & A \oplus B \mid A \oslash B \mid B \obslash A & \text{coproduct, right vs left difference} \end{array} \tag{4.1}$$

Algebraically, the Lambek operators form a residuated triple; likewise, the $\oplus$ family forms a dual residuated triple. The *minimal* symmetric categorial grammar, which we will refer to as $\mathbf{LG}_\emptyset$, consists of just the preorder axioms of (4.2), i.e. reflexivity and transitivity of the derivability relation, together with the (dual) residuation principles given in (4.3). This minimal system preserves the polynomiality of the asymmetric **NL** as shown in [2].

$$\text{(refl)} \quad A \vdash A; \qquad \text{from } A \vdash B \text{ and } B \vdash C \text{ infer } A \vdash C \quad \text{(trans)} \tag{4.2}$$

$$\begin{array}{ll} \text{(rp)} & A \vdash C/B \quad \text{iff} \quad A \otimes B \vdash C \quad \text{iff} \quad B \vdash A \backslash C \\ \text{(drp)} & B \obslash C \vdash A \quad \text{iff} \quad C \vdash B \oplus A \quad \text{iff} \quad C \oslash A \vdash B \end{array} \tag{4.3}$$

The minimal symmetric system $\mathbf{LG}_\emptyset$ doesn't have the required expressivity to address the linguistic problems mentioned in the introduction. For every theorem of the (non-associative) Lambek calculus, $\mathbf{LG}_\emptyset$ also has its image under arrow reversal. Interaction between the $\otimes$ and the $\oplus$ family, however, is limited to glueing together theorems of the two families with the transitivity rule.

What makes Grishin's work attractive from the perspective of categorial grammar, is the systematic theory he presents for extending $\mathbf{LG}_\emptyset$ with extra axioms. In section 2.7 of his paper, Grishin presents sixteen options for extending $\mathbf{LG}_\emptyset$. Eight of these represent the familiar associativity and/or commutativity postulates for $\otimes$ and symmetrically $\oplus$. Since these choices destroy sensitivity for word order and/or constituent structure, we will ignore them. The remaining eight options are principles of *interaction* relating connectives from the $\otimes$ and the $\oplus$ family. They naturally cluster in two groups of four, which we will refer to as $\mathcal{G}^\uparrow$ and $\mathcal{G}^\downarrow$.

Consider first the group $\mathcal{G}^\uparrow$ (the Class IV postulates, in Grishin's own terminology) which consists of the principles in (4.4). $G1$ and $G3$ have been called mixed associativity principles, $G2$ and $G4$ mixed commutativity principles. We think the use of the concepts "associativity" and "commutativity" is misleading here: as we will see below, the $\otimes$ and $\oplus$ families have individual interpreting relations of fusion and fission respectively. We prefer to refer to $G1$–$G4$ as (weak) distributivity principles.

$$\begin{array}{llll} (G1) & (A \obslash B) \otimes C \vdash A \obslash (B \otimes C) & C \otimes (B \oslash A) \vdash (C \otimes B) \oslash A & (G3) \\ (G2) & C \otimes (A \obslash B) \vdash A \obslash (C \otimes B) & (B \oslash A) \otimes C \vdash (B \otimes C) \oslash A & (G4) \end{array} \tag{4.4}$$

Intuitively, the interacion principles in (4.4) deal with the situation where a difference operation ($\obslash$ or $\oslash$) is *trapped* in a $\otimes$ context where they are inaccessible for logical manipulation. Consider first $G1$ and $G2$. On the lefthand side of the turnstile, a formula $A \obslash B$ occurs as the first or second coordinate of a product. The postulates invert the dominance relation between $\otimes$ and $\obslash$, raising the subformula A to a position where it can be shifted to the righthand side by means of the dual residuation principles of (4.3). $G3$ and $G4$ are the images of $G1$ and $G2$ under left-right symmetry.

Interaction principles *dual* to those in (4.4) are given in (4.5): they deal with the situation where a left or right implication is trapped within a $\oplus$ context, this time raising the A subformula to the position where it can be shifted to the lefthand side by means of the residuation principles

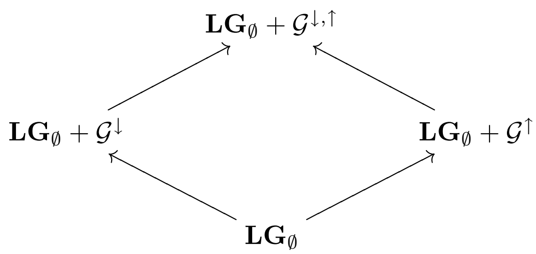


Figure 4.1: The Lambek-Grishin landscape

of (4.3). We leave it to the reader to check that the forms Gn' in (4.5) are indeed derivable from the respective Gn in (4.4) and (4.2)-(4.3).

$$\begin{array}{llll} (G1') & (C \oplus B)/A \vdash C \oplus (B/A) & A\backslash(B \oplus C) \vdash (A\backslash B) \oplus C & (G3') \\ (G2') & (B \oplus C)/A \vdash (B/A) \oplus C & A\backslash(C \oplus B) \vdash C \oplus (A\backslash B) & (G4') \end{array} \tag{4.5}$$

Consider next the group $\mathcal{G}^{\downarrow}$ (the Class I postulates in Grishin's classification), consisting of the interaction principles in (4.6) — the *converses* of the principles in group $\mathcal{G}^{\uparrow}$. $\mathcal{G}^{\uparrow}$ and $\mathcal{G}^{\downarrow}$ represent two independent options to extend $\mathbf{LG}_\emptyset$ with interaction principles. The general picture that emerges then is the landscape of Figure 4.1 where the minimal symmetric Lambek calculus is extended either with $G1$–$G4$ or with their converses, or with the combination of the two.

$$\begin{array}{llll} (G1^{-1}) & (A \obslash B) \otimes C \dashv A \obslash (B \otimes C) & C \otimes (B \oslash A) \dashv (C \otimes B) \oslash A & (G3^{-1}) \\ (G2^{-1}) & C \otimes (A \obslash B) \dashv A \obslash (C \otimes B) & (B \oslash A) \otimes C \dashv (B \otimes C) \oslash A & (G4^{-1}) \end{array} \tag{4.6}$$

Linguistic applications The choice we are making for the extension of $\mathbf{LG}_\emptyset$ is motivated by the linguistic application: we consider the full set of *structure-preserving* interaction principles, while rejecting same-sort associativity and/or commutativity options. Grishin's own paper opts for an associative regime, with both the same-sort associativity for $\otimes$ and $\oplus$, and the mixed associativity of the $\mathcal{G}^{\downarrow}$ group, i.e. $G1^{-1}$ and $G3^{-1}$ of (4.6). In the first thorough exposition of Grishin's work before it was translated in English, Lambek [6] also adopts the associative regime, but explores the mixed associativities of both the $\mathcal{G}^{\downarrow}$ and the $\mathcal{G}^{\uparrow}$ groups.

We give two simple illustrations of the potential of the Lambek-Grishin systems in Figure 4.1 to address the problems with asymmetric Lambek calculi mentioned in the introduction. The first is an example of non-local scope construal; the second is a case of non-peripheral extraction. In both cases, we start from a lexical type assignment from which the usual Lambek type is derivable, cf. the assignments in (4.7). What this means is that whatever could be done with the Lambek types can still be done; but thanks to the Grishin interaction principles, we will be able to do more.

$$\begin{array}{lrcl} \text{someone} & (s \oslash s) \obslash np & \vdash & s/(np\backslash s) \\ \text{which} & (n\backslash n)/((s \oslash s) \oplus (s/np)) & \vdash & (n\backslash n)/(s/np) \end{array} \tag{4.7}$$

In (4.8) one finds one of the two derivations for a sentence of the type 'Alice thinks someone left'. Whereas the Lambek assignment $s/(np\backslash s)$ is restricted to local construal in the embedded clause, the assignment $(s \oslash s) \obslash np$ also allows construal at the main clause level. It is this non-local construal that is represented by (4.8). By means of the interaction principle $G2$, the $(s \oslash s)$ subformula raises 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. Notice that with only the mixed associative interactions, the $\obslash$ connective would be trapped in the $\otimes$ context, and the derivation would fail. The reader is invited to consult [1] for a detailed analysis of scope construal along these lines. Semantically, the analysis of [1] is based on a Curry-Howard interpretation of Lambek-Grishin derivations in the continuation-passing style; this interpretation associates (4.8) with the reading $(\exists\ \lambda x.((\textsf{thinks}\ (\textsf{left}\ x))\ \textsf{a}))$ as required.

$$\dfrac{\dfrac{\dfrac{np \otimes (((np\backslash s)/s) \otimes (np \otimes (np\backslash s))) \vdash s \quad s \vdash (s \oslash s) \oplus s}{np \otimes (((np\backslash s)/s) \otimes (np \otimes (np\backslash s))) \vdash (s \oslash s) \oplus s}\,\text{trans}}{(s \oslash s) \obslash (np \otimes (((np\backslash s)/s) \otimes (np \otimes (np\backslash s)))) \vdash s}\,\text{drp}}{np \otimes (((np\backslash s)/s) \otimes (\underbrace{((s \oslash s) \obslash np)}_{\textsf{someone}} \otimes (np\backslash s))) \vdash s}\,G2 \tag{4.8}$$

The second example is for displacement as in '(movie which) John saw on TV'. In the derivation (4.9) we make use of the combined $\mathcal{G}^{\downarrow,\uparrow}$ principles, i.e. the principles (4.4) and their converses (4.6). We abbriviate $(np\backslash s)/np$ as tv (transitive verb) and $(np\backslash s)\backslash(np\backslash s)$ as adv (adverb). 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.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{np \otimes ((tv \otimes np) \otimes adv) \vdash s \quad s \vdash (s \oslash s) \oplus s}{np \otimes ((tv \otimes np) \otimes adv) \vdash (s \oslash s) \oplus s}\,\text{trans}}{tv \vdash ((np\backslash(s \oslash s))/adv) \oplus (s/np)}\,Gn,\text{rp}}{np \otimes ((tv \oslash (s/np)) \otimes adv) \vdash s \oslash s}\,\text{rp,drp}}{(np \otimes (tv \otimes adv)) \oslash (s/np) \vdash s \oslash s}\,Gn^{-1}}{np \otimes (tv \otimes adv) \vdash (s \oslash s) \oplus (s/np)}\,\text{drp} \tag{4.9}$$

An attractive property of the Lambek-Grishin systems in Figure 4.1 is that the expressivity resides entirely in the interaction principles: the composition operation $\otimes$ in itself (and the dual $\oplus$) allows no structural rules at all, which means that the resulting notion of wellformedness is fully sensitive to linear order and constituent structure of the grammatical material. It is shown in [7] that the relation of type similarity of $\mathbf{LG}_\emptyset + \mathcal{G}^\uparrow$ is as strong as similarity in (associative, commutative) **LP**: $A \sim B$ iff the images of A and B in a free Abelian group interpretation are equal. In **LP** one obtains this notion of $\sim$ by sacrificing order and constituent sensitivity; in the Lambek-Grishin setting, the same notion of similarity is obtained in a structure-preserving way.

