Multimodal linguistic inference*

Michael Moortgat
Research Institute for Language and Speech (OTS)
Trans 10, Utrecht
moortgat@let.ruu.nl

To appear in *Journal of Logic, Language and Information*
Special Issue 'Proof Theory and Natural Language'.

Abstract

In this paper we compare grammatical inference in the context of *simple* and of *mixed* Lambek systems. Simple Lambek systems are obtained by taking the logic of residuation for a family of multiplicative connectives $/, \bullet, \backslash$, together with a package of structural postulates characterizing the resource management properties of the $\bullet$ connective. Different choices for Associativity and Commutativity yield the familiar logics **NL**, **L**, **NLP**, **LP**. Semantically, a simple Lambek system is a unimodal logic: the connectives get a Kripke style interpretation in terms of a single ternary accessibility relation modeling the notion of linguistic composition for each individual system.

The simple systems each have their virtues in linguistic analysis. But none of them in isolation provides a basis for a full theory of grammar. In the second part of the paper, we consider two types of *mixed* Lambek systems.

The first type is obtained by combining a number of unimodal systems into one multimodal logic. The combined multimodal logic is set up in such a way that the individual resource management properties of the constituting logics are preserved. But the inferential capacity of the mixed logic is greater than the sum of its component parts through the addition of interaction postulates, together with the corresponding interpretive constraints on frames, regulating the *communication* between the component logics.

The second type of mixed system is obtained by generalizing the residuation scheme for binary connectives to families of n-ary connectives, and by putting together families of different arities in one logic. We focus on residuation for unary connectives, hence on mixed (2,3) frames. The unary connectives play the role of *control* devices, both with respect to the static aspects of linguistic structure, and the dynamic aspects of putting this structure together. We prove a number of elementary logical results for unary families of residuated connectives and their combination with binary families, and situate existing proposals for 'structural modalities' within a more general framework.

*An earlier version of the paper (under the title 'Residuation in mixed Lambek systems') was presented at the 'Deduction and Language' workshop, London, March 1994, and the Rome workshop on Lambek Calculus, May 1994, and appeared in the *IGPL* Bulletin special issue *Deduction and Language* (Kempson (ed), Vol 3, 2–3). The materials in this paper prepare the ground for the sections on multiplicative connectives in a chapter on categorial grammar for the *Handbook of Logic and Language* (Van Benthem and ter Meulen (eds), Elsevier, to appear). I thank the editors of the Handbook and the audiences at these workshops for comments. I have greatly benefited from discussion, and joint work, with Natasha Kurtonina, Glyn Morrill and Dick Oehrle. All errors are my own.

1 Linguistic inference: simple Lambek systems

In this paper, we present categorial grammar as a system of linguistic inference — a logic for reasoning about linguistic resources. The logic has a language of type formulae: atomic formulae, or complex ones, constructed in terms of type-forming connectives — our logical constants. We study the categorial language from a modeltheoretic and a prooftheoretic point of view. The language of type formulae is used to talk about the linguistic reality that forms the object of grammatical analysis: a reality of structured linguistic expressions. The models for the type language are abstract mathematical structures that capture the relevant aspects of the linguistic reality we are interested in. Moving to the prooftheoretic perspective, we want to know how to perform valid inferences on the basis of our interpreted type language. We are not interested in syntax as the manipulation of meaningless symbols: we want our grammatical proof theory to be sound and complete with respect to the intended models of the linguistic reality. And, from a more computational point of view, we are interested in decidabilty and tractability as well.

BINARY MULTIPLICATIVES. To prepare the ground for the exploration of multimodal architectures in §2.1 and §2.2, let us briefly review the essentials (modeltheoretically and prooftheoretically) of the more familiar inhabitants of the categorial landscape.

Consider the language $\mathcal{F}$ of category formulae of a simple Lambek system. $\mathcal{F}$ is obtained by closing a set $\mathcal{A}$ of atomic formulae (or: basic types, prime formulae, e.g. s, np, n, ...) under binary connectives (or: type forming operators) $/, \bullet, \backslash$.

$$\mathcal{F} ::= \mathcal{A} \mid \mathcal{F}/\mathcal{F} \mid \mathcal{F} \bullet \mathcal{F} \mid \mathcal{F}\backslash\mathcal{F}$$

Type formulae have a quite general interpretation in the power set algebra of Kripke style relational structures — ternary relational structures in the case of the binary connectives ([11]). A *ternary frame* is a structure $\langle W, R^3\rangle$. In the application to formal grammar envisaged here, the domain W is to be thought of as a set of *linguistic resources* (or: signs, pieces of multidimensional linguistic information). The accessibility relation R can be understood as representing linguistic composition: $Rxyz$ holds in case one can fuse together the information of signs y and z into a sign x. We obtain a model by adding a valuation v sending prime formulae to subsets of W and satisfying the clauses below for compound formulae.

$$\begin{array}{rcl} v(A \bullet B) & = & \{x \mid \exists y \exists z [Rxyz \ \& \ y \in v(A) \ \& \ z \in v(B)]\} \\ v(C/B) & = & \{y \mid \forall x \forall z [(Rxyz \ \& \ z \in v(B)) \Rightarrow x \in v(C)]\} \\ v(A\backslash C) & = & \{z \mid \forall x \forall y [(Rxyz \ \& \ y \in v(A)) \Rightarrow x \in v(C)]\} \end{array}$$

We are interested in characterizing a relation of derivability between formulae such that $A \to B$ is provable iff $v(A) \subseteq v(B)$. It is not difficult to check that given the above interpretation of compound formulae, the Residuation laws below determine the properties of $\bullet$ vis à vis $/, \backslash$ with respect to derivability.

$$(\textsc{res}) \qquad A \to C/B \quad \Longleftrightarrow \quad A \bullet B \to C \quad \Longleftrightarrow \quad B \to A\backslash C$$

Putting things together, we see that the anatomy of the most elementary Lambek type logic is given by the basic properties of the derivability relation (Reflexivity, Transitivity) plus the Residuation Laws establishing the relation between $\bullet$ and the two implications $/, \backslash$. Below we give the axiomatic presentation of the system known as **NL**. Following [34], we add combinator proof terms: they will provide a compact way of referring to complete deductions later on. Via a canonical model construction Došen [11] obtains the elementary soundness and completeness result: in **NL** provability coincides with semantic inclusion for all ternary frames and all interpretations v.

NL: THE PURE LOGIC OF RESIDUATION. Combinator proof terms. We write $f : A \to B$ for a proof of the inclusion $v(A) \subseteq v(B)$.

$$\mathbf{id}_A : A \to A \qquad \frac{f : A \to B \quad g : B \to C}{g \circ f : A \to C}$$

$$\frac{f : A \bullet B \to C}{\beta(f) : A \to C/B} \qquad \frac{f : A \bullet B \to C}{\gamma(f) : B \to A\backslash C}$$

$$\frac{g : A \to C/B}{\beta^{-1}(g) : A \bullet B \to C} \qquad \frac{g : B \to A\backslash C}{\gamma^{-1}(g) : A \bullet B \to C}$$

STRUCTURAL POSTULATES, CONSTRAINTS ON FRAMES. Starting from the pure logic of residuation **NL** one can unfold a landscape of categorial type logics by gradually relaxing structure sensitivity in a number linguistically relevant dimensions. Below we consider the dimensions of linear precedence (order sensitivity) and immediate dominance (constituent sensitivity). Adding the structural postulates for Associativity or Commutativity (or both) to the pure logic of residuation, one obtains the systems **L**, **NLP**, **LP**. Using Correspondence Theory [6] one computes frame conditions restricting the interpretation of R^3 for the different structural postulates. Došen's completeness result for **NL** is then extended to the stronger logics by restricting the attention to ASS (**L**), COMM (**NLP**) or ASS+COMM frames (**LP**).