4.2 Relational semantics

Let us turn now to the frame semantics for **LG**. In (4.10) and (4.11) we compare the truth conditions for the fusion and fission operations. 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 slashes are the corresponding universal modalities for the rotations of $R_\otimes$. For fission $\oplus$ 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 $\mathbf{LG}_\emptyset$, $R_\oplus$ and $R_\otimes$ are *distinct* accessibility relations. Frame constraints corresponding to the Grishin interaction postulates of the group $\mathcal{G}^\uparrow$ or $\mathcal{G}^\downarrow$ will determine how their interpretation is related.

$$\begin{array}{l} x \Vdash A \otimes B \text{ iff } \exists yz.R_\otimes xyz \text{ and } y \Vdash A \text{ and } z \Vdash B \\ y \Vdash C/B \text{ iff } \forall xz.(R_\otimes xyz \text{ and } z \Vdash B) \text{ implies } x \Vdash C \\ z \Vdash A\backslash C \text{ iff } \forall xy.(R_\otimes xyz \text{ and } y \Vdash A) \text{ implies } x \Vdash C \end{array} \tag{4.10}$$

$$\begin{array}{l} x \Vdash A \oplus B \text{ iff } \forall yz.R_\oplus xyz \text{ implies } (y \Vdash A \text{ or } z \Vdash B) \\ y \Vdash C \oslash B \text{ iff } \exists xz.R_\oplus xyz \text{ and } z \nVdash B \text{ and } x \Vdash C \\ z \Vdash A \obslash C \text{ iff } \exists xy.R_\oplus xyz \text{ and } y \nVdash A \text{ and } x \Vdash C \end{array} \tag{4.11}$$

Henkin construction To establish completeness, we use a Henkin construction. In the Henkin setting, “worlds” are (weak) *filters*: sets of formulas closed under $\vdash$. Let $\mathcal{F}$ be the formula language of (4.1). Let $\mathcal{F}_\vdash = \{X \in \mathcal{P}(\mathcal{F}) \mid (\forall A \in X)(\forall B \in \mathcal{F})\ A \vdash B \text{ implies } B \in X\}$. The set of filters $\mathcal{F}_\vdash$ is closed under the operations $(\cdot \widehat{\otimes} \cdot)$, $(\cdot \widehat{\obslash} \cdot)$ defined in (4.12) below. It is easy to show that $X \widehat{\otimes} Y$ and $X \widehat{\obslash} Y$ are indeed members of $\mathcal{F}_\vdash$.

$$\begin{array}{rcll} X \widehat{\otimes} Y & = & \{C \mid \exists A, B\ (A \in X \text{ and } B \in Y \text{ and } A \otimes B \vdash C)\} & \\ X \widehat{\obslash} Y & = & \{B \mid \exists A, C\ (A \notin X \text{ and } C \in Y \text{ and } A \obslash C \vdash B\}, & \text{alternatively} \\ X \widehat{\obslash} Y & = & \{B \mid \exists A, C\ (A \notin X \text{ and } C \in Y \text{ and } C \vdash A \oplus B\} & \end{array} \tag{4.12}$$

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

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

PROOF $(\dagger)(\subseteq)$ Suppose $C \in \lfloor A \otimes B \rfloor$, i.e. $A \otimes B \vdash C$. With $A' := A$ and $B' := B$ we claim $\exists A', B'$ such that $A \vdash A'$, $B \vdash B'$ and $A' \otimes B' \vdash C$, which by (Def $\widehat{\otimes}$) means that $C \in \lfloor A \rfloor \widehat{\otimes} \lfloor B \rfloor$ as desired. For the $(\supseteq)$ direction, we will prove the following lemma:

Lemma 1. $A \otimes B \in X$ implies $\lfloor A \rfloor \widehat{\otimes} \lfloor B \rfloor \subseteq X$.

Since $A \otimes B \in \lfloor A \otimes B \rfloor$ by definition, we then have $\lfloor A \rfloor \widehat{\otimes} \lfloor B \rfloor \subseteq \lfloor A \otimes B \rfloor$.

PROOF OF LEMMA 1. Suppose $C \in \lfloor A \rfloor \widehat{\otimes} \lfloor B \rfloor$, i.e. $\exists A', B'$ such that $A' \in \lfloor A \rfloor$ i.e. $A \vdash A'$, $B' \in \lfloor B \rfloor$ i.e. $B \vdash B'$, and $A' \otimes B' \vdash C$. By Monotonicity, $A \otimes B \vdash A' \otimes B'$. By Transitivity, $A \otimes B \vdash C$. Together with $A \otimes B \in X$ this implies $C \in X$ as desired.

The $(\ddagger)$ case is entirely similar. $(\ddagger)(\subseteq)$ Suppose $B \in \lfloor A \obslash C \rfloor$, i.e. $A \obslash C \vdash B$. With $A' := A$ and $C' := C$ we claim $\exists A', C'$ such that $A' \vdash A$, $C \vdash C'$ and $A' \obslash C' \vdash B$, which by (Def $\widehat{\obslash}$) means that $B \in \lceil A \rceil^\sim \widehat{\obslash} \lfloor C \rfloor$ as desired. For the $(\supseteq)$ direction, we show that the folloing holds:

Lemma 2. $A \obslash C \in X$ implies $\lceil A \rceil^\sim \widehat{\obslash} \lfloor C \rfloor \subseteq X$.

Since $A \obslash C \in \lfloor A \obslash C \rfloor$ by definition, we then have $\lceil A \rceil^\sim \widehat{\obslash} \lfloor C \rfloor \subseteq \lfloor A \obslash C \rfloor$.

PROOF OF LEMMA 2. Suppose $B \in \lceil A \rceil^\sim \widehat{\obslash} \lfloor C \rfloor$, i.e. $\exists A', C'$ such that $A' \notin \lceil A \rceil^\sim$ i.e. $A' \vdash A$, $C' \in \lfloor C \rfloor$ i.e. $C \vdash C'$, and $A' \obslash C' \vdash B$. By Monotonicity, $A \obslash C \vdash A' \obslash C'$. By Transitivity, $A \obslash C \vdash B$. Together with $A \obslash C \in X$ this implies $B \in X$ as desired.

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

$W^c = \mathcal{F}_\vdash$

$R^c_\otimes XYZ$ iff $Y \mathbin{\widehat{\otimes}} Z \subseteq X$

$R^c_\oplus XYZ$ iff $Y \mathbin{\widehat{\oslash}} X \subseteq Z$

$V^c(p) = \{X \in W^c \mid p \in X\}$

Truth lemma We want to show for any formula $A \in \mathcal{F}$ and filter $X \in \mathcal{F}_\vdash$ that $X \Vdash A$ iff $A \in X$. The proof is by induction on the complexity of A. The base case is handled by V^c. Let us look first at the connectives $\oplus, \obslash, \oslash$.

Coproduct $X \Vdash A \oplus B$ iff $A \oplus B \in X$

($\Rightarrow$) Suppose $X \Vdash A \oplus B$. We have to show that $A \oplus B \in X$. By (Def $\oplus$) we have that $\forall Y, Z$ ($Y \mathbin{\widehat{\obslash}} X \subseteq Z$ and $Y \nVdash A$) implies $Z \Vdash B$. Setting $Y := \lceil A \rceil^\sim$ (therefore, $A \notin Y$ and, by IH for Y, $Y \nVdash A$) and $Z := Y \mathbin{\widehat{\obslash}} X$, the antecedent holds, implying $Z \Vdash B$. By IH and the choice of Z we then have $B \in Z$ and $B \in \lceil A \rceil^\sim \mathbin{\widehat{\obslash}} X$. By (Def $\widehat{\obslash}$) $B \in \lceil A \rceil^\sim \mathbin{\widehat{\obslash}} X$ means $\exists A_1, A_2$ such that $A_1 \notin \lceil A \rceil^\sim$, $A_2 \in X$ and $A_2 \vdash A_1 \oplus B$. $A_1 \notin \lceil A \rceil^\sim$ means $A_1 \vdash A$, hence from $A_2 \vdash A_1 \oplus B$ we get $A_2 \vdash A \oplus B$ by Transitivity. Since X is a filter, from $A_2 \in X$ and $A_2 \vdash A \oplus B$ we obtain $A \oplus B \in X$ as desired.

($\Leftarrow$) Suppose $A \oplus B \in X$. We have to show that $X \Vdash A \oplus B$, i.e. $\forall Y, Z$ ($R^c_\oplus XYZ$ and $Y \nVdash A$) implies $z \Vdash B$. Assume $R^c_\oplus XYZ$ and $Y \nVdash A$. We have to show $Z \Vdash B$. Using IH and the facts we already have ($R^c_\oplus XYZ$ and $A \notin Y$ and $A \oplus B \in X$) we conclude that $A \obslash (A \oplus B) \in Z$. But $A \obslash (A \oplus B) \vdash B$, so $B \in Z$ and by IH $Z \Vdash B$. This is what was needed to show.

Left difference $X \Vdash A \obslash B$ iff $A \obslash B \in X$

($\Rightarrow$) Suppose $X \Vdash A \obslash B$. We have to show that $A \obslash B \in X$. $X \Vdash A \obslash B$ means $\exists Y, Z$ such that $R^c_\oplus ZYX$, i.e. $Y \mathbin{\widehat{\obslash}} Z \subseteq X$, and $Y \nVdash A$ and $Z \Vdash B$. By IH $A \notin Y$ and $B \in Z$. Since also $A \obslash B \vdash A \obslash B$, from (Def $\widehat{\obslash}$) we conclude that $A \obslash B \in Y \mathbin{\widehat{\obslash}} Z$ and therefore $A \obslash B \in X$ as desired.

($\Leftarrow$) Suppose $A \obslash B \in X$. We have to show that $X \Vdash A \obslash B$. It was shown in Lemma 2 that $A \obslash B \in X$ implies $\lceil A \rceil^\sim \mathbin{\widehat{\obslash}} \lfloor B \rfloor \subseteq X$, which means we have $R^c_\oplus \lfloor B \rfloor \lceil A \rceil^\sim X$. Since $A \notin \lceil A \rceil^\sim$ and $B \in \lfloor B \rfloor$, by IH we claim $\exists Y, Z$ such that $R^c_\oplus ZYX$ and $Y \nVdash A$ and $Z \Vdash B$, which means $X \Vdash A \obslash B$ as desired.

Right difference $X \Vdash B \oslash A$ iff $B \oslash A \in X$

($\Rightarrow$) Suppose $X \Vdash B \oslash A$, i.e. $\exists Y, Z$ such that $X \mathbin{\widehat{\obslash}} Z \subseteq Y$ (Def $R^c_\oplus$) and $Y \nVdash A$ and $Z \Vdash B$, i.e. by IH $B \in Z$. To show that $B \oslash A \in X$, we reason by contradiction and assume $B \oslash A \notin X$. From this assumption and $B \in Z$ we have $(B \oslash A) \obslash B \in X \mathbin{\widehat{\obslash}} Z$ by (Def $\widehat{\obslash}$). Since $(B \oslash A) \obslash B \vdash A$, $A \in X \mathbin{\widehat{\obslash}} Z$, so also $A \in Y$. Contradiction with $Y \nVdash A$, hence the assumption $B \oslash A \notin X$ doesn't hold, as required.

($\Leftarrow$) Suppose $B \oslash A \in X$. To show that $X \Vdash B \oslash A$ we proceed by contraposition and assume $X \nVdash B \oslash A$, i.e. $\forall Y, Z$ ($R^c_\oplus ZXY$ and $Y \nVdash A$) implies $Z \nVdash B$, alternatively ($X \mathbin{\widehat{\obslash}} Z \subseteq Y$ and $Z \Vdash B$) implies $Y \Vdash A$. Setting $Y := X \mathbin{\widehat{\obslash}} Z$ and $Z := \lfloor B \rfloor$, the antecedent holds, hence $X \mathbin{\widehat{\obslash}} Z \Vdash A$ and by IH $A \in X \mathbin{\widehat{\obslash}} Z$. By (Def $\widehat{\obslash}$) this means $\exists A_1, A_2$ such that $A_1 \notin X$, $A_2 \in \lfloor B \rfloor$ and $A_2 \vdash A_1 \oplus A$. ¿From $A_2 \in \lfloor B \rfloor$ we have $B \vdash A_2$, so by Transitivity, $B \vdash> A_1 \oplus A$, and by Dual residuation, $B \oslash A \vdash A_1$. Since $A_1 \notin X$, $B \oslash A \notin X$, contradicting our original assumption.

For the $\otimes, /, \backslash$ connectives, we refer to [5] (Theorem 3.3.2, p 75), repeated here for convenience.

Product $\quad X \Vdash A \otimes B$ iff $A \otimes B \in X$

($\Rightarrow$) Suppose $X \Vdash A \otimes B$, i.e. $\exists Y, Z$ such that $Y \mathbin{\widehat{\otimes}} Z \subseteq X$, $Y \Vdash A$ and $Z \Vdash B$. By IH, $A \in Y$ and $B \in Z$. Since $A \otimes B \vdash A \otimes B$, by (Def $\widehat{\otimes}$) we have $A \otimes B \in X$ as desired.

($\Leftarrow$) Suppose $A \otimes B \in X$. In Lemma 1 we have shown that this implies $\lfloor A \rfloor \mathbin{\widehat{\otimes}} \lfloor B \rfloor \subseteq X$, i.e. $R^c_\otimes X \lfloor A \rfloor \lfloor B \rfloor$ by (Def $R^c_\otimes$). Since $A \in \lfloor A \rfloor$, $B \in \lfloor B \rfloor$, by IH we have $\lfloor A \rfloor \Vdash A$, $\lfloor B \rfloor \Vdash B$. By the truth condition for $\otimes$ this means $X \Vdash A \otimes B$ as desired.

Right division $\quad$ We do $X \Vdash A \backslash B$ iff $A \backslash B \in X$. The / case is symmetric.

($\Rightarrow$) Suppose $X \Vdash A \backslash B$, i.e. $\forall Y, Z$ if $R^c_\otimes ZYX$ and $Y \Vdash A$ then $Z \Vdash B$. Putting $Y := \lfloor A \rfloor$ and $Z := \lfloor A \rfloor \mathbin{\widehat{\otimes}} X$, since $\lfloor A \rfloor \mathbin{\widehat{\otimes}} X \subseteq \lfloor A \rfloor \mathbin{\widehat{\otimes}} X$ we have $R^c_\otimes ZYX$ by (Def $R^c_\otimes$), and since $A \in \lfloor A \rfloor$ also $\lfloor A \rfloor \Vdash A$ by IH, hence $\lfloor A \rfloor \mathbin{\widehat{\otimes}} X \Vdash B$, and by IH $B \in \lfloor A \rfloor \mathbin{\widehat{\otimes}} X$. By (Def $\widehat{\otimes}$) this means $\exists C, D$ such that $C \in \lfloor A \rfloor$ i.e. $A \vdash C$, $D \in X$ and $C \otimes D \vdash B$. By Transitivity, $A \otimes D \vdash B$ and by Residuation, $D \vdash A \backslash B$. Hence $A \backslash B \in X$ as desired.

($\Leftarrow$) Suppose $A \backslash B \in X$. We have to show that $X \Vdash A \backslash B$, i.e. $\forall Y, Z$ if $R^c_\otimes ZYX$ and $Y \Vdash A$ then $Z \Vdash B$. Suppose the antecedent holds, which means $Y \mathbin{\widehat{\otimes}} X \subseteq Z$ by (Def $R^c_\otimes$) and $A \in Y$ by IH. Together with $A \backslash B \in X$ we have $A \otimes (A \backslash B) \in Z$ by (Def $\widehat{\otimes}$). Since $A \otimes (A \backslash B) \vdash B$, also $B \in Z$. By IH $Z \Vdash B$ which means the consequent of the truth condition for $\backslash$ holds, hence $X \Vdash A \backslash B$ as desired.

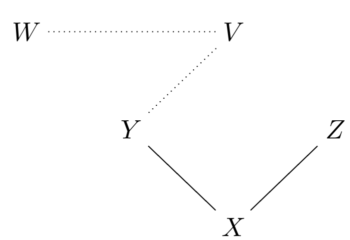
This establishes the Truth Lemma, from which completeness immediately follows.

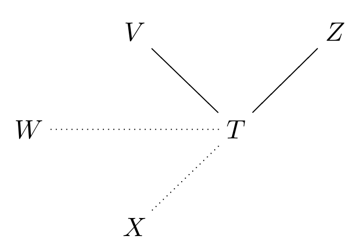
Theorem $\quad$ Completeness of $\mathbf{LG}_\emptyset$. If $\models A \vdash B$, then $A \vdash B$ is provable in $\mathbf{LG}_\emptyset$.

PROOF Suppose $A \vdash B$ is not provable. Then, by the Truth Lemma, $\mathcal{M}^c, A \nVdash B$. Since $\mathcal{M}^c, A \Vdash A$, we have $\mathcal{M}^c \not\models A \vdash B$, and hence $\not\models A \vdash B$.

Completeness of extensions with $\mathcal{G}^\uparrow$ and/or $\mathcal{G}^\downarrow$ $\quad$ In the minimal symmetric system, the $R_\otimes$ and $R_\oplus$ accessibility relations are distinct. For the extensions with Grishin interaction principles, we have frame constraints relating the interpretation of $R_\otimes$ and $R_\oplus$. Consider first the group $\mathcal{G}^\uparrow$. We take $(G1)$ as a representative: $(A \oslash B) \otimes C \vdash A \oslash (B \otimes C)$. The other axioms in the group are dealt with analogously. For $(G1)$ we have the constraint in (4.13) (where $R^{(-2)}xyz = Rzyx$).

$$\forall xyzwv \;\; (R_\otimes xyz \;\wedge\; R^{(-2)}_\oplus ywv) \Rightarrow \exists t \; (R^{(-2)}_\oplus xwt \;\wedge\; R_\otimes tvz) \tag{4.13}$$

(†)


(‡)


In (†) we depict $X \Vdash (A \oslash B) \otimes C$, with $W \nVdash A$, $V \Vdash B$ and $Z \Vdash C$; in (‡) $X \Vdash A \oslash (B \otimes C)$. Dotted lines represent $R^c_\oplus$, solid lines $R^c_\otimes$.

We have to show that in the Henkin model $\forall X, Y, Z, V, W$ construed as in (†), there is a fresh internal T connecting the root X to the leaves W, V, Z as in (‡). The solution $T := V \mathbin{\widehat{\otimes}} Z$ gives us $R^c_\otimes TVZ$ since $V \mathbin{\widehat{\otimes}} Z \subseteq V \mathbin{\widehat{\otimes}} Z$. To also show $R^c_\oplus TWX$, i.e. $W \mathbin{\widehat{\oslash}} T \subseteq X$, suppose $A' \in W \mathbin{\widehat{\oslash}} T$.

We need to show that $A' \in X$. By (Def $\widehat{\oslash}$) $A' \in W \widehat{\oslash} T$ means $\exists A_1, A_2$ such that $A_1 \notin W$, $A_2 \in T$ and $A_1 \oslash A_2 \vdash A'$. Since $T := V \widehat{\otimes} Z$, $A_2 \in T$ means $\exists B_1, B_2$ such that $B_1 \in V$, $B_2 \in Z$ and $B_1 \otimes B_2 \vdash A_2$. Taking the configuration (†) together with $A_1 \notin W$ and $B_1 \in V$, we conclude $Y \Vdash A_1 \oslash B_1$ which in (†) together with $B_2 \in Z$ implies that $X \Vdash (A_1 \oslash B_1) \otimes B_2$. By the Truth Lemma, this means that $(A_1 \oslash B_1) \otimes B_2 \in X$ and since X is a filter and $(Gl1)$ an axiom, $A_1 \oslash (B_1 \otimes B_2) \in X$. But since $B_1 \otimes B_2 \vdash A_2$ we conclude that $A_1 \oslash A_2 \in X$. Together with $A_1 \oslash A_2 \vdash A'$, since X is a filter, we obtain $A' \in X$ as desired.

Consider next the group of interaction principles $\mathcal{G}^{\downarrow}$, the converses of $\mathcal{G}^{\uparrow}$. As a representative, we take $(Gl1)^{-1}$: $(A \oslash B) \otimes C \dashv A \oslash (B \otimes C)$.

This time, we have to show that in the Henkin model $\forall X, T, Z, V, W$ construed as in (‡), there is a fresh internal Y connecting the root X to the leaves W, V, Z as in (†). Let $Y := W \widehat{\oslash} V$. Since $W \widehat{\oslash} V \subseteq W \widehat{\oslash} V$, $R^c_{\oplus} VWY$ holds. To show that also $R^c_{\otimes} XYZ$, i.e. $Y \widehat{\otimes} Z \subseteq X$, suppose $A' \in Y \widehat{\otimes} Z$, and let us show that $A' \in X$. By (Def $\widehat{\otimes}$), $A' \in Y \widehat{\otimes} Z$ means $\exists A_2 B_1$ such that $A_2 \in Y$, $B_1 \in Z$ and $A_2 \otimes B_1 \vdash A'$. Since we had $Y := W \widehat{\oslash} V$, $A_2 \in Y$ by (Def $\widehat{\oslash}$) means $\exists A_3 C_1$ such that $A_3 \notin W$, $C_1 \in V$ and $A_3 \oslash C_1 \vdash A_2$. Given that $C_1 \in V$ and $B_1 \in Z$, in the configuration (‡) we have $T \Vdash C_1 \otimes B_1$, and since $A_3 \notin W$, $X \Vdash A_3 \oslash (C_1 \otimes B_1)$. By the Truth Lemma this means that $A_3 \oslash (C_1 \otimes B_1) \in X$, and also $(A_3 \oslash C_1) \otimes B_1 \in X$, since X is a filter and we have $(Gl1)^{-1}$. Since $A_3 \oslash C_1 \vdash A_2$, we can conclude $A_2 \otimes B_1 \in X$, and since $A_2 \otimes B_1 \vdash A'$, also $A' \in X$ as desired.

4.3 Concluding remarks

We have established completeness for the minimal symmetric Lambek calculus $\mathbf{LG}_{\emptyset}$ and for its extension with interaction principles. The construction is neutral with respect to the choice between $\mathcal{G}^{\uparrow}$ and $\mathcal{G}^{\downarrow}$: it accommodates $G1$–$G4$ and the converses $G1^{-1}$–$G4^{-1}$ in an entirely similar way. In further research, we would like to consider more concrete models with a bias towards either $G1$–$G4$ or the converse principles, and to relate these models to the distinction between 'overt' and 'covert' forms of displacement, as illustrated in the examples of scope construal (4.8) and extraction (4.9).

4.4 Bibliography

[1] Raffaella Bernardi and Michael Moortgat. Continuation semantics for symmetric categorial grammar. In Daniel Leivant and Ruy de Queiros, editors, *Proceedings 14th Workshop on Logic, Language, Information and Computation (WoLLIC'07)*, LNCS 4576. Springer, 2007.

[2] Matteo Capelletti. *Parsing with structure-preserving categorial grammars.* PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University, 2007.

[3] Philippe de Groote. The non-associative Lambek calculus with product in polynomial time. In N. V. Murray, editor, *Automated Reasoning With Analytic Tableaux and Related Methods*, volume 1617 of *Lecture Notes in Artificial Intelligence*, pages 128–139. Springer, 1999.

[4] V.N. Grishin. On a generalization of the Ajdukiewicz-Lambek system. In A.I. Mikhailov, editor, *Studies in Nonclassical Logics and Formal Systems*, pages 315–334. Nauka, Moscow, 1983. [English translation in Abrusci and Casadio (eds.) Proceedings 5th Roma Workshop, Bulzoni Editore, Roma, 2002].

[5] N. Kurtonina. *Frames and labels. A modal analysis of categorial inference.* PhD thesis, OTS Utrecht University, ILLC Amsterdam University, 1995.

[6] J. Lambek. From categorial to bilinear logic. In K. Došen P. Schröder-Heister, editor, *Substructural Logics*, pages 207–237. Oxford University Press, 1993.

[7] Michael Moortgat and Mati Pentus. Type similarity for the Lambek-Grishin calculus. In *Proceedings 12th Conference on Formal Grammar*, Dublin, 2007.

[8] Mati Pentus. Lambek calculus is NP-complete. *Theoretical Computer Science*, 357:186–201, 2006.

5

Relational completeness of the Lambek-Grishin calculus with unary modalities

Anna Chernilovskaya

Abstract We study Kripke semantics for $LG(\Diamond, \Box, \blacklozenge, \blacksquare, \backslash, \otimes, /, \oslash, \oplus, \oslash)$: bi-Lambek calculus with Grishin Class IV interaction principles for a vocabulary extended with four unary modalities. We extend the filter-based completeness result of Kurtonina and Moortgat [4] to the language with unary modalities.

5.1 Introduction

In this paper we study an extension of the Lambek–Grishin calculus with two pairs of unary modalities. In the middle Nineties Moortgat and Kurtonina [5, 3] introduced an enrichment of the type language of the Lambek calculus with two unary type-forming operations $\Diamond$ and $\Box$. The operations $\Diamond$ and $\Box$ form a residuated pair: they are order-preserving (isotone), and their compositions satisfy $\Diamond\Box A \vdash A \vdash \Box\Diamond A$. Equivalently, they satisfy the biconditional $\Diamond A \vdash B$ iff $A \vdash \Box B$. In the relational semantics, $\Diamond$ and $\Box$ are interpreted in terms of a binary relation of accessibility $R_\Diamond$ which one can think of as a feature checking relation. The unary modalities have found two types of application in linguistic analysis. First, the pattern $\Diamond\Box A \vdash A \vdash \Box\Diamond A$ makes it possible to express selectional restrictions: for each type A there is a more specific $\Diamond\Box A$ and a more general $\Box\Diamond A$. Second, unary modalities have been used to formulate controlled versions of structural rules such as associativity or commutativity, which would be overgenerating when applied globally.

In the symmetric setting of Lambek-Grishin calculus, we consider $\Diamond, \Box$ together with a *dual* pair of unary operators which we will denote as $\blacklozenge$ and $\blacksquare$. The operators $\blacklozenge$ and $\blacksquare$ also form a residuated pair satisfying $\blacklozenge A \vdash B$ iff $A \vdash \blacksquare B$. In addition to the residuation property, we now have to consider the interaction between the white and black modalities, and between the unary and binary type-forming operations, i.e. the analogues of the $\otimes/\oplus$ interactions investigated in [2]. These interactions take the form of weak distributivity principles and are expressed in three postulates:

$$(D1) \quad \blacklozenge A \otimes B \vdash \blacklozenge(A \otimes B) \qquad (D2) \quad A \otimes \blacklozenge B \vdash \blacklozenge(A \otimes B) \qquad (D3) \quad \Diamond\blacklozenge A \vdash \blacklozenge\Diamond A.$$

5.2 The binary vocabulary

Let us briefly recall the Henkin construction of Kurtonina and Moortgat [4].