	STRUCTURAL POSTULATE	FRAME CONDITION
ASS	$A \bullet (B \bullet C) \longleftrightarrow (A \bullet B) \bullet C$	$\exists t.Rtxy \;\&\; Rutz \Leftrightarrow \exists v.Rvyz \;\&\; Ruxv$
COMM	$A \bullet B \to B \bullet A$	$Rzxy \Leftrightarrow Rzyx$

GENTZEN CALCULUS. The axiomatic presentation is the proper vehicle for model-theoretic investigation of the logics we have considered: it closely follows the semantics, thus providing a suitable basis for 'easy' completeness results. But proof-theoretically the axiomatic presentation has a serious drawback: because it is essentially based on Transitivity, it does not offer an appropriate basis for proof search. For proof-theoretic investigation of the categorial type logics one introduces a Gentzen presentation, and proves a Cut Elimination result, with its corollaries of decidability and the subformula property. Of course, one has to establish the equivalence between the axiomatic and the Gentzen presentations of the logic for all this to make sense. For **L** Lambek [33] has established the essential results. They have been extended to the full landscape of type logics in [26, 12].

In the axiomatic presentation, we consider arrows $A \to B$ with $A, B \in \mathcal{F}$. In Gentzen presentation, the derivability relation is stated to hold between a *term* $\mathcal{T}$ (the antecedent) and a type formula (the succedent). A Gentzen term is a structured configuration of formulae — a structured database, in the terminology of Gabbay [16]. The term language is defined inductively as $\mathcal{T} ::= \mathcal{F} \mid (\mathcal{T}, \mathcal{T})$. The binary structural connective $(\cdot,\cdot)$ in the term language tells you how to put together structured databases Δ_1 and Δ_2 into a structured database (Δ_1, Δ_2). The structural connective mimics the logical connective $\bullet$ in the type language. A sequent is a pair (Γ, A) with $\Gamma \in \mathcal{T}$ and $A \in \mathcal{F}$, written as $\Gamma \Rightarrow A$.

To compare the two presentations, we define the formula equivalent Δ° of a structured database Δ. Let $(\Delta_1, \Delta_2)^\circ = \Delta_1^\circ \bullet \Delta_2^\circ$, and $A^\circ = A$ for $A \in \mathcal{F}$. The Gentzen presentation can be shown to be equivalent to the combinator axiomatisation in the sense of the following proposition from [33].

> Every combinator $f : A \to B$ gives a proof of $A \Rightarrow B$, and every proof of a sequent $\Gamma \Rightarrow B$ gives a combinator $f : \Gamma^\circ \to B$.

As was the case for the combinator presentation, the sequent architecture consists of three components: (i) [Ax] and [Cut] capture the basic properties of the derivability relation '$\Rightarrow$': reflexivity and contextualized transitivity for the 'surgical' Cut, (ii) each connective comes with two *logical rules*:

a rule of use introducing the connective to the left of '$\Rightarrow$' and a rule of proof introducing it on the right of '$\Rightarrow$', finally (iii) there is a block of *structural rules*, possibly empty, with different packages of structural rules resulting in systems with different resource management properties.

Gentzen presentation: structured databases. Sequents $\mathcal{T} \Rightarrow \mathcal{F}$ where $\mathcal{T} ::= \mathcal{F} \mid (\mathcal{T}, \mathcal{T})$. Notation: $\Gamma[\Delta]$ for an antecedent term Γ containing a distinguished occurrence of the subterm Δ.

$$[\mathrm{Ax}]\frac{}{A \Rightarrow A} \qquad \frac{\Delta \Rightarrow A \quad \Gamma[A] \Rightarrow C}{\Gamma[\Delta] \Rightarrow C}[\mathrm{Cut}]$$

$$[/\mathrm{R}]\frac{(\Gamma, B) \Rightarrow A}{\Gamma \Rightarrow A/B} \qquad \frac{\Delta \Rightarrow B \qquad \Gamma[A] \Rightarrow C}{\Gamma[(A/B, \Delta)] \Rightarrow C}[/\mathrm{L}]$$

$$[\backslash\mathrm{R}]\frac{(B, \Gamma) \Rightarrow A}{\Gamma \Rightarrow B\backslash A} \qquad \frac{\Delta \Rightarrow B \qquad \Gamma[A] \Rightarrow C}{\Gamma[(\Delta, B\backslash A)] \Rightarrow C}[\backslash\mathrm{L}]$$

$$[\bullet\mathrm{L}]\frac{\Gamma[(A, B)] \Rightarrow C}{\Gamma[A \bullet B] \Rightarrow C} \qquad \frac{\Gamma \Rightarrow A \qquad \Delta \Rightarrow B}{(\Gamma, \Delta) \Rightarrow A \bullet B}[\bullet\mathrm{R}]$$

Structural rules. Adding the structural rules of Permutation and/or Associativity one obtains coarser notions of linguistic inference, where structural discrimination with respect to the dimensions of precedence and/or dominance is destroyed.

$$\frac{\Gamma[(\Delta_2, \Delta_1)] \Rightarrow A}{\Gamma[(\Delta_1, \Delta_2)] \Rightarrow A}[\mathrm{P}] \qquad \frac{\Gamma[((\Delta_1, \Delta_2), \Delta_3)] \Rightarrow A}{\Gamma[(\Delta_1, (\Delta_2, \Delta_3))] \Rightarrow A}[\mathrm{A}]$$

For the logics **L** and **LP** where $\bullet$ is associative, resp. associative and commutative, explicit application of the structural rules is generally compiled away by means of syntactic sugaring of the sequent language. Antecendent terms then take the form of sequences of formulae $\mathcal{F}, \ldots, \mathcal{F}$ where the comma is now of variable arity, rather than a binary connective. Reading these antecedents as sequences, one avoids explicit reference to the Associativity rule; reading them as multisets, one also makes Permutation implicit.

Characteristic theorems, derived rules of inference. We close this overview with an inventory of theorems and derived inference rules for the various logics.

1. Application: $A/B \bullet B \to A$, $B \bullet B\backslash A \to A$
2. Co-application: $A \to (A \bullet B)/B$, $A \to B\backslash(B \bullet A)$
3. Monotonicity $\bullet$: if $A \to B$ and $C \to D$, then $A \bullet C \to B \bullet D$
4. Isotonicity $\cdot/C$, $C\backslash\cdot$: if $A \to B$, then $A/C \to B/C$ and $C\backslash A \to C\backslash B$
5. Antitonicity $C/\cdot$, $\cdot\backslash C$: if $A \to B$, then $C/B \to C/A$ and $B\backslash C \to A\backslash C$
6. Lifting: $A \to B/(A\backslash B)$, $A \to (B/A)\backslash B$
7. Geach (main functor): $A/B \to (A/C)/(B/C)$, $B\backslash A \to (C\backslash B)\backslash(C\backslash A)$
8. Geach (secondary functor): $B/C \to (A/B)\backslash(A/C)$, $C\backslash B \to (C\backslash A)/(B\backslash A)$
9. Composition: $A/B \bullet B/C \to A/C$, $C\backslash B \bullet B\backslash A \to C\backslash A$
10. Restructuring: $(A\backslash B)/C \longleftrightarrow A\backslash(B/C)$
11. (De)Currying: $A/(B \bullet C) \longleftrightarrow (A/C)/B$, $(A \bullet B)\backslash C \longleftrightarrow B\backslash(A\backslash C)$

12. Permutation: if $A \to B\backslash C$ then $B \to A\backslash C$

13. Exchange: $A/B \longleftrightarrow B\backslash A$

14. Preposing/Postposing: $A \to B/(B/A)$, $A \to (A\backslash B)\backslash B$

15. Mixed Composition: $A/B \bullet C\backslash B \to C\backslash A$, $B/C \bullet B\backslash A \to A/C$

Items (1) to (5) are valid in the weakest logic **NL**. Together they provide an alternative way of characterizing ($\bullet$,/) and ($\bullet$,\) as residuated pairs, i.e. one can replace the RES inferences by (1)–(5). See [12] and §2.2 below. Lifting is the closest one can get to (2) in 'product-free' type languages, i.e. type languages where the role of the product operator (generally left implicit) is restricted to glue together types on the left-hand side of the arrow. Items (7) to (11) mark the transition to **L**: their derivation involves the structural postulate of associativity for $\bullet$. Rule (12) is characteristic for systems with a commutative $\bullet$, **NLP** and **LP**. From (12) one immediately derives the collapse of the implications / and \, (13). As a result of this collapse, one gets variants of the earlier theorems obtained by substituting subtypes of the form A/B by $B\backslash A$ or vice versa. Examples are (14), an **NLP** variant of Lifting, or (15), an **LP** variant of Composition.