Truth conditions Fusion/composition ($\otimes$) and its residuals:
$x \Vdash A \otimes B$ iff $\exists yz.R_{\otimes}xyz$ and $y \Vdash A$ and $z \Vdash B$
$y \Vdash C/B$ iff $\forall xz.(R_{\otimes}xyz$ and $z \Vdash B)$ implies $x \Vdash C$
$z \Vdash A\backslash C$ iff $\forall xy.(R_{\otimes}xyz$ and $y \Vdash A)$ implies $x \Vdash C$
Fission ($\oplus$) and its residuals:
$x \Vdash A \oplus B$ iff $\forall yz.R_{\oplus}xyz$ implies ($y \Vdash A$ or $z \Vdash B$)
$y \Vdash C \oslash B$ iff $\exists xz.R_{\oplus}xyz$ and $z \nVdash B$ and $x \Vdash C$
$z \Vdash A \obslash C$ iff $\exists xy.R_{\oplus}xyz$ and $y \nVdash A$ and $x \Vdash C$

Henkin construction To establish completeness, one uses a Henkin model with (weak) filters $\mathcal{F}_{\vdash}$ as worlds.
The set of filters $\mathcal{F}_{\vdash}$ is closed under the following two operations, in terms of which one then defines the canonical model.
$X \mathbin{\widehat{\otimes}} Y = \{C \mid \exists A, B\ (A \in X$ and $B \in Y$ and $A \otimes B \vdash C)\}$
$X \mathbin{\widehat{\obslash}} Y = \{B \mid \exists A, C\ (A \notin X$ and $C \in Y$ and $A \obslash C \vdash B\}$

Canonical model The canonical model is defined as $\mathcal{M}^c = \langle W^c, R^c_{\otimes}, R^c_{\oplus}, V^c\rangle$ with

$$W^c = \mathcal{F}_{\vdash}$$

$$R^c_{\otimes}XYZ \text{ iff } Y \mathbin{\widehat{\otimes}} Z \subseteq X$$

$$R^c_{\oplus}XYZ \text{ iff } Y \mathbin{\widehat{\obslash}} X \subseteq Z$$

$$V^c(p) = \{X \in W^c \mid p \in X\}$$

Completeness theorem If $\models A \to B$, then $A \vdash B$ is provable in the miminal symmetric Lambek-Grishin calculus.

The proof follows immediately from the usual Truth Lemma, showing that for any formula $A \in \mathcal{F}$ and filter $X \in \mathcal{F}_{\vdash}$, $X \Vdash A$ iff $A \in X$. For the minimal symmetric Lambek-Grishin calculus, completeness holds for arbitrary frames (there are no constraints on the interpretation of $R_{\otimes}$ and $R_{\oplus}$). For systems with extra interaction postulates relating the $\otimes$ and $\oplus$ families, one has extended completeness for interpretations respecting the frame constraints corresponding to these postulates.

In the following section, we extend this construction to the language with unary modalities. Before doing so, we remark that, in addition to $\cdot \mathbin{\widehat{\obslash}} \cdot$, one can also define operations $\cdot \mathbin{\widehat{\oslash}} \cdot$ and $\cdot \mathbin{\widehat{\oplus}} \cdot$ on the set of filters $\mathcal{F}_{\vdash}$:
$Y \mathbin{\widehat{\oslash}} X = \{A \mid \exists B, C\ (B \notin X$ and $C \in Y$ and $C \oslash B \vdash A\}$, alternatively
$Y \mathbin{\widehat{\oslash}} X = \{A \mid \exists B, C\ (B \notin X$ and $C \in Y$ and $C \vdash A \oplus B\}$
$X \mathbin{\widehat{\oplus}} Y = \{C \mid \forall A, B\ (C \vdash A \oplus B$ implies $(A \in X$ or $B \in Y)\}$
To see that $X \mathbin{\widehat{\oplus}} Y$ is a filter, assume $C \in X \mathbin{\widehat{\oplus}} Y$ and $C \vdash D$. We have to show that $D \in X \mathbin{\widehat{\oplus}} Y$. $C \in X \mathbin{\widehat{\oplus}} Y$ iff $\forall A, B\ (C \vdash A \oplus B$ implies $(A \in X$ or $B \in Y))$. Take A', B' such that $D \vdash A' \oplus B'$

and $A' \notin X$. We need to prove that $B' \in Y$. We know that $C \vdash D$, hence $C \vdash A' \oplus B'$. Together with $A' \notin X$, we get $B' \in Y$, as desired.

Theorem The structure $(\mathcal{F}_{\vdash}, \subseteq^{op}, \widehat{\oplus}, \widehat{\otimes}, \widehat{\oslash})$ is a residuated groupoid, i.e.

$$X \subseteq^{op} Z \mathbin{\widehat{\oslash}} Y \quad \text{iff} \quad X \mathbin{\widehat{\oplus}} Y \subseteq^{op} Z \quad \text{iff} \quad Y \subseteq^{op} X \mathbin{\widehat{\otimes}} Z$$

and we can reformulate the canonical accessibility relations as

$$R^c_{\otimes} XYZ \text{ iff } Y \mathbin{\widehat{\otimes}} Z \subseteq X \qquad R^c_{\oplus} XYZ \text{ iff } X \subseteq Y \mathbin{\widehat{\oplus}} Z.$$

PROOF

In fact, we have to prove that

$$X \supseteq Z \mathbin{\widehat{\oslash}} Y \quad \text{iff} \quad X \mathbin{\widehat{\oplus}} Y \supseteq Z \quad \text{iff} \quad Y \supseteq X \mathbin{\widehat{\otimes}} Z.$$

Let us prove $X \mathbin{\widehat{\oplus}} Y \supseteq Z$ iff $Y \supseteq X \mathbin{\widehat{\otimes}} Z$, the left equivalence can be done symmetrically.

($\Rightarrow$) Take $B \in X \mathbin{\widehat{\otimes}} Z$. This means that $\exists A, C$ ($A \notin X$ and $C \in Z$ and $A \otimes C \vdash B$). We know that $X \mathbin{\widehat{\oplus}} Y \supseteq Z$ then $C \in X \mathbin{\widehat{\oplus}} Y$, i.e. $\forall C_1, C_2$ ($C \vdash C_1 \oplus C_2$ implies ($C_1 \in X$ or $C_2 \in Y$)). By residuation, $C \vdash A \oplus B$. Taking into account that $A \notin X$, we conclude $B \in Y$ as desired.

($\Leftarrow$) Take $C \in Z$. We have to show that $C \in X \mathbin{\widehat{\oplus}} Y$, i.e. that $\forall A, B$ ($C \vdash A \oplus B$ implies ($A \in X$ or $B \in Y$)). Take A, B such that $C \vdash A \oplus B$ and $A \notin X$. Let us prove that $B \in Y$. By Residuation, we have $A \otimes C \vdash B$. Hence we get $B \in X \mathbin{\widehat{\otimes}} Z$ since $\exists A, C$ ($A \notin X$ and $C \in Z$ and $A \otimes C \vdash B$). By the condition $Y \supseteq X \mathbin{\widehat{\otimes}} Z$, hence $B \in Y$ as desired. □

5.3 Unary modalities

For the binary vocabulary, we have an axiomatization in terms of the residuation rules, monotonicity rules and (if desired) interaction postulates in rule form. Transitivity is an admissible rule for this axiomatization, which then yields a decision procedure. For our four unary modalities we initially consider the following set of rules. (Rules for the interaction principles $D1 - D3$ will be introduced later.)

(Dual) residuation principles

$$\frac{A \vdash \Box B}{\Diamond A \vdash B} \qquad \frac{A \vdash \blacksquare B}{\blacklozenge A \vdash B}$$

$$\frac{\Diamond A \vdash B}{A \vdash \Box B} \qquad \frac{\blacklozenge A \vdash B}{A \vdash \blacksquare B}$$

Monotonicity principles

$$\frac{A \vdash B}{\Diamond A \vdash \Diamond B} \qquad \frac{A \vdash B}{\blacklozenge A \vdash \blacklozenge B}$$

$$\frac{A \vdash B}{\Box A \vdash \Box B} \qquad \frac{A \vdash B}{\blacksquare A \vdash \blacksquare B}$$

Consider the two symmetries that we have found in $LG(\backslash, \otimes, /, \oslash, \oplus, \oslash)$: one left-right order-preserving symmetry $\bowtie$, the other is order-reversing ∞: $A \vdash B$ iff $A^{\bowtie} \vdash B^{\bowtie}$ iff $B^{\infty} \vdash A^{\infty}$.

$$\bowtie \quad \frac{C/D \quad A \otimes B \quad B \oplus A \quad D \oslash C}{D\backslash C \quad B \otimes A \quad A \oplus B \quad C \oslash D}$$

$$\infty \quad \frac{C/B \quad A \otimes B \quad A\backslash C}{B \oslash C \quad B \oplus A \quad C \oslash A}$$

The images of the modalities are the following:

$$\begin{array}{rclcrcl} (\Diamond A)^{\bowtie} & = & \blacklozenge(A^{\bowtie}) & \quad & (\Box A)^{\bowtie} & = & \blacksquare(A^{\bowtie}) \\ (\Diamond A)^{\infty} & = & \blacksquare(A^{\infty}) & \quad & (\Box A)^{\infty} & = & \blacklozenge(A^{\infty}) \end{array}$$

The rules for the modalities then comply with a theorem that was originally stated for the Lambek–Grishin calculus without unary modalities: $A \vdash B$ iff $A^{\bowtie} \vdash B^{\bowtie}$ iff $B^{\infty} \vdash A^{\infty}$ (proof is by induction on the length of derivations of $A \vdash B$ in the monotonicity based system with unary modalities).

Truth conditions for unary modalities Truth conditions for $\Diamond$ and $\Box$:
$x \Vdash \Diamond A$ iff $\exists y.R_{\Diamond}xy$ and $y \Vdash A$
$x \Vdash \Box A$ iff $\forall y.R_{\Diamond}yx$ implies $y \Vdash A$
Truth conditions for $\blacklozenge$ and $\blacksquare$:
$x \Vdash \blacklozenge A$ iff $\exists y.R_{\blacklozenge}xy$ and $y \Vdash A$
$x \Vdash \blacksquare A$ iff $\forall y.R_{\blacklozenge}yx$ implies $y \Vdash A$

Henkin construction We define two new (existential) operations on the set of filters:
${}^{\Diamond}X = \{B \mid \exists A\ (A \in X \text{ and } \Diamond A \vdash B)\}$
${}^{\blacklozenge}X = \{B \mid \exists A\ (A \in X \text{ and } \blacklozenge A \vdash B)\}$
Both ${}^{\Diamond}X$ and ${}^{\blacklozenge}X$ are filters: take $B \in {}^{\Diamond}X$ such that $B \vdash C$. Then there is $A \in X$ such that $\Diamond A \vdash B$. Hence, for C there exists $A' = A$ such that $A' \in X$ and $\Diamond A' \vdash C$, i.e. $C \in {}^{\Diamond}X$, as required.

Remark Analogously to the case of the binary connectives, we can define universal unary operations on the set of filters:
${}^{\Box}X = \{B \mid \forall A\ (B \vdash \Box A \text{ implies } A \in X)\}$
${}^{\blacksquare}X = \{B \mid \forall A\ (B \vdash \blacksquare A \text{ implies } A \in X)\}$
These sets are filters. Indeed, take $B \in {}^{\Box}X$ and C such that $B \vdash C$. We want to prove that $C \in X$. Take A' such that $C \vdash \Box A'$. By Transitivity, $B \vdash \Box A'$, hence $A' \in X$, as desired.

Canonical model Let us add to the canonical model having been constructed two new relations:

$R^{c}_{\Diamond}XY$ iff ${}^{\Diamond}Y \subseteq X$

$R^{c}_{\blacklozenge}XY$ iff ${}^{\blacklozenge}Y \subseteq X$

Truth lemma $X \Vdash \Diamond A$ iff $\Diamond A \in X$ (and analogously for $\blacklozenge$: $X \Vdash \blacklozenge A$ iff $\blacklozenge A \in X$).

PROOF

($\Rightarrow$) $X \Vdash \Diamond A$, i.e. $\exists Y$ such that $R^c_\Diamond XY$ and $Y \Vdash A$. By the definition of $R^c_\Diamond XY$ and IH, ${}^\Diamond Y \subseteq X$ and $A \in Y$. By the definition of ${}^\Diamond Y$, $\Diamond A \in {}^\Diamond Y$. Hence $\Diamond A \in X$, as required.