DISCUSSION: RULE-BASED VERSUS LOGIC BASED APPROACHES. The simple Lambek systems each have their merits and their limitations when it comes to grammatical analysis. As a grammar writer, one would like to exploit the inferential capacities of a combination of different systems. In rule-based frameworks, such as Combinatory Categorial Grammar (CCG, cf [44]), instances of the full scala of type transitions illustrated in (1)–(15) above indeed live together. In the logical setup adopted in this paper, such promiscuity has unpleasant consequences. As we have seen above, the Residuation laws capture the basic properties of the interpretation of the type-forming connectives. In the presence of Residuation, the introduction of theorems from a system with more relaxed resource management into a logic with a higher degree of structural discrimination instantly destroys sensitivity for the relevant structural parameter of the more discriminating logic. For example: **NL** has a hierarchically structured database which respects constituent structure. For cases of so-called non-constituent coordination, one would like to relax constituent structure. One could try to achieve this by adding Composition (or the Geach laws) to **NL**. But the addition of such postulates makes **NL** collapse into **L**: from Geach one easily obtains the unconditional Associativity postulate for $\bullet$ via Residuation. We leave this as an exercise for the reader. Similarly, it has been argued that an analysis of Dutch crossed dependencies requires the Mixed Composition laws. Again, the introduction of this **LP** theorem within an order-sensitive system such as **L** causes permutation collapse.

CCG avoids these problems by restricting the attention to a database of rule schemata without facing the semantic consequences of their combination. This route is not open to us if we want to leave intact the idea of a grammar logic, i.e. a semantically interpreted grammar formalism. In the following sections we develop a logical framework supporting mixed styles of categorial inference. Structural collapse is avoided by moving to a multimodal architecture which is better adapted to deal with the fine-structure of linguistic composition.

2 Residuation in mixed logics

2.1 Mixed inference: multimodal systems

Our first generalizing move is from a unimodal setup, where the type-forming connectives are interpreted in terms of a single notion of linguistic composition, to a *multimodal* architecture.[1] The objective here is to combine the virtues of the distinct logics we have discussed before in one multimodal system, and at the same time to overcome the limitations of the individual systems in isolation. Each of the component logics has its own specific resource management properties: when combining the different logics, we have to take care that these individual characteristics are left intact. We

[1] See [37, 38, 36] for linguistic applications of the multimodal style of inference.

do this by relativizing linguistic composition to specific resource management *modes*. But also, we want the inferential capacity of the combined logic to be more than the sum of the parts. The extra expressivity comes from *interaction postulates* that hold when different modes are in construction with one another.

On the syntactic level, the category formulae for the multimodal system are defined inductively on the basis of a set of category atoms $\mathcal{A}$ and a set of indices I as shown below. We refer to the $i \in I$ as *resource management modes*, or modes for short.

$$\mathcal{F} ::= \mathcal{A} \mid \mathcal{F}/_i\mathcal{F} \mid \mathcal{F} \bullet_i \mathcal{F} \mid \mathcal{F}\backslash_i\mathcal{F}$$

The semantics for the mixed language is a straightforward generalisation of frame semantics for the simple systems. Rather than interpret multiplicative connectives in terms of *one* privileged notion of linguistic composition, we throw different forms of linguistic composition together and interpret in multimodal frames $\langle W, \{R_i^3\}_{i\in I}\rangle$. A valuation on a frame respects the structure of the complex types in the familiar way, interpreting each of the modes $i \in I$ with its own accessibility relation.

$$\begin{array}{lcl} v(A \bullet_i B) & = & \{x \mid \exists y \exists z[R_i xyz \ \& \ y \in v(A) \ \& \ z \in v(B)]\} \\ v(C/_iB) & = & \{y \mid \forall x \forall z[(R_i xyz \ \& \ z \in v(B)) \Rightarrow x \in v(C)]\} \\ v(A\backslash_i C) & = & \{z \mid \forall x \forall y[(R_i xyz \ \& \ y \in v(A)) \Rightarrow x \in v(C)]\} \end{array}$$

We can present the multimodal logic axiomatically or in Gentzen style. In the axiomatic presentation, we have the familiar residuation pattern now relativized to resource management modes:

$$A \rightarrow C/_iB \quad \textit{iff} \quad A \bullet_i B \rightarrow C \quad \textit{iff} \quad B \rightarrow A\backslash_i C$$

In sequent presentation, each residuated family of multiplicatives $\{/_i, \bullet_i, \backslash_i\}$ has a matching structural connective, again relativized to resource management modes. Antecedent terms are inductively defined as $\mathcal{T} ::= \mathcal{F} \mid (\mathcal{T}, \mathcal{T})^i$. Logical rules insist that use and proof of connectives respect the resource management modes. The explicit construction of the antecedent database in terms of structural connectives derives directly from Belnap's [7] work on Display Logic, where it serves exactly the same purpose as it does here, viz. to combine logics with different resource management regimes. [2] In addition, the mode information makes it possible to distinguish distinct forms of linguistic composition with the same resource management properties. For an example, see [36] where the product is split up in a left-headed $\bullet_l$ and a right-headed $\bullet_r$, introducing a dimension of *dependency* structure next to the dimensions of precedence and dominance.

The multimodal Gentzen rules for the connectives are presented below. The Axiom sequent and Cut rule remain unchanged — they have no mode restrictions.

$$[\mathrm{R}/_i]\frac{(\Gamma, B)^i \Rightarrow A}{\Gamma \Rightarrow A/_iB} \qquad \frac{\Gamma \Rightarrow B \qquad \Delta[A] \Rightarrow C}{\Delta[(A/_iB, \Gamma)^i] \Rightarrow C}[\mathrm{L}/_i]$$

$$[\mathrm{R}\backslash_i]\frac{(B, \Gamma)^i \Rightarrow A}{\Gamma \Rightarrow B\backslash_i A} \qquad \frac{\Gamma \Rightarrow B \qquad \Delta[A] \Rightarrow C}{\Delta[(\Gamma, B\backslash_i A)^i] \Rightarrow C}[\mathrm{L}\backslash_i]$$

$$[\mathrm{L}\bullet_i]\frac{\Gamma[(A, B)^i] \Rightarrow C}{\Gamma[A \bullet_i B] \Rightarrow C} \qquad \frac{\Gamma \Rightarrow A \qquad \Delta \Rightarrow B}{(\Gamma, \Delta)^i \Rightarrow A \bullet_i B}[\mathrm{R}\bullet_i]$$

In addition to the residuation inferences which are shared by all resource management modes, we now have mode-specific structural options. In axiomatic style, they take the form of structural postulates; in sequent presentation, we have the corresponding structural rules. As an illustration, see the structural postulates/rules for a commutative mode c. In the semantics the R_c interpreting this connective will be constrained to satisfy $(\forall x, y, z \in W)\ R_c xyz \Rightarrow R_c xzy$.

$$A \circ_c B \longleftrightarrow B \circ_c A \qquad \frac{\Gamma[(\Delta_2, \Delta_1)^c] \Rightarrow A}{\Gamma[(\Delta_1, \Delta_2)^c] \Rightarrow A}[\mathrm{P}]$$

[2] See [30, 49] for recent applications in a modal setting. More recently, the same idea has been introduced in Linear Logic in [18].

Multimodal communication. What we have done so far is simply put together the individual systems discussed before in isolation. This is enough to gain combined access to the inferential capacities of the component logics, and one avoids the unpleasant collapse into the least discriminating logic that results from combining logics without taking into account the mode specifications, cf our discussion of CCG in §1. But as things are, the borders between the constituting logics in our multimodal setting are still hermetically closed. Let us turn then to the question of multimodal communication.

Communication between modes i, j is obtained via interaction postulates with the corresponding frame conditions linking the interpretation of the composition relations R_i and R_j. Frame conditions 'mixing' distinct modes i, j allow for the statement of distributivity principles regulating the interaction between R_i, R_j.

Among the multimodal interaction principles, we distinguish cases of weak and strong distributivity. The weak distributivity principles do not affect the multiplicity of the linguistic resources. They allow for the realization of mixed associativity or commutativity laws as the multimodal counterparts of the unimodal versions discussed above. Interaction principles of the strong distributivity type duplicate resources, thus giving access to mode-restricted forms of Contraction.

Weak distributivity. Consider first interaction of the weak distributivity type. Below one finds principles of *mixed* associativity and commutativity. Instead of the global associativity and commutativity options characterizing **L**, **NLP**, **LP**, we can now formulate constrained forms of associativity/commutativity, restricted to the situation where modes i and j are in construction. (Symmetric duals can be added with the i mode distributing from the right, and one can split up the two-directional inferences in their one-directional components, if so required.)

$$\begin{array}{ll} MP: & A \bullet_i (B \bullet_j C) \longleftrightarrow B \bullet_j (A \bullet_i C) \\ MA: & A \bullet_i (B \bullet_j C) \longleftrightarrow (A \bullet_i B) \bullet_j C \end{array}$$

The interaction postulates correspond to the frame conditions one finds below ($\forall xyzu \in W$):

$$\begin{array}{ll} MP: & \exists t (R_i uxt \ \&\ R_j tyz) \Leftrightarrow \exists t' (R_j uyt' \ \&\ R_i t'xz) \\ MA: & \exists t (R_i uxt \ \&\ R_j tyz) \Leftrightarrow \exists t' (R_j ut'z \ \&\ R_i t'xy) \end{array}$$

And they manifest themselves in structural rules in Gentzen presentation.

$$\frac{\Gamma[(\Delta_2, (\Delta_1, \Delta_3)^i)^j] \Rightarrow A}{\Gamma[(\Delta_1, (\Delta_2, \Delta_3)^j)^i] \Rightarrow A}[\mathrm{MC}] \qquad \frac{\Gamma[((\Delta_1, \Delta_2)^i, \Delta_3)^j] \Rightarrow A}{\Gamma[(\Delta_1, (\Delta_2, \Delta_3)^j)^i] \Rightarrow A}[\mathrm{MA}]$$

For linguistic application of these general postulates, we refer to the analysis of Dutch Verb Raising in [38, 36], where it is shown that a multimodal variant of the CCG 'mixed composition' law — which in the absence of mode constraints causes collapse of **L** into **LP**, as we saw above — is in fact a theorem in combined logics with the MP/MA interaction principles. An example is given below. The verb cluster *wil lezen* is obtained in terms of a bimodal interaction principle relating, in this particular case, the right-headed dependency mode $\bullet_r$ and the pre-head Dutch head adjunction mode $\bullet_w$. The former characterizes the head-final clausal structure of Dutch, and is used in the typing of the verb *lezen* as $np\backslash_r iv$. The latter allows the verb-raising trigger *wil*, typed $vp/_w iv$, to form a verb cluster together with the head of its iv infinitival complement.

(dat Marie) boeken (wil lezen)
(that Mary) books (wants read)/that M. wants to read books
$vp/_w iv \Rightarrow (np\backslash_r vp)/_w(np\backslash_r iv)$

Schematically, in 'Geach' version, we have the following derivation. Notice that the order sensitivity of the individual modes $\bullet_r$ and $\bullet_w$ is respected: the valid forms of mixed composition form a subset of the composition laws derivable within unimodal **LP**. The principles of Directional Consistency and Directional Inheritance, introduced as theoretical primitives in the rule-based setting of CCG, can be seen here to follow automatically from the individual resource management properties of the modes involved and the distributivity principle governing their communication.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{C \Rightarrow C \quad B \Rightarrow B}{(C, C\backslash_r B)^r \Rightarrow B}\ \backslash_r L \quad A \Rightarrow A}{(A/_w B, (C, C\backslash_r B)^r)^w \Rightarrow A}\ /_w L}{(C, (A/_w B, C\backslash_r B)^w)^r \Rightarrow A}\ MP}{(A/_w B, C\backslash_r B)^w \Rightarrow C\backslash_r A}\ \backslash_r R}{A/_w B \Rightarrow (C\backslash_r A)/_w (C\backslash_r B)}\ /_w R$$

Interaction principles: strong distributivity. As remarked above, the weak distributivity principles MP, MA keep us within the family of resource neutral logics: they do not affect the multiplicity of the resources in a configuration. Strong distributivity principles are not resource neutral: they duplicate resources. As an example, consider the interaction principle MC below, which strongly distributes mode j over mode i thus copying a C datum. Rather than introducing global Contraction, this interaction principle allows for a constrained form of copying, restricted to the case where modes i and j are in construction. The computation of the relevant frame condition and structural rule will be familiar by now. (As with the Mixed Associativity/Commutativity principles, a symmetric case for distributivity from the left can be added straightforwardly.)

$$MC: \quad (A \bullet_i B) \bullet_j C \rightarrow (A \bullet_j C) \bullet_i (B \bullet_j C)$$

$$(R_i txy \ \&\ R_j utz) \Rightarrow \exists t' \exists t'' (R_j t'xz \ \&\ R_j t''yz \ \&\ R_i ut't'')$$

$$\dfrac{\Gamma[((\Delta_1, \Delta_3)^j, (\Delta_2, \Delta_3)^j)^i] \Rightarrow A}{\Gamma[((\Delta_1, \Delta_2)^i, \Delta_3)^j] \Rightarrow A}\ MC$$

Grammatical inference requires restricted access to Contraction for the analysis of parasitic gap constructions (†) and coordination of incomplete material (‡). In the former case, one would like the abstractor associated with the *wh* element to bind multiple occurrences of the same variable. Such multiple binding is beyond the scope of resource sensitive inference. In the (‡) example, sentential coordination is generalized to the coordination of sentences missing an object — a process which again requires the copying of resources.

(†) Which books did John (file _ without reading _)
(‡) John loves but Mary hates beans

In the rule-based framework of CCG, parasitic gaps are handled by means of the combinator **S** which is introduced as a primitive for this purpose, cf. [45].

$$\mathbf{S}: \quad A/C, (A\backslash B)/C \Rightarrow B/C$$

In a unimodal setting the **S** combinator in combination with Residuation causes disaster. In the multimodal framework presented here, a mode-restricted form of the **S** combinator can be derived from the strong distributivity principle discussed above. In the Gentzen proof below, we give the relevant instance for the derivation of the example sentence (instantiate $A/_j C$ as $vp/_j np$ for *file*, and $(A\backslash_i B)/_j C$ as $(vp\backslash_i vp)/_j np$ for *without reading*). Mode j here would be the default mode by which the transitive verbs *file* and *read* consume their direct objects; the combination of the vp adjunct *without reading* _ with the vp it modifies is given in terms of mode i, the 'parasitic' mode which licenses the secondary gap depending on the primary one, the argument of *file*.

$$\dfrac{\dfrac{\dfrac{\&c}{((A/_j C, C)^j, ((A\backslash_i B)/_j C, C)^j)^i \Rightarrow B}}{((A/_j C, (A\backslash_i B)/_j C)^i, C)^j \Rightarrow B}\ MC}{(A/_j C, (A\backslash_i B)/_j C)^i \Rightarrow B/_j C}\ /_j R$$

Notice that in the case of Right-Node Raising (‡), we can appeal to the same interaction principle to derive the non-constituent coordination from sentential coordination, provided the incomplete conjuncts are put together in the duplicating mode i — structural information that can be projected straightforwardly from the type assignment to the conjunction particle, e.g. $(s\backslash_i s)/s$. This suggests a multimodal treatment of generalized coordination in terms of restricted Contraction — a topic that must be left for future research.