($\Leftarrow$) Suppose $\Diamond A \in X$. We need to prove that $X \Vdash \Diamond A$, i.e. we should find Y such that ${}^\Diamond Y \subseteq X$ and $Y \Vdash A$. Assume $Y = \lfloor A \rfloor$. Then ${}^\Diamond Y = {}^\Diamond \lfloor A \rfloor = \lfloor \Diamond A \rfloor \subseteq X$ since X is a filter and $\Diamond A \in X$. $A \in \lfloor A \rfloor = Y$ then, by IH, $Y \Vdash A$ that realises the second condition. We have proved $X \Vdash \Diamond A$. $\Box$

Interaction principles Let us turn now to the frame conditions imposed on the relations $R^c_\Diamond$ and $R^c_\blacklozenge$ corresponding with the weak distributivity principles (in terms of the existential modalities $\Diamond$, $\otimes$ and $\blacklozenge$):

$$(D1) \quad \blacklozenge A \otimes B \vdash \blacklozenge(A \otimes B) \qquad (D2) \quad A \otimes \blacklozenge B \vdash \blacklozenge(A \otimes B) \qquad (D3) \quad \Diamond\blacklozenge A \vdash \blacklozenge\Diamond A.$$

In terms of universal operations $\Box$, $\blacksquare$, $\oplus$ we would have the postulates below. We leave it to the reader to derive $D1' - D3'$ from $D1 - D3$.

$$(D1') \quad \Box(A \oplus B) \vdash \Box A \oplus B \qquad (D2') \quad \Box(A \oplus B) \vdash A \oplus \Box B \qquad (D3') \quad \Box\blacksquare A \vdash \blacksquare\Box A.$$

For decidable proof search, we put these postulates in rule form (compiling away the use of transitivity):

$$\frac{\blacklozenge(A \otimes B) \vdash C}{\blacklozenge A \otimes B \vdash C}\ R1 \qquad \frac{\blacklozenge(A \otimes B) \vdash C}{A \otimes \blacklozenge B \vdash C}\ R2 \qquad \frac{\blacklozenge\Diamond A \vdash B}{\Diamond\blacklozenge A \vdash B}\ R3$$

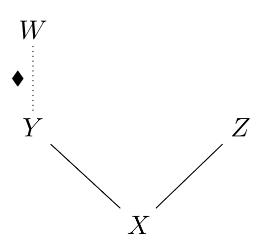
Let us prove that the canonical model with relations for unary modalities satisfies these weak distributivity principles.

Interaction between $\blacklozenge$ and $\otimes$ As a representative, we consider the axiom $(D1)$: $\blacklozenge A \otimes B \vdash \blacklozenge(A \otimes B)$; the second one is dealt with analogously. The frame constraint for $(D1)$ is:

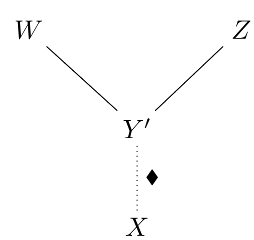
$$\forall xyzw.(R_\otimes xyz \ \wedge \ R_\blacklozenge yw) \Rightarrow \exists y'.(R_\blacklozenge xy' \ \wedge \ R_\otimes y'wz)$$

In the picture below, vertical dotted line represents $R^c_\blacklozenge$ (with an optional $\blacklozenge$ mark on the line), vertical solid line $R^c_\Diamond$ (with an optional $\Diamond$ mark).

(†)



(‡)



In (†) we depict $X \Vdash \blacklozenge A \otimes B$, with $W \Vdash A$ and $Z \Vdash B$; in (‡) $X \Vdash \blacklozenge(A \otimes B)$.

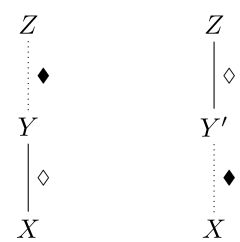
We have to show that in the Henkin model $\forall X, Y, Z, W$ construed as in (†), there is a fresh internal Y' as in (‡). The solution $Y' := W \widehat{\otimes} Z$ gives us $R^c_\otimes Y'WZ$ since $W \widehat{\otimes} Z \subseteq W \widehat{\otimes} Z$. We need also to show that $R^c_\blacklozenge XY'$ holds, i.e. ${}^\blacklozenge Y' \subseteq X$. Suppose $C \in {}^\blacklozenge Y'$. We want to prove that $C \in X$. By the definition of ${}^\blacklozenge Y'$, we have C' such that $C' \in Y'$ and $\blacklozenge C' \vdash C$. By assumption, $C' \in Y'$

means that $C' \in W \,\widehat{\otimes}\, Z$, i.e. there are C_1, C_2 such that $C_1 \in W$ and $C_2 \in Z$ and $C_1 \otimes C_2 \vdash C'$. Taking the configuration (†) together with $C_1 \in W$, we conclude $Y \Vdash \blacklozenge C_1$. Together with the condition $C_2 \in Z$, in (†) it implies that $X \Vdash \blacklozenge C_1 \otimes C_2$. By the Truth lemma, this means that $\blacklozenge C_1 \otimes C_2 \in X$ and since X is a filter and $\blacklozenge C_1 \otimes C_2 \vdash \blacklozenge(C_1 \otimes C_2)$ is an axiom, $\blacklozenge(C_1 \otimes C_2) \in X$. We know that $C_1 \otimes C_2 \vdash C'$, then $\blacklozenge(C_1 \otimes C_2) \vdash \blacklozenge C'$. X is a filter, so $\blacklozenge C' \in X$. But we also have $\blacklozenge C' \vdash C$, hence $C \in X$, as required.

Interaction between $\lozenge$ and $\blacklozenge$ The frame constraint is

$$\forall xyz.(R_\lozenge xy \ \wedge\ R_\blacklozenge yz) \Rightarrow \exists y'.(R_\blacklozenge xy' \ \wedge\ R_\lozenge y'z)$$

.

(†)  (‡)

We have to show that in the Henkin model for all X, Y, Z depicted as in (†) there is a fresh node Y' connecting the root X to the leaf Z. Assuming $Y' := {}^\lozenge Z$, we have $R^c_\lozenge Y'Z$ since ${}^\lozenge Z \subseteq {}^\lozenge Z$. Now we have to prove that $R^c_\blacklozenge XY'$ holds. This means to prove $\blacklozenge Y' \subseteq X$. Take $B \in \blacklozenge Y'$, i.e. such that there is $B_1 \in Y'$ and $\blacklozenge B_1 \vdash B$. Since $Y' = {}^\lozenge Z$, the fact that $B_1 \in Y'$ actually means $\exists B_2$ such that $B_2 \in Z$ and $\lozenge B_2 \vdash B_1$. Taking the configuration (†) together with $B_2 \in Z$, we conclude that $Y \Vdash \blacklozenge B_2$. Again from the configuration (†) we see that $X \Vdash \lozenge\blacklozenge B_2$. By the Truth lemma, $\lozenge\blacklozenge B_2 \in X$. Since X is a filter and $\lozenge\blacklozenge B_2 \vdash \blacklozenge\lozenge B_2$ is an axiom, we have $\blacklozenge\lozenge B_2 \in X$. We know that $\lozenge B_2 \vdash B_1$, then, by Monotonicity for $\blacklozenge$, $\blacklozenge\lozenge B_2 \vdash \blacklozenge B_1$. Hence $\blacklozenge B_1 \in X$. But also $\blacklozenge B_1 \vdash B$ holds, so we have $B \in X$, what we needed to prove.

Remark Suppose that we consider the converses of the weak distributivity principles. Then we can deal with them analogously to how we have just done.

Soundness can be established by a straightforward induction on the length of derivations in $LG(\lozenge, \Box, \blacklozenge, \blacksquare, \backslash, \otimes, /, \oslash, \oplus, \oslash)$. We then arrive at the theorem below for symmetric Lambek-Grishin calculus without interaction postulates. Completeness extends to the systems *with* interactions, if one restricts to interpretation on frames respecting the constraints corresponding to the interaction postulates.

Theorem (Soundness/Completeness) In $\mathbf{LG}(\lozenge, \Box, \blacklozenge, \blacksquare, \backslash, \otimes, /, \oslash, \oplus, \oslash)$,

$$A \vdash B \text{ is provable iff } \models A \vdash B.$$

5.4 Linguistic illustration

Usually unary modalities $\lozenge$ and $\Box$ are used to describe linguistic phenomena like extraction. Here one uses unary modalities in order to mark a particular structure in newly introduced structural rules of rebracketing.

A nice example (discussed in [6]) that can illustrate the need for having these two unary modalities $\Diamond$ and $\Box$ shows as well a strategy of assigning a meet type. It concerns the phenomenon of crossed dependency in Dutch, as in the phrase '(dat Jan) boeken (c) wil (a/b) lezen ($c\backslash b$)' with the order of 'object–modal auxiliary–transitive infinitive'. We want to derive a type a that is tensed verb phrase. Our aim is twofold: to allow the transitive infinitive to communicate with its direct object across the modal auxiliary; at the same time, we want to rule out the ungrammatical order $(a/b) \otimes (c \otimes (c\backslash b))$ which with the indicated types would make a derivable.

We skip all intermediate steps showing how to deal with the type similarity relation, and stick attention just to the case of making ungrammatical order 'modal auxiliary–object–transitive infinitive' underivable. We make use of modal decoration saying that modal auxiliary will have a type $a/\Diamond\Box b$ instead of just a/b. Then the type for transitive infinitive will be the following: $((c\backslash b)/C) \otimes (C \oslash (((a/\Diamond\Box b)\backslash(c\backslash a)) \obslash C))$ where $C = ((a/b)\backslash a) \oplus (c\backslash(a \obslash a))$. Now recall that in $LG(\Diamond, \Box, \blacklozenge, \blacksquare, \backslash, \otimes, /, \obslash, \oplus, \oslash)$ we have $\Diamond\Box A \vdash A$.

The unary operators provide fine-grained distinction in the type assignments both within and across languages. For example, in languages where there are singular and plural articles (like French, German, Italian) they both are interpreted semantically as $(e \to t) \to t$. However, it is evident they are used in different ways. The unary operators handle this difference in such a way that it is not visible on the level of the domains of interpretation.

Similarly, the cross-linguistic contrast between the way adjectives may combine with nouns in Italian and in English does not play any role in the assignment of the meaning to their composition, but it is relevant for their syntactic assembly in the two languages.

Unfortunately, I haven't been able so far to find an example showing the need for having all four unary modalities together.

5.5 Conclusions, future directions

We have established completeness for $LG(\Diamond, \Box, \blacklozenge, \blacksquare, \backslash, \otimes, /, \obslash, \oplus, \oslash)$, the symmetric Lambek-Grishin calculus with four unary modalities: a pair of residuated operators, and a dual pair. In further research, we would like to consider *LG* would like to compare and combine the extension with residuated operators with the extension with (dual) *Galois connected* modalities, studied by Areces, Bernardi and Moortgat [1]. These Galois connected operators are order-reversing, and they give rise to new interaction possibilities. Moreover, we would like to study the linguistic motivation for having these families of unary operators, and for their interaction.

5.6 Bibliography

[1] Carlos Areces, Raffaella Bernardi, and Michael Moortgat. Galois connections in categorial type logic. In *Proceedings FG/MOL*, volume 47 of *Electronic Notes in Theoretical Computer Science*, pages 1–12. Elsevier, 2001.

[2] V.N. Grishin. On a generalization of the Ajdukiewicz-Lambek system. In A.I. Mikhailov, editor, *Studies in Nonclassical Logics and Formal Systems*, pages 315–334. Nauka, Moscow, 1983. [English translation in Abrusci and Casadio (eds.) Proceedings 5th Roma Workshop, Bulzoni Editore, Roma, 2002].

[3] Natasha Kurtonina and Michael Moortgat. Structural control. In Patrick Blackburn and Maarten de Rijke, editors, *Specifying Syntactic Structures*, pages 75–113. CSLI Publications, Stanford, 1997.

[4] Natasha Kurtonina and Michael Moortgat. Relational semantics for the Lambek-Grishin calculus. Technical report, AI Preprint Series, Utrecht, 2007. (Submitted to MOL'10, UCLA, Los Angeles).

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

[6] Michael Moortgat. Symmetries in natural language syntax and semantics: the Lambek-Grishin calculus. In Daniel Leivant and Ruy de Queiros, editors, *Proceedings 14th Workshop on Logic, Language, Information and Computation (WoLLIC'07)*, LNCS 4576. Springer, 2007.

6

Type similarity for the Lambek-Grishin calculus

Michael Moortgat and Mati Pentus

Abstract We discuss the relation of type similarity for the type calculus **LG**: a symmetric version of the pure logic of residuation with structure-preserving interaction principles. We show that in **LG** the $\sim$ relation can be characterized in terms of an interpretation in the free Abelian group generated by the type atoms. **LG** thus achieves the same level of expressivity with respect to $\sim$ as the associative/commutative calculus **LP** without any loss of structural discrimination.

Thanks The research reported on here was partially supported by the Dutch-Russian cooperation program NWO/RFBR, project No. 047.017.01 "Logical models of human and mechanical reasoning".

6.1 Background, motivation

In the algebraic remarks at the end of [7], Lambek introduces the notion of type similarity $\sim$, i.e. the reflexive, transitive, symmetric closure of the derivability relation.

Definition $A \sim B$ iff there exists a sequence $C_1 \ldots C_n$ $(1 \leq n)$ such that $C_1 = A$, $C_n = B$ and $C_i \to C_{i+1}$ or $C_{i+1} \to C_i$ $(1 \leq i < n)$.

He proves that $A \sim B$ if and only if one of the following equivalent statements hold (the so-called diamond property):

$\exists C$ such that $A \to C$ and $B \to C$

$\exists D$ such that $D \to A$ and $D \to B$

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 [7] are given in (6.1). It is shown in [4] that these solutions are in fact adequate for the pure logic of residuation, i.e. the non-associative calculus **NL**.

$$\begin{array}{ll}\textbf{NL}: & D = (A/((C/C)\backslash C)) \otimes ((C/C)\backslash B), \\ & C = (A \otimes (D\backslash D))/(B\backslash(D \otimes (D\backslash D)))\end{array} \tag{6.1}$$

CALCULUS	INTERPRETATION
NL	free quasigroup [4]
L	free group [9]
LP	free Abelian group[9]

Table 6.1: Models for type equivalence

For associative **L**, [9] has the shorter solution in (6.2). The possibility of rebracketing the types for D and C is what makes this solution work.

$$\mathbf{L}: \; D = (A/C) \otimes C \otimes (C\backslash B), \quad C = (D/A)\backslash D/(B\backslash D) \tag{6.2}$$

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 6.1 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: $[\![p]\!] = p$, $[\![A/B]\!] = [\![A]\!]/[\![B]\!]$, $[\![B\backslash A]\!] = [\![B]\!]\backslash[\![A]\!]$, $[\![A \otimes B]\!] = [\![A]\!] \cdot [\![B]\!]$.[1] 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 $[\![p]\!] = p$, $[\![A \otimes B]\!] = [\![A]\!] \cdot [\![B]\!]$, $[\![A/B]\!] = [\![A]\!] \cdot [\![B]\!]^{-1}$, $[\![B\backslash A]\!] = [\![B]\!]^{-1} \cdot [\![A]\!]$.
We see in Table 6.1 that expressivity for $\sim$ is inversely proportional to structural discrimination: the structural rules of associativity and commutativity destroy sensitivity for constituent structure and word order. We want to investigate an alternative strategy to overcome the expressive limitations of **NL** based on [5].

6.2 Lambek-Grishin calculus

The generalizations of Lambek calculus studied by Grishin in his 1983 paper all start from a *symmetric version* of the type language. In addition to the product $\otimes$ and left and right division operations $\backslash, /$, the language also contains a dual family consisting of a coproduct $\oplus$ and right and left difference operations $\oslash$, $\obslash$. The minimal symmetric calculus is defined by the preorder axioms (6.3), together with the residuation laws of (6.4) and (6.5).

$$A \to A; \qquad \text{from } A \to B \text{ and } B \to C \text{ infer } A \to C \tag{6.3}$$

$$A \to C/B \quad \text{iff} \quad A \otimes B \to C \quad \text{iff} \quad B \to A\backslash C \tag{6.4}$$

$$B \obslash C \to A \quad \text{iff} \quad C \to B \oplus A \quad \text{iff} \quad C \oslash A \to B \tag{6.5}$$

This minimal system can be extended in two directions. On the one hand, there are the familiar same-sort associativity and/or commutativity options that give rise to (symmetric versions of) **L** and **LP**. We do not consider these options, since they are not structure-preserving. On the other hand, Grishin introduces interaction principles for the communication between the $\otimes$ and $\oplus$ families. It is this second strategy that we will explore here. With **LG** we designate the type calculus that extends the (6.3), (6.4), (6.5) system with the interaction principles given in (6.6).

$$\begin{array}{ll} (A \obslash B) \otimes C \to A \obslash (B \otimes C) & C \otimes (B \oslash A) \to (C \otimes B) \oslash A \\ C \otimes (A \obslash B) \to A \obslash (C \otimes B) & (B \oslash A) \otimes C \to (B \otimes C) \oslash A \end{array} \tag{6.6}$$

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

The above group of principles (Class IV in Grishin's original terminology) represents one particular choice for $\otimes/\oplus$ interaction. The alternative choice (Grishin's Class I) consists of the converses of (6.6), obtained by turning around the turnstile. Linguistic investigation of the Lambek-Grishin approach has provided solid support for the Class IV interactions, cf. the analysis of scope construal in [1]. Linguistic motivation for Class I interaction so far is less clear.

For decidable proof search, we are interested in an axiomatization which has transitivity as an admissible rule. Such an axiomatization for **LG** can be given in terms of the identity axiom $1_A : A \to A$ together with the residuation principles (6.7), the Grishin axioms in rule form (6.8), and the monotonicity rules of (6.9). We give these rules with combinator proof terms, so that in the remainder we can succinctly refer to derivations by their combinator. The residuation rules are invertible; we write $\triangleleft', \triangleright', \blacktriangleleft', \blacktriangleright'$ for the reverse direction.

$$\frac{f : A \otimes B \to C}{\triangleleft f : B \to A\backslash C} \quad \frac{f : C \to A \oplus B}{\blacktriangleleft f : A \oslash C \to B}$$
$$\frac{f : A \otimes B \to C}{\triangleright f : A \to C/B} \quad \frac{f : C \to A \oplus B}{\blacktriangleright f : C \oslash B \to A} \tag{6.7}$$

$$\frac{f : A \obslash (B \otimes C) \to D}{\circledless f : (A \obslash B) \otimes C \to D} \quad \frac{f : (A \otimes B) \oslash C \to D}{\circledgtr f : A \otimes (B \oslash C) \to D}$$
$$\frac{f : B \obslash (A \otimes C) \to D}{\circledless^{\star} f : A \otimes (B \obslash C) \to D} \quad \frac{f : (A \otimes C) \oslash B \to D}{\circledgtr^{\star} f : (A \oslash B) \otimes C \to D} \tag{6.8}$$

$$\frac{f : A \to B \quad g : C \to D}{f \otimes g : A \otimes C \to B \otimes D} \quad \frac{f : A \to B \quad g : C \to D}{f \oplus g : A \oplus C \to B \oplus D}$$
$$\frac{f : A \to B \quad g : C \to D}{f/g : A/D \to B/C} \quad \frac{f : A \to B \quad g : C \to D}{f \oslash g : A \oslash D \to B \oslash C} \tag{6.9}$$
$$\frac{f : A \to B \quad g : C \to D}{g\backslash f : D\backslash A \to C\backslash B} \quad \frac{f : A \to B \quad g : C \to D}{g \obslash f : D \obslash A \to C \obslash B}$$

Admissibility of the transitivity rule for this axiomatization is established in [8]. We refer the reader to [6] for a proof of completeness of **LG** with respect to ternary relational semantics, and to [1] for a continuation-passing-style Curry-Howard semantics of **LG** derivations and an analysis of scope construal.

6.3 Type similarity for LG

Let us turn to the characterization of type similarity for **LG**. In §6.3.1 we establish the diamond property and prove some useful lemmas. In §6.3.2 we discuss similarity for formulas falling within the pure $\otimes, /, \backslash$ (or $\oplus, \oslash, \obslash$) fragments. In §6.3.3 we then consider the full vocabulary.

6.3.1 Preliminaries

Let us first consider the diamond property for **LG** . Whereas **NL** has a solution of length 7 for the meet and join types, Lemma 6.3.1 shows that **LG** has a length 5 solution, as was the case for **L**. The **LG** solution relies on the Grishin interaction principles.

Lemma 6.3.1 *In **LG**, the diamond property is established with meet $D = (A/C)\otimes(C\oslash(B\obslash C))$ and join $C = (A\oslash D)\oplus(D/(B\backslash D))$.*

Proof Given $f : B \to C$ and $g : A \to C$ we have $\triangleright'(1_A/ \blacktriangleright \blacktriangleleft'(f \obslash 1_C))$ for $D \to A$ and $\ogreaterthan \blacktriangleright \triangleright'(\blacktriangleleft'(1_B \obslash g)/1_C)$ for $D \to B$. The proofs in tree format are given below.

$$
\dfrac{A \to A \qquad \dfrac{\dfrac{\dfrac{B \to C \quad C \to C}{(C \obslash C) \to (B \obslash C)}\,\obslash}{C \to (C \oplus (B \obslash C))}\,\blacktriangleleft'}{(C \oslash (B \obslash C)) \to C}\,\blacktriangleright}{\dfrac{(A/C) \to (A/(C \oslash (B \obslash C)))}{((A/C) \otimes (C \oslash (B \obslash C))) \to A}\,\triangleright'}\,/
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{B \to B \quad A \to C}{(B \obslash A) \to (B \obslash C)}\,\obslash}{A \to (B \oplus (B \obslash C))}\,\blacktriangleleft' \qquad C \to C}{(A/C) \to ((B \oplus (B \obslash C))/C)}\,/}{((A/C) \otimes C) \to (B \oplus (B \obslash C))}\,\triangleright'}{(((A/C) \otimes C) \oslash (B \obslash C)) \to B}\,\blacktriangleright}{((A/C) \otimes (C \oslash (B \obslash C))) \to B}\,\ogreaterthan
$$

Given $f' : D \to B$ and $g' : D \to A$, we have $\blacktriangleright'(1_A \oslash \triangleright \triangleleft' (f'\backslash 1_D))$ for $A \to C$ and $\blacktriangleleft' \triangleright \olessthan \blacktriangleleft \blacktriangleright'(\triangleleft'(1_B\backslash g') \oslash 1_D)$ for $B \to C$. In tree format:

$$
\dfrac{\dfrac{A \to A \qquad \dfrac{\dfrac{\dfrac{D \to B \quad D \to D}{(B\backslash D) \to (D\backslash D)}\,\backslash}{(D \otimes (B\backslash D)) \to D}\,\triangleleft'}{D \to (D/(B\backslash D))}\,\triangleright}{(A \oslash (D/(B\backslash D))) \to (A \oslash D)}\,\oslash}{A \to ((A \oslash D) \oplus (D/(B\backslash D)))}\,\blacktriangleright'
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{B \to B \quad D \to A}{(B\backslash D) \to (B\backslash A)}\,\backslash}{(B \otimes (B\backslash D)) \to A}\,\triangleleft' \qquad D \to D}{((B \otimes (B\backslash D)) \oslash D) \to (A \oslash D)}\,\oslash}{(B \otimes (B\backslash D)) \to ((A \oslash D) \oplus D)}\,\blacktriangleright'}{((A \oslash D) \obslash (B \otimes (B\backslash D))) \to D}\,\blacktriangleleft}{(((A \oslash D) \obslash B) \otimes (B\backslash D)) \to D}\,\olessthan}{((A \oslash D) \obslash B) \to (D/(B\backslash D))}\,\triangleright}{B \to ((A \oslash D) \oplus (D/(B\backslash D)))}\,\blacktriangleleft'
$$

□

We consider some further properties of type similarity which, in **L**, would be dependent on $\otimes$ associativity. In the absence of associativity, these properties obtain in **LG** by virtue of the Grishin interactions.

Lemma 6.3.2 *Rotation. In **LG**, we have $(A\backslash B)/C \sim A\backslash(B/C)$ for arbitrary A, B, C.*

Proof A join type for $(A\backslash B)/C$ and $A\backslash(B/C)$ is $(A\backslash B) \oplus ((B \obslash B)/C)$ as shown by the derivations in Figure 6.1

□

Next, in **L**, for arbitrary types C, D we have $C\backslash C \sim D/D$. A join type for **L** is $C\backslash((C \otimes D)/D)$, as shown in [9]. In **LG** we have a solution for neutral types which is *derivable* from the **L** solution.