2.2 Mixed inference: combining 1-ary and 2-ary families

What we have studied so far is the language of binary connectives — a language well adapted to talk about forms of linguistic composition where two resources are put together. But sometimes one would like to attribute particular resource management properties to individual resources, rather than to configurations of resources. The required expressivity can be introduced by extending the type language with *unary* connectives decorating individual formulae.

Unary connectives entered the linguistic discussion in 1990 in the work of a number of Edinburgh researchers, cf. [3]. Taking their inspiration from the '!' operator of Linear Logic which licenses Contraction and Weakening for '!' decorated formulae, these authors have introduced structural modalities — unary operators providing controlled access to linguistically relevant structural options, such as Permutation. In the recent literature one finds a panoply of unary operators in addition to the binary multiplicatives. Apart from the structural modalities, we can mention the 'domain modalities' of Morrill and Hepple, identifying semantic intensionality domains ([39]) or purely syntactic domains of locality ([23]), the 'bracket operators' $[], []^{-1}$ of Morrill [40, 41], implementing locality domains in a different way, or the $\diamond$ operator of Morrill [40], declaring argument positions as licensing extraction.

Our aim in this section is to develop a general framework that will naturally accommodate the different proposals for unary operators while at the same time providing more fine-grained notions of resource control. The key concept, again, is residuation. We extend the language of binary multiplicatives with a unary pair of residual operators $\Diamond, \Box$ and establish a number of elementary logical results for the extended language. Parallel to our treatment of the binary multiplicatives $/, \bullet, \backslash$ in the previous section, we start from the most discriminating system, i.e. the pure logic of residuation for $\Diamond, \Box$. By gradually adding structural postulates, we obtain versions of these unary operators with a coarser resource management regime. And where the linguistic applications require this, we can put together different variants in a multimodal logic $\langle i \rangle, [i]^{\downarrow}$.

Our agenda for this section is given below. Items (1) and (3), (4) closely follow Lambek's [33] treatment of the binary multiplicatives.

1. Axiomatic ('combinator') presentations of the pure logic of residuation for $\Diamond, \Box$.
2. Soundness and completeness via the Došen canonical model construction.
3. Gentzen presentation, equivalence between the axiomatic and the Gentzen presentation.
4. Cut elimination for the Gentzen presentation. Decidability, subformula property.
5. Structural postulates $T, 4, K$. Items (1)–(4) for the systems with a choice from $\mathcal{P}(\{T, 4, K\})$.

Residuation: n-ary generalisation. The concept of residuation, which as we saw above lies at the heart of categorial type logic, arises in the study of order-preserving mappings. In order to widen our framework from binary to n-ary families of residuated connectives, we first have a brief look at the general algebraic concept. (In [37] the reader can find a more thorough treatment with reference to the source material, such as [15, 8]).

Let $\mathbf{A} = (A, \leq_A)$ and $\mathbf{B} = (B, \leq_B)$ be partially ordered sets. Consider a pair of functions $f : A \mapsto B$ and $g : B \mapsto A$. The pair (f, g) is called *residuated* if the inequalities of $(\star)$ hold.

Alternatively, a pair of functions (f,g) is characterized as residuated by requiring f and g to be isotone (†), and by having the composition of the functions satisfy the inequalities of (‡).

$$(\star) \qquad fx \leq_B y \quad \text{iff} \quad x \leq_A gy$$

$$(\dagger) \quad \text{if } x \leq_A y \ (x \leq_B y) \text{ then } fx \leq_B fy \ (gx \leq_A gy)$$
$$(\ddagger) \qquad fgx \leq_B x, \quad x \leq_A gfx$$

Dunn's papers on 'gaggle theory' ([13, 14]) provide an excellent survey of the many guises under which Residuation presents itself in (intuitionistic, modal, relevance, dynamic, temporal, linear,...) logic, and in Lambek style type logics. Indeed, the pairs of connectives $(\bullet, /)$ and $(\bullet, \backslash)$ are easily recognized as the binary incarnations of the notion of residuation just defined for the case of unary operations f, g. Interpret the partially ordered set $\mathbf{A}$ $(= \mathbf{B})$ as the set of type formulae $\mathcal{F}$, ordered by derivability $\rightarrow$ (i.e. set-theoretic inclusion, semantically). For the *right residual* pair $(\bullet, /)$ we can read f as $- \bullet B$ and g as $-/B$, i.e. the product and division operations indexed by some fixed type B. The defining biconditional $fx \leq y$ iff $x \leq gy$ then becomes $A \bullet B \rightarrow C$ iff $A \rightarrow C/B$. Similarly for the *left residual* pair $(\bullet, \backslash)$, where we read f as $A \bullet -$ and g as $A\backslash -$, and obtain $A \bullet B \rightarrow C$ iff $B \rightarrow A \backslash C$.

The concept of residuation can be readily generalized to the case of n-ary connectives, as is shown in [13] in the general logical setting. Discussion of such generalizations for categorial type logics can be found in [10] and [37]. In the context of our Kripke style frame semantics, we now find n-ary products interpreted via $n+1$-ary accessibility relations. These products have a residual implication for each of their n factors. Let us write $f_\bullet(A_1, \ldots, A_n)$ for the product and $f^i_\rightarrow(A_1, \ldots, A_n)$ for the i-th place residual. And define $R^{-i} y_i y_1, \ldots, x, \ldots, y_n$ iff $Rxy_1, \ldots, y_i, \ldots, y_n$ to facilitate the statement of the interpretation clauses. We require the valuation for the n-ary families to exhibit the familiar pattern: existential closure of a conjunctive statement for the product, universal closure of disjunctions for the residual implications.

$$v(f_\bullet(A_1, \ldots, A_n)) = \{x \mid \exists y_1 \ldots y_n (Rxy_1 \ldots y_n \ \& \ y_1 \in v(A_1) \ \& \ \ldots \& \ y_n \in v(A_n)\}$$
$$v(f^i_\rightarrow(A_1, \ldots, A_n)) = \{x \mid \forall y_1 \ldots y_n ((R^{-i} xy_1 \ldots y_n \ \& \ y_{j(j \neq i)} \in v(A_j)) \Rightarrow y_i \in v(A_i)\}$$

Given such an interpretation for the compound formulae, the residuation laws are realized in the form shown below.

$$f_\bullet(A_1, \ldots, A_n) \rightarrow B \Longleftrightarrow A_i \rightarrow f^i_\rightarrow(A_1, \ldots, A_{i-1}, B, A_{i+1}, \ldots, A_n)$$