Lemma 6.3.3 *Neutrals. In **LG**, for arbitrary C, D we have $C\backslash C \sim D/D$.*

Proof A join type is $(C\backslash((C \otimes D) \oslash D)) \oplus (D/D)$ as shown in Figure 6.2

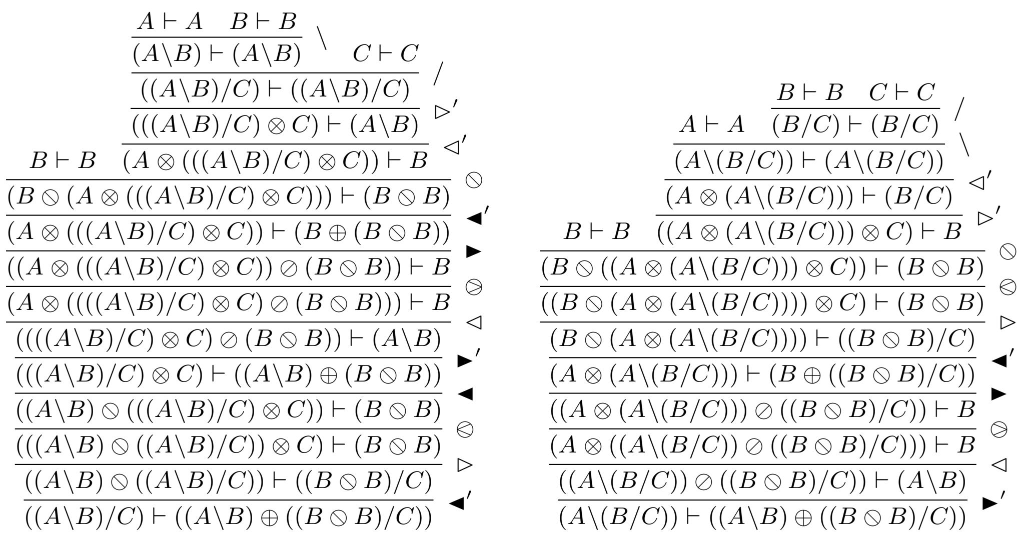


Figure 6.1: Join for $(A\backslash B)/C$ and $A\backslash(B/C)$.

□

The lemma below may be more surprising, and shows that **LG** has the kind of expressivity we expect for **LP**.

Lemma 6.3.4 *Symmetry. For arbitrary **LG** types A, B:* $B\backslash A \sim A/B$.

Proof This time, 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. $Y := ((b/b)/C) \otimes (C \oslash ((b\backslash b) \obslash C))$, where C is $((b\backslash((b \otimes b) \oslash b)) \oplus (b/b)))$.

□

6.3.2 The $\otimes$ and $\oplus$ fragments

We first give a characterization of **LG** similarity for pure $\otimes, \backslash, /$ formulas (and by arrow reversal, pure $\oplus, \obslash, \oslash$ formulas).

$$
\begin{array}{ll}
C \to C \quad C \to C & \\ \hline
(C\backslash C) \to (C\backslash C) & \backslash \\ \hline
(C \otimes (C\backslash C)) \to C \qquad D \to D & \triangleleft' \\ \hline
((C \otimes (C\backslash C)) \otimes D) \to (C \otimes D) \qquad D \to D & \otimes \\ \hline
(((C \otimes (C\backslash C)) \otimes D) \oslash D) \to ((C \otimes D) \oslash D) & \oslash \\ \hline
((C \otimes (C\backslash C)) \otimes D) \to (((C \otimes D) \oslash D) \oplus D) & \blacktriangleright' \\ \hline
(((C \otimes D) \oslash D) \obslash ((C \otimes (C\backslash C)) \otimes D)) \to D & \blacktriangleleft \\ \hline
((((C \otimes D) \oslash D) \obslash (C \otimes (C\backslash C))) \otimes D) \to D & \olessthan \\ \hline
(((C \otimes D) \oslash D) \obslash (C \otimes (C\backslash C))) \to (D/D) & \triangleright \\ \hline
(C \otimes (C\backslash C)) \to (((C \otimes D) \oslash D) \oplus (D/D)) & \blacktriangleleft' \\ \hline
((C \otimes (C\backslash C)) \oslash (D/D)) \to ((C \otimes D) \oslash D) & \blacktriangleright \\ \hline
(C \otimes ((C\backslash C) \oslash (D/D))) \to ((C \otimes D) \oslash D) & \ogreaterthan \\ \hline
((C\backslash C) \oslash (D/D)) \to (C\backslash((C \otimes D) \oslash D)) & \triangleleft \\ \hline
(C\backslash C) \to ((C\backslash((C \otimes D) \oslash D)) \oplus (D/D)) & \blacktriangleright'
\end{array}
$$

$$
\begin{array}{ll}
D \to D \quad D \to D & \\ \hline
(D/D) \to (D/D) & / \\ \hline
C \to C \qquad ((D/D) \otimes D) \to D & \triangleright' \\ \hline
(C \otimes ((D/D) \otimes D)) \to (C \otimes D) \qquad D \to D & \otimes \\ \hline
((C \otimes ((D/D) \otimes D)) \oslash D) \to ((C \otimes D) \oslash D) & \oslash \\ \hline
(C \otimes (((D/D) \otimes D) \oslash D)) \to ((C \otimes D) \oslash D) & \ogreaterthan \\ \hline
(((D/D) \otimes D) \oslash D) \to (C\backslash((C \otimes D) \oslash D)) & \triangleleft \\ \hline
((D/D) \otimes D) \to ((C\backslash((C \otimes D) \oslash D)) \oplus D) & \blacktriangleright' \\ \hline
((C\backslash((C \otimes D) \oslash D)) \obslash ((D/D) \otimes D)) \to D & \blacktriangleleft \\ \hline
(((C\backslash((C \otimes D) \oslash D)) \obslash (D/D)) \otimes D) \to D & \olessthan \\ \hline
((C\backslash((C \otimes D) \oslash D)) \obslash (D/D)) \to (D/D) & \triangleright \\ \hline
(D/D) \to ((C\backslash((C \otimes D) \oslash D)) \oplus (D/D)) & \blacktriangleleft'
\end{array}
$$

Figure 6.2: Join for $C\backslash C$ and D/D.

Theorem 6.3.5 *Let A and B belong to $\boldsymbol{Frm}(\otimes, \backslash, /)$. Then $A \sim B$ in **LG** iff $[\![A]\!] =_{FAG} [\![B]\!]$, where $=_{FAG}$ is equality in the free Abelian group generated by the atomic types.*

Proof Recall the proof in [9] for the free *group* characterization of type equivalence in **L**. One constructs a group $G_\sim$ on the $[\cdot]_\sim$ equivalence classes with respect to the $\sim$ relation, defining $\mathbf{1}_\sim$, the unit of $G_\sim$, as $\mathbf{1}_\sim = [A/A]_\sim = [B\backslash B]_\sim$, and

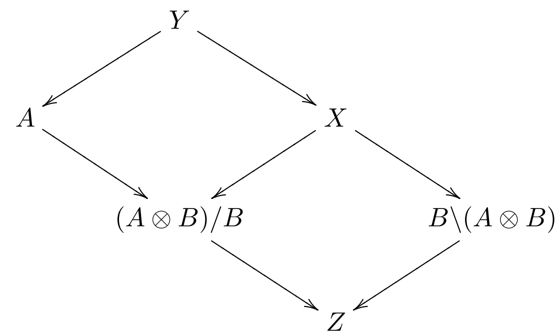
$$[C]_\sim \cdot [D]_\sim = [C \otimes D]_\sim, \qquad [C]_\sim^{-1} = [(C\backslash C)/C]_\sim.$$

One then extends the mapping $[\![p]\!] \mapsto [p]_\sim$ from the generators of FG into $G_\sim$ into a homomorphism h, showing that for any type C, $h([\![C]\!]) = [C]_\sim$. From this it follows that if $[\![A]\!] =_{FG} [\![B]\!]$, then $h([\![A]\!]) = h([\![B]\!])$, i.e. $[A]_\sim = [B]_\sim$, hence $A \sim B$.

We extend this proof to the situation of **LG** . We have to show that $G_\sim$ satisfies the properties of an *Abelian* group.

(Identity) As in **L**.

(Commutativity) We have to show that for all A, B, $[A]_\sim \cdot [B]_\sim = [B]_\sim \cdot [A]_\sim$, i.e. $[A \otimes B]_\sim = [B \otimes A]_\sim$. We have established the existence of a meet type X for $(A \otimes B)/B$ and $B\backslash(A \otimes B)$ in Lemma 6.3.4, showing that $(A \otimes B)/B \sim B\backslash(A \otimes B)$. Since $A \to (A \otimes B)/B$, we also have $A \sim (A \otimes B)/B$. By the diamond property, we can complete the picture with solutions for Y and Z.



We now have $A \sim B\backslash(A \otimes B)$, hence $B \otimes A \sim B \otimes (B\backslash(A \otimes B))$, and since $B \otimes (B\backslash(A \otimes B)) \to (A \otimes B)$ also $B \otimes A \sim A \otimes B$ as desired.

(Inverse) $[C]_\sim^{-1} \cdot [C]_\sim = \mathbf{1}_\sim$ directly follows from the fact that $((C\backslash C)/C) \otimes C \to C\backslash C$ as in **L**. Now $[C]_\sim \cdot [C]_\sim^{-1} = \mathbf{1}_\sim$ follows from commutativity.

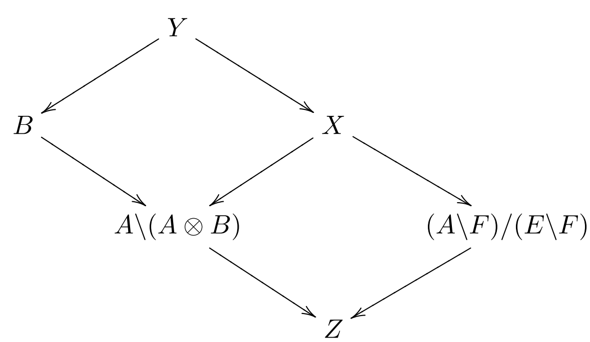
(Associativity) We have to show that for all A, B, C,

$$([A]_\sim \cdot [B]_\sim) \cdot [C]_\sim = [A]_\sim \cdot ([B]_\sim \cdot [C]_\sim)$$

which by the definition of the group product means $[(A \otimes B) \otimes C]_\sim = [A \otimes (B \otimes C)]_\sim$. We follow the same strategy as for the (Commutativity) property. We have $D\backslash E \sim (D\backslash F)/(E\backslash F)$ since $D\backslash E \to D\backslash(F/(E\backslash F))$ by lifting, and $D\backslash(F/(E\backslash F)) \sim (D\backslash F)/(E\backslash F)$ by rotation (Lemma 6.3.2). Let $D := A$, $E := (A \otimes B)$, and $F := ((A \otimes B) \otimes C)$. From the fact that $B \to D\backslash E$, i.e. $B \to A\backslash(A \otimes B)$, we then conclude that also

$$B \sim (D\backslash F)/(E\backslash F) \text{ i.e. } B \sim (A\backslash((A \otimes B) \otimes C))/((A \otimes B)\backslash((A \otimes B) \otimes C))$$

where the existence of meet and join types X, Y, Z in the picture below is guaranteed by the diamond property.



We then have (with $F := ((A \otimes B) \otimes C)$)

$$A \otimes (B \otimes C) \sim A \otimes (((A\backslash F)/(E\backslash F)) \otimes C)$$

and since $C \to E\backslash F$, i.e. $C \to (A \otimes B)\backslash((A \otimes B) \otimes C)$, also $A \otimes (((A\backslash F)/(E\backslash F)) \otimes C) \to F$, which establishes the fact that $A \otimes (B \otimes C) \sim (A \otimes B) \otimes C$ as desired.

□

The above characterization applies to the **Frm**$(/, \otimes, \backslash)$ fragment. We can set up a similar interpretation for formulas from the **Frm**$(\oslash, \oplus, \obslash)$ fragment.

$$\begin{array}{rcl} [\![p]\!] & = & p \\ [\![A \oplus B]\!] & = & [\![A]\!] \cdot [\![B]\!] \\ [\![A \oslash B]\!] & = & [\![A]\!] \cdot [\![B]\!]^{-1} \\ [\![B \obslash B]\!] & = & [\![B]\!]^{-1} \cdot [\![A]\!] \end{array}$$

By arrow reversal, a characterization in terms of free Abelian group equality carries over directly to the formulas from the **Frm**$(\oslash, \oplus, \obslash)$ fragment.

Theorem 6.3.6 *For A, B from* ***Frm***$(\oplus, \oslash, \obslash)$*, in* ***LG*** *we have $A \sim B$ iff $[\![A]\!] =_{FAG} [\![B]\!]$.*

6.3.3 The full language

We turn now to the full language **Frm**$(/, \otimes, \backslash, \oslash, \oplus, \obslash)$. On the one hand, we want to avoid collapse of the vocabulary, so we shouldn't have $A/B \sim A \oslash B$ etc. On the other hand, we should have $(B \oslash C) \obslash A \sim C/(A\backslash B)$ since $(B \oslash C) \obslash A \to C/(A\backslash B)$.