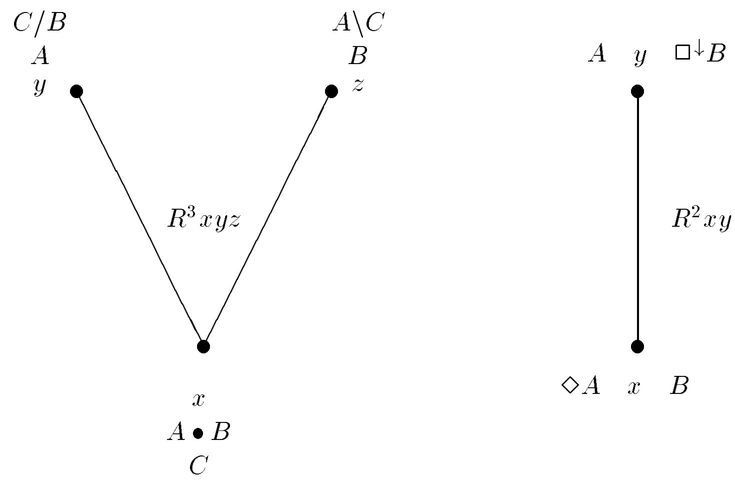
UNARY RESIDUATED PAIRS. Let us focus now on the case of *unary* connectives. Consider a residuated pair of connectives $\Diamond$, $\Box$ for which the defining residuation inference $fx \leq y$ iff $x \leq gy$ takes the form (†), given the interpretation clauses shown below.

$$(\dagger) \quad \Diamond A \rightarrow B \quad \Longleftrightarrow \quad A \rightarrow \Box B$$

$$v(\Diamond A) = \{x \mid \exists y (Rxy \ \wedge \ y \in v(A)\}$$
$$v(\Box A) = \{x \mid \forall y (Ryx \ \Rightarrow y \in v(A)\}$$

The valuation for the $\Diamond, \Box$ formulae has the required properties for residuation to arise: existential closure of a conjunctive statement for $\Diamond$, universal closure of a disjunction for the residual $\Box$. Note carefully that the interpretation of $\Diamond$ and $\Box$ moves you in opposite directions along the R^2 accessibility relation. The typographically non-standard choice for the diamond operator is there to remind you of this fact.

Figure 1 may clarify the relation between the unary and the binary residuated pairs of connectives. In the case of $\bullet$ we make an existential move along the branching accessibility relation R^3. In the case of $\Diamond$ we make an existential move in the same direction, this time for a non-branching accessibility relation. In both cases, universal moves in the opposite direction bring you back to the point of origin.



$$A \to C/B \Leftrightarrow A \bullet B \to C$$
$$A \bullet B \to C \Leftrightarrow B \to A\backslash C$$

$$\Diamond A \to B \Leftrightarrow A \to \Box B$$

Figure 1: Kripke graphs: binary and unary multiplicatives

How shall we interpret R^2 in this caseΓ If we were talking about temporal organization, $\Diamond$ and $\Box$ could be interpreted as future possibility and past necessity, respectively. But in our grammatical application, R^2 just like R^3 is to be interpreted in terms of structural composition. Where a ternary configuration $(xyz) \in R^3$ interpreting the product connective abstractly represents putting together the components y and z into a structured configuration x in the manner indicated by R^3, a binary configuration $(xy) \in R^2$ interpreting the unary $\Diamond$ can be seen as the construction of the sign x out of a structural component y in terms of the building instructions referred to by R^2.[3]

Axiomatic presentations. Putting together the binary and the unary families of connectives we can now consider the mixed language

$$\mathcal{F} ::= \mathcal{A} \mid \mathcal{F}/\mathcal{F} \mid \mathcal{F} \bullet \mathcal{F} \mid \mathcal{F}\backslash\mathcal{F} \mid \Diamond\mathcal{F} \mid \Box\mathcal{F}$$

$$\mathbf{id}_A : A \to A \qquad \frac{f : A \to B \quad g : B \to C}{g \circ f : A \to C}$$

$$\frac{f : \Diamond A \to B}{\mu(f) : A \to \Box B} \qquad \frac{g : A \to \Box B}{\mu^{-1}(g) : \Diamond A \to B}$$

$$\frac{f : A \bullet B \to C}{\beta(f) : A \to C/B} \qquad \frac{g : A \to C/B}{\beta^{-1}(g) : A \bullet B \to C}$$

$$\frac{f : A \bullet B \to C}{\gamma(f) : B \to A\backslash C} \qquad \frac{g : B \to A\backslash C}{\gamma^{-1}(g) : A \bullet B \to C}$$

Figure 2: Axiomatization: Lambek style

In Figures 2 and 3 we juxtapose two equivalent axiomatic presentations of the pure logic of residuation for the extended language. The Lambek style presentation of Fig. 2 is based on the residuation

[3] For an 'additive' alternative to our 'multiplicative' view on unary operators, see [46].

$$\mathbf{id}_A : A \to A \qquad \frac{f : A \to B \quad g : B \to C}{g \circ f : A \to C}$$

$$\begin{array}{rr} \mathbf{unit}_\Box : \Diamond\Box A \to A & \mathbf{co\text{-}unit}_\Box : A \to \Box\Diamond A \\ \mathbf{unit}_/ : A/B \bullet B \to A & \mathbf{co\text{-}unit}_/ : A \to (A \bullet B)/B \\ \mathbf{unit}_\backslash : B \bullet B\backslash A \to A & \mathbf{co\text{-}unit}_\backslash : A \to B\backslash(B \bullet A) \end{array}$$

$$\frac{f : A \to B}{(f)^\Diamond : \Diamond A \to \Diamond B} \qquad \frac{f : A \to B}{(f)^\Box : \Box A \to \Box B}$$

$$\frac{f : A \to B \quad g : C \to D}{f \cdot g : A \bullet C \to B \bullet D}$$

$$\frac{f : A \to B \quad g : C \to D}{f/g : A/D \to B/C} \qquad \frac{f : A \to B \quad g : C \to D}{g\backslash f : D\backslash A \to C\backslash B}$$

Figure 3: Axiomatization: Došen style

inferences. The Došen style presentation of Fig. 3 uses the alternative way of characterizing a pair of residual operations f, g in terms of Isotonicity ($\star$) and the inequalities ($\star\star$) for the compositions fg, gf.

$$(\star) \quad x \leq y \Rightarrow fx \leq fy, gx \leq gy \qquad (\star\star) \quad fgx \leq x, \quad x \leq gfx$$

In this presentation, the **unit**, **co-unit** combinators are primitive type transitions, recursively generalized via the Isotonicity rules of inference (Antitonicity for the negative subtype of implications $/, \backslash$).

For the $/, \bullet, \backslash$ fragment, we know the two deductive presentations are equivalent, cf. Lambek [33] for one direction, Došen [12] for the other. We take the Lambek presentation as our starting point here, and show for the extended system how from μ, μ^{-1} we obtain the alternative axiomatisation in terms of Isotonicity and the inequalities for the compositions $\Diamond\Box$ and $\Box\Diamond$ (Term decoration for the right column left to the reader.)

$$\frac{\mathbf{id}_{\Box A} : \Box A \to \Box A}{\mu^{-1}(\mathbf{id}_{\Box A}) : \Diamond\Box A \to A} \qquad \frac{\Diamond A \to \Diamond A}{A \to \Box\Diamond A}$$

$$\frac{\dfrac{f : A \to B \quad \dfrac{\mathbf{id}_{\Diamond B} : \Diamond B \to \Diamond B}{\mu(\mathbf{id}_{\Diamond B}) : B \to \Box\Diamond B}}{\mu(\mathbf{id}_{\Diamond B}) \circ f : A \to \Box\Diamond B}}{\mu^{-1}(\mu(\mathbf{id}_{\Diamond B}) \circ f) : \Diamond A \to \Diamond B} \qquad \frac{\dfrac{\dfrac{\Box A \to \Box A}{\Diamond\Box A \to A} \quad A \to B}{\Diamond\Box A \to B}}{\Box A \to \Box B}$$

COMPLETENESS. For the $\mathcal{F}(/, \bullet, \backslash)$ fragment, [11] shows that **NL** is complete with respect to the class of all ternary models, and **L**, **NLP**, **LP** with respects to the classes of models satisfying the frame constraints for the relevant packages of structural postulates. The completeness result extends unproblematically to the language enriched with $\Diamond, \Box$ as soon as one realizes that $\Diamond$ can be seen as a 'truncated' product and $\Box$ its residual implication.

Define the canonical model for mixed (2,3) frames as $\mathcal{M} = \langle W, R^2, R^3, v\rangle$, where

- W is the set of formulae $\mathcal{F}(/, \bullet, \backslash, \Diamond, \Box)$
- $R^3(A, B, C)$ iff $\vdash A \to B \bullet C$, $R^2(A, B)$ iff $\vdash A \to \Diamond B$
- $A \in v(p)$ iff $\vdash A \to p$.

The Truth Lemma then states that, for any formula ϕ, $\mathcal{M}, A \models \phi$ iff $\vdash A \rightarrow \phi$. Now suppose $v(A) \subseteq v(B)$ but $\not\vdash A \rightarrow B$. If $\not\vdash A \rightarrow B$ with the canonical valuation on the canonical frame, $A \in v(A)$ but $A \notin v(B)$ so $v(A) \not\subseteq v(B)$. Contradiction.

We have to check the Truth Lemma for the new compound formulae $\Diamond A$, $\Box A$. Below the direction that requires a little thinking.

($\Diamond$) Assume $A \in v(\Diamond B)$. We have to show $\vdash A \rightarrow \Diamond B$. $A \in v(\Diamond B)$ implies $\exists A'$ such that R^2AA' and $A' \in v(B)$. By inductive hypothesis, $\vdash A' \rightarrow B$. By Isotonicity for $\Diamond$ this implies $\vdash \Diamond A' \rightarrow \Diamond B$. We have $\vdash A \rightarrow \Diamond A'$ by (Def R^2) in the canonical frame. By Transitivity, $\vdash A \rightarrow \Diamond B$.

($\Box$) Assume $A \in v(\Box B)$. We have to show $\vdash A \rightarrow \Box B$. $A \in v(\Box B)$ implies that $\forall A'$ such that $R^2A'A$ we have $A' \in v(B)$. Let A' be $\Diamond A$. $R^2A'A$ holds in the canonical frame since $\vdash \Diamond A \rightarrow \Diamond A$. By inductive hypothesis we have $\vdash A' \rightarrow B$, i.e. $\vdash \Diamond A \rightarrow B$. By Residuation this gives $\vdash A \rightarrow \Box B$.

Gentzen calculus. Following the agenda set out in §1 for the binary connectives, we now introduce Gentzen sequent rules for the connectives $\Diamond, \Box$. In the Appendix, we show that the Genzten rules are equivalent to the deductive presentation, and that the Gentzen presentation allows Cut Elimination, with its pleasant corollaries of Decidability and the Subformula property.

In order to present Gentzen calculus for the extended type language, we need an n-ary structural operator for every family of n-ary logical operators: binary $(\cdot,\cdot)$ for the family $/, \bullet, \backslash$, and unary $(\cdot)^\diamond$ for the family $\Diamond, \Box$. Corresponding to the formula language $\mathcal{F}$ we have a language of terms $\mathcal{T}$ (structured configurations of formulae).

$$\mathcal{F} ::= \mathcal{A} \mid \mathcal{F}/\mathcal{F} \mid \mathcal{F} \bullet \mathcal{F} \mid \mathcal{F}\backslash\mathcal{F} \mid \Diamond\mathcal{F} \mid \Box\mathcal{F}$$

$$\mathcal{T} ::= \mathcal{F} \mid (\mathcal{T},\mathcal{T}) \mid (\mathcal{T})^\diamond$$

As before, sequents are pairs (Γ, A), $\Gamma \in \mathcal{T}$, $A \in \mathcal{F}$, written $\Gamma \Rightarrow A$. We have Belnap-style antecedent punctuation, with for $\Diamond, \Box$ the *unary* structural connective $(\cdot)^\diamond$ matching the unary logical connectives. Below the rules of use $[\Diamond L]$, $[\Box L]$ and the rules of proof $[\Diamond R]$, $[\Box R]$ for the new connectives.

$$\frac{\Gamma \Rightarrow A}{(\Gamma)^\diamond \Rightarrow \Diamond A}\ \Diamond R \qquad \frac{\Gamma[(A)^\diamond] \Rightarrow B}{\Gamma[\Diamond A] \Rightarrow B}\ \Diamond L$$

$$\frac{(\Gamma)^\diamond \Rightarrow A}{\Gamma \Rightarrow \Box A}\ \Box R \qquad \frac{\Gamma[A] \Rightarrow B}{\Gamma[(\Box A)^\diamond] \Rightarrow B}\ \Box L$$

Figure 4: $\Diamond, \Box$: Gentzen rules

As we remarked above, $\Diamond$ and $\Box$ can be seen as truncated forms of product and implication. It may be helpful to compare the $\Diamond$ rules with the rules for $\bullet$, and the $\Box$ rules with the rules for an implication, say $/$.

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

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

Illustration: Residuation laws. For a simple illustration of the Gentzen style presentation we check the compositions $\Diamond\Box$ and $\Box\Diamond$ (cf Application, $fgx \leq x$, Co-Application, $x \leq gfx$). Below their cut-free Gentzen derivations.

$$\cfrac{\cfrac{A \Rightarrow A}{(\Box A)^\diamond \Rightarrow A}\ \Box L}{\Diamond\Box A \Rightarrow A}\ \Diamond L \qquad \cfrac{\cfrac{A \Rightarrow A}{(A)^\diamond \Rightarrow \Diamond A}\ \Diamond R}{A \Rightarrow \Box\Diamond A}\ \Box R$$

Structural postulates. What we have discussed so far is the pure logic of residuation for the unary family $\Diamond, \Box$. In the previous section we saw how by imposing conditions ASS, COMM or their combination on ternary frames, one generates the landscape **NL, L, NLP, LP** with completeness results for the relevant classes of frames (Došen). Along the same lines, we can develop the substructural landscape for the unary family $\Diamond, \Box$ and its binary accessibility relation R^2, and for the mixed R^2, R^3 system.[4]

The following structural postulates constrain R^2 to be transitive (4), or reflexive (T). Communication between R^2 and R^3 can be established via the strong distributivity postulate K, which distributes unary $\Diamond$ over both components of a binary $\bullet$, or, in a more constrained way, via the weak distributivity postulates $K1$, $K2$, where $\Diamond$ selects the left or right subtype of a product.

$$
\begin{array}{rc}
4: & \Diamond\Diamond A \to \Diamond A \\
T: & A \to \Diamond A \\
K1: & \Diamond(A \bullet B) \to \Diamond A \bullet B \\
K2: & \Diamond(A \bullet B) \to A \bullet \Diamond B \\
K: & \Diamond(A \bullet B) \to \Diamond A \bullet \Diamond B
\end{array}
$$

Below the correponding frame conditions ($\forall x, y, z, w \in W$).

$$
\begin{array}{rc}
4: & (Rxy \ \& \ Ryz) \Rightarrow Rxz \\
T: & Rxx \\
K(1,2): & (Rwx \ \& \ Rxyz) \Rightarrow \exists y'(Ry'y \ \& \ Rwy'z) \ \vee \ \exists z'(Rz'z \ \& \ Rwyz') \\
K: & (Rwx \ \& \ Rxyz) \Rightarrow \exists y' \exists z'(Ry'y \ \& \ Rz'z \ \& \ Rwy'z')
\end{array}
$$

We have shown above how the Došen completeness result for **NL** can be generalized to the pure residuation logic for $\Diamond, \Box$. Kurtonina [31] moreover shows that structural postulates like $4, T, K(1,2)$ have the appropriate form for an extended Sahlqvist completeness result: the pure residuation logic augmented with these postulates is frame complete for the first-order frame condition corresponding to the postulate in question.