It is well known that equality in the Abelian group can be characterized in terms of literal counts, which are defined as follows.

$$\begin{array}{l} |p|_p = 1, \qquad |q|_p = 0 \text{ if } p \neq q \\ |A \otimes B|_p = |A \oplus B|_p = |A|_p + |B|_p \\ |A/B|_p = |B\backslash A|_p = |A|_p - |B|_p \\ |A \oslash B|_p = |B \obslash A|_p = |A|_p - |B|_p \end{array} \tag{6.10}$$

For example, if A and B belong to **Frm**$(/, \otimes, \backslash)$, then the equality $[\![A]\!] =_{FAG} [\![B]\!]$ holds if and only if $|A|_p = |B|_p$ for all p.

It is natural to bring into play also the operator count, with $|p|_\otimes = |p|_\oplus = 0$ and

$$\begin{array}{ll} |A \otimes B|_\otimes = |A|_\otimes + |B|_\otimes + 1 & |A \otimes B|_\oplus = |A|_\oplus + |B|_\oplus \\ |A \oplus B|_\otimes = |A|_\otimes + |B|_\otimes & |A \oplus B|_\oplus = |A|_\oplus + |B|_\oplus + 1 \\ |A/B|_\otimes = |A|_\otimes - |B|_\otimes - 1 & |A/B|_\oplus = |A|_\oplus - |B|_\oplus \\ |B\backslash A|_\otimes = |A|_\otimes - |B|_\otimes - 1 & |B\backslash A|_\oplus = |A|_\oplus - |B|_\oplus \\ |A \oslash B|_\otimes = |A|_\otimes - |B|_\otimes & |A \oslash B|_\oplus = |A|_\oplus - |B|_\oplus - 1 \\ |B \obslash A|_\otimes = |A|_\otimes - |B|_\otimes & |B \obslash A|_\oplus = |A|_\oplus - |B|_\oplus - 1 \end{array} \tag{6.11}$$

We have the invariant that if $A \to B$, then $|A|_\otimes = |B|_\otimes$ and $|A|_\oplus = |B|_\oplus$, since the operator balance holds for literals, and the inference rules preserve it (monotonicity, residuation shifting and the Grishin rules).

Operator balance fails for a/b vs $a \oslash b$ etc, and holds for formulas such as $(b \oslash c) \obslash a \sim c/(a\backslash b)$, $a/b \sim (a \oslash c)/(b \oslash c)$.

It turns out that the literal counts and the two operator counts are sufficient to characterize type similarity in the full language. Moreover, one of the operator counts can be dropped, since the equalities $|A|_\otimes = |B|_\otimes$ and $|A|_\oplus = |B|_\oplus$ are equivalent provided that $|A|_p = |B|_p$ for all p. This equivalence can be established by a straightforward proof of the equality $\sum_p |A|_p - |A|_\otimes - |A|_\oplus = 1$ by induction on the structure of A.

Now let us define the group interpretation for **Frm**$(/, \otimes, \backslash, \oslash, \oplus, \obslash)$. We shall use the free Abelian group generated by all primitive types and one additional element, denoted by φ. The definition is compatible with that for the **Frm**$(/, \otimes, \backslash)$ fragment, but not with that for the **Frm**$(\oslash, \oplus, \obslash)$ fragment, since we incorporate also the operator count $|\cdot|_\otimes$ into the free Abelian group.

Definition LG group interpretation for **Frm**$(/, \otimes, \backslash, \oslash, \oplus, \obslash)$.

$$\begin{array}{rcl} [\![p]\!] & = & p \\ [\![A \otimes B]\!] & = & [\![A]\!] \cdot [\![B]\!] \\ [\![A/B]\!] & = & [\![A]\!] \cdot [\![B]\!]^{-1} \\ [\![B \backslash A]\!] & = & [\![B]\!]^{-1} \cdot [\![A]\!] \\ [\![A \oplus B]\!] & = & [\![A]\!] \cdot \Phi^{-1} \cdot [\![B]\!] \\ [\![A \oslash B]\!] & = & [\![A]\!] \cdot \Phi \cdot [\![B]\!]^{-1} \\ [\![B \obslash A]\!] & = & [\![B]\!]^{-1} \cdot \Phi \cdot [\![A]\!] \end{array}$$

Lemma 6.3.7 *For any A, B, and C we have*

$$(A \oplus B) \otimes C \sim A \oplus (B \otimes C) \qquad \text{and} \quad (A \oplus B) \otimes C \sim (C \oplus B) \otimes A.$$

Proof We already know that $A \oplus B \sim ((A \obslash A) \oslash A) \obslash B$. The first claim of the lemma is established by the following:

$(A \oplus B) \otimes C \sim (((A \obslash A) \oslash A) \obslash B) \otimes C \sim ((A \obslash A) \oslash A) \obslash (B \otimes C) \sim A \oplus (B \otimes C)$.

It remains to prove the second claim:

$(A \oplus B) \otimes C \sim (B \oplus A) \otimes C \sim B \oplus (A \otimes C) \sim B \oplus (C \otimes A) \sim (B \oplus C) \otimes A \sim (C \oplus B) \otimes A$.

□

Theorem 6.3.8 *Let A, B belong to **Frm**$(/, \otimes, \backslash, \oslash, \oplus, \obslash)$. Then in **LG***

$$A \sim B \quad \textit{iff} \quad [\![A]\!] =_{FAG} [\![B]\!].$$

Proof We extend the proof that was given for the **Frm**$(/, \otimes, \backslash)$ case. On the equivalence classes of **Frm**$(/, \otimes, \backslash, \oslash, \oplus, \obslash)$, a group is constructed in exactly the same way, and the verification of the Abelian group axioms remains the same.

We consider the following mapping from the generators of the free Abelian group into the group of equivalence classes of types:

$$p \mapsto [p]_\sim, \qquad \Phi \mapsto [p_1 \oslash p_1]_\sim.$$

We extend this mapping into a group homomorphism h. It remains to show that $h([\![C]\!]) = [C]_\sim$ for any type C. This is done by induction on the structure of C. The induction base is provided by the construction. The induction step consists of six cases, three of which are exactly the same as in [9]. In what follows, we shall sometimes denote $p_1 \oslash p_1$ by J.

For the case $C = A \oplus B$ we need to establish that

$$A \oplus B \sim (A \otimes ((J \backslash J)/J)) \otimes B.$$

From the fragment **Frm**$(/, \otimes, \backslash)$ we know that $A \otimes ((J \backslash J)/J) \sim A/J$, whence

$$(A \otimes ((J \backslash J)/J)) \otimes B \sim (A/J) \otimes B \sim (A \otimes B)/J.$$

It remains to prove that

$$A \oplus B \sim (A \otimes B)/J,$$

which is equivalent to

$$(A \oplus B) \otimes J \sim A \otimes B.$$

We prove the latter similarity as follows:

$$(A \oplus B) \otimes (p_1 \oslash p_1) \sim ((p_1 \oslash p_1) \oplus B) \otimes A \sim B \otimes A \sim A \otimes B.$$

Here the first similarity follows from the lemma, the second one is obtained from the **Frm**$(\oslash, \oplus, \obslash)$ fragment, and the third one has already been proved for the **Frm**$(/, \otimes, \backslash)$ fragment.

For the case $C = A \oslash B$ we need to establish that

$$A \oslash B \sim (A \otimes J) \otimes ((B \backslash B)/B).$$

From the fragment **Frm**$(/, \otimes, \backslash)$ we know that

$$(A \otimes J) \otimes ((B \backslash B)/B) \sim (A \otimes J)/B.$$

It remains to prove that

$$A \oslash B \sim (A \otimes J)/B,$$

which is equivalent to

$$(A \oslash B) \otimes B \sim A \otimes J.$$

We prove the latter similarity as follows:

$$(A \oslash B) \otimes B \sim (A \otimes B) \oslash B \sim (B \otimes A) \oslash B \sim (B \oslash B) \otimes A \sim J \otimes A \sim A \otimes J.$$

Now the case $C = B \obslash A$ is obvious, since $B \obslash A \sim A \oslash B$ and

$$(((B \backslash B)/B) \otimes J) \otimes A \sim (A \otimes J) \otimes ((B \backslash B)/B).$$

□

6.4 Discussion

To overcome the expressive limitations of **NL** different options present themselves. A well explored strategy is to add structural rules for the composition operations, either globally, or controlled by structural modalities. As an alternative, we have lifted the intuitionistic restriction by considering a residuated family $\oplus, \obslash, \oslash$ which is dual to the usual $\otimes, /, \backslash$ family. In this symmetric setting, one can keep the composition operations free of structural rules, and obtain extra expressivity purely via the *interactions* governing the dual families.

The resulting calculus **LG** is only one of the possible symmetric generalizations of the Lambek calculus considered in [5]. An alternative generalization would replace the four interaction principles (6.6) by their converses. Or one could consider the calculus that results from combining (6.6) and their converses. We have preferred here to study the simplest extension: our results show that already with this extension one can obtain **LP** expressivity with respect to type similarity. We leave the investigation of the converse and/or combined Grishin principles for another occasion.

Discussion of the potential linguistic applications of the $\sim$ relation is beyond the scope of this paper. We sketch one possibility. It is well-known that complement extraction is beyond the reach of **NL**: the object position of 'dedicated' $(((np \backslash s)/pp)/np)$ in 'book that Lewis dedicated _ to Alice' is invisible for the relative pronoun 'that' of type $(n \backslash n)/(s/np)$. The object would be accessible from a type-assignment $(np \backslash (s/np))/pp$ to the verb, with the object promoted to the external argument role. In the pure residuation logic, there is no derivability relation between

$((np\backslash s)/pp)/np$ and $(np\backslash(s/np))/pp$ — and indeed there shouldn't be if one wants to preserve word order. But in **LG** we have $((np\backslash s)/pp)/np \sim (np\backslash(s/np))/pp$, which means there is a meet type for these two types. Assigning this meet type lexically creates a lexical entry that can non-deterministically adapt to its context and behave either as a normal complement-taking verb, or as the head in an extraction environment.

An obvious topic for further research suggested by the results of this paper would be the generative capacity and computational complexity of **LG** and its relatives. The pure residuation logic **NL** and its symmetric extension are known to have a polynomial recognition problem [3, 2]. The simplest structural extension (**L**) is NP-complete [10]. The complexity effects of adding a package of Grishin interactions to the symmetric pure residuation logic are unknown.

6.5 Bibliography

[1] Raffaella Bernardi and Michael Moortgat. Continuation semantics for symmetric categorial grammar. In Daniel Leivant and Ruy de Queiros, editors, *Proceedings 14th Workshop on Logic, Language, Information and Computation (WoLLIC'07)*, LNCS 4576. Springer, 2007.

[2] Matteo Capelletti. *Parsing with structure-preserving categorial grammars*. PhD thesis, Utrecht Institute of Linguistics OTS, Utrecht University, 2007.

[3] Philippe de Groote. The non-associative Lambek calculus with product in polynomial time. In N. V. Murray, editor, *Automated Reasoning With Analytic Tableaux and Related Methods*, volume 1617 of *Lecture Notes in Artificial Intelligence*, pages 128–139. Springer, 1999.

[4] Annie Foret. On the computation of joins for non-associative Lambek categorial grammars. In Jordi Levy, Michael Kohlhase, Joachim Niehren, and Mateu Villaret, editors, *Proceedings of the 17th International Workshop on Unification, UNIF'03 Valencia*, pages 25–38, 2003.

[5] V.N. Grishin. On a generalization of the Ajdukiewicz-Lambek system. In A.I. Mikhailov, editor, *Studies in Nonclassical Logics and Formal Systems*, pages 315–334, Moscow, 1983. Nauka. [English translation in Abrusci and Casadio (eds.) Proceedings 5th Roma Workshop, Bulzoni Editore, Roma, 2002].

[6] Natasha Kurtonina and Michael Moortgat. Relational semantics for the Lambek-Grishin calculus. Technical report, AI Preprint Series, Utrecht, 2007. (Submitted to MOL'10, UCLA, Los Angeles).

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

[8] Michael Moortgat. Symmetries in natural language syntax and semantics: the Lambek-Grishin calculus. In Daniel Leivant and Ruy de Queiros, editors, *Proceedings 14th Workshop on Logic, Language, Information and Computation (WoLLIC'07)*, LNCS 4576. Springer, 2007.

[9] Mati Pentus. 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.

[10] Mati Pentus. Lambek calculus is NP-complete. *Theoretical Computer Science*, 357:186–201, 2006.