Structural rules. The structural rules below translate the postulates $T, 4, K1, K2, K$ from the formula level to the term level. (In the Appendix, we prove equivalence between the rule and the postulate versions, and show that the Gentzen formulation allows cut-free proof search.)

$$
\frac{\Gamma[(\Delta)^\diamond] \Rightarrow A}{\Gamma[((\Delta)^\diamond)^\diamond] \Rightarrow A}\ 4 \qquad \frac{\Gamma[(\Delta)^\diamond] \Rightarrow A}{\Gamma[\Delta] \Rightarrow A}\ T
$$

$$
\frac{\Gamma[((\Delta_1)^\diamond, \Delta_2)] \Rightarrow A}{\Gamma[((\Delta_1, \Delta_2))^\diamond] \Rightarrow A}\ K1 \quad \frac{\Gamma[((\Delta_1)^\diamond, (\Delta_2)^\diamond)] \Rightarrow A}{\Gamma[((\Delta_1, \Delta_2))^\diamond] \Rightarrow A}\ K \quad \frac{\Gamma[(\Delta_1, (\Delta_2)^\diamond)] \Rightarrow A}{\Gamma[((\Delta_1, \Delta_2))^\diamond] \Rightarrow A}\ K2
$$

Structural postulates: universal variant. In our discussion of structural postulates for $\bullet$, we have seen that we can express Associativity, Commutativity either via a $\bullet$ postulate, or via implicational postulates, if we prefer to keep the language product-free. In a similar vein we could have presented $T, 4, K$ in their $\Box$ forms:

$$
\begin{array}{rc}
4\Box: & \Box A \to \Box\Box A \\
T\Box: & \Box A \to A \\
K\Box/: & \Box(A/B) \to \Box A/\Box B \\
K\Box\backslash: & \Box(B\backslash A) \to \Box B\backslash\Box A
\end{array}
$$

[4] For a related decomposition of the S4 '!' modality of Linear Logic, see [9, 19].

Below an illustration for the derivation of the universal variant $K\Box/$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B \quad A \Rightarrow A}{(A/B, B) \Rightarrow A}\ /L}{((\Box(A/B))^{\diamond}, (\Box B)^{\diamond}) \Rightarrow A}\ \Box L, \Box L}{((\Box(A/B), \Box B))^{\diamond} \Rightarrow A}\ K}{(\Box(A/B), \Box B) \Rightarrow \Box A}\ \Box R}{\Box(A/B) \Rightarrow \Box A/\Box B}\ /R$$

S4: Compilation of structural rules. We saw above that in the presence of Associativity for $\bullet$, we have a sugared Gentzen presentation where the structural rule is compiled away, and the binary sequent punctuation $(\cdot,\cdot)$ omitted. Analogously, for $\Box$ with the combination $KT4$ (i.e. $S4$), we have a sugared version of the Gentzen calculus, where the $KT4$ structural rules are compiled away, so that the unary $(\cdot)^{\diamond}$ punctuation can be omitted. Compare the following. (Notation $\dagger\Gamma$ for a term Γ of which the (pre)terminal subterms are of the form $\dagger A$. The $4(cut)$ step is a series of replacements of terminal $\Box A$ by $\Box\Box A$ via cuts depending on 4.)

$$\dfrac{\dfrac{\Gamma[A] \Rightarrow B}{\Gamma[(\Box A)^{\diamond}] \Rightarrow B}\ \Box L}{\Gamma[\Box A] \Rightarrow B}\ T \quad \rightsquigarrow \quad \dfrac{\Gamma[A] \Rightarrow B}{\Gamma[\Box A] \Rightarrow B}\ \Box L(S4)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Box\Gamma \Rightarrow A}{(\Box\Box)^{\diamond}\Gamma \Rightarrow A}\ \Box L}{(\Box)^{\diamond}\Gamma \Rightarrow A}\ 4(cut)}{(\Box\Gamma)^{\diamond} \Rightarrow A}\ K}{\Box\Gamma \Rightarrow \Box A}\ \Box R \quad \rightsquigarrow \quad \dfrac{\Box\Gamma \Rightarrow A}{\Box\Gamma \Rightarrow \Box A}\ \Box R(S4)$$

In the sugared version, we recognize the rules of use and proof for the domain modalities of [23, 39].

Multimodal generalization. We have presented the landscape of unary residuated operators from the perspective of one pair of connectives $\Diamond, \Box$. The move to a multimodal system where different families of unary operators live together and communicate (with families of the same or of different arity) via inclusion and interaction postulates is entirely straightforward. Below we give the mode restricted forms of the interpretation clauses, the residuation inferences, and the Gentzen rules.

$$\begin{array}{rcl} v(\Diamond_i A) & = & \{x \mid \exists y(R_i xy \ \wedge \ y \in v(A)\} \\ v(\Box_i A) & = & \{x \mid \forall y(R_i yx \ \Rightarrow y \in v(A)\} \end{array}$$

$$\Diamond_i A \to B \quad \textit{iff} \quad A \to \Box_i B$$

$$[\mathrm{R}\Diamond_i]\dfrac{\Gamma \Rightarrow A}{(\Gamma)^i \Rightarrow \Diamond_i A} \qquad \dfrac{\Gamma[(A)^i] \Rightarrow B}{\Gamma[\Diamond_i A] \Rightarrow B}[\mathrm{L}\Diamond_i]$$

$$[\mathrm{R}\Box_i]\dfrac{(\Gamma)^i \Rightarrow A}{\Gamma \Rightarrow \Box_i A} \qquad \dfrac{\Gamma[A] \Rightarrow B}{\Gamma[(\Box_i A)^i] \Rightarrow B}[\mathrm{L}\Box_i]$$

Discussion. To close this section, we briefly indicate where the various unary operators that have been proposed in the literature can be situated within the general landscape developed here.

At one end of the spectrum, the proposals that come closest to the pure logic of residuation for $\Diamond, \Box$ are Morrill's bracket operators. In one of their incarnations (the version of [40]) these operators are presented with the $[\Diamond L]$, $[\Diamond R]$ and $[\Box R]$ rules we have given above. The $[\Box L]$ rule of [40], however, is inappropriate for the pure residuation system: it derives the non-theorem $\Box\Diamond A \Rightarrow A$, next to the theorem $A \Rightarrow \Box\Diamond A$. On the semantic level, Morrill assumes the bracket operators to be interpreted in terms of a *functional* accessibility relation R^2 — an interpretation which imposes constraints on the allowable models which we have not assumed in our presentation. The linguistic applications of the bracket operators as markers of locality domains can be recast straightforwardly in terms of the more discriminating pure residuation logic for $\Diamond, \Box$ where no functionality constraints are imposed on R^2.

At the other end of the spectrum, we find the S4 domain modality of [23, 39], a universal modality $\Box$ which assumes the full set of postulates $KT4$. Adding modally controlled structural rules, we obtain the structural modalities of [3] and others. A crucial feature of these operators is the Reflexivity Postulate $\Box A \rightarrow A$: a modally marked resource $\Box A$ will at a certain point in the derivation be used as an ordinary datum of type A. Recall that we have presented the binary accessibility relation as a form of linguistic composition: Rxy holds in case x is the sign one obtains by augmenting y with the information added in the R^2 move. From this perspective, Reflexivity is an undesirable property, trivializing the R^2 augmentation. In the framework presented here, where we consider a residuated *pair* of modalities $\Diamond, \Box$ rather than a single modal operator $\Box$, we can capture the proof-theoretic behaviour of the S4 structural modalities *without* making Reflexivity (or Transitivity) assumptions about the R^2 accessibility relation. With a translation $(\Box A)^\sharp = \Diamond\Box(A)^\sharp$, the T and 4 postulates for $\Box$ become valid type transitions in the pure residuation system for $\Diamond, \Box$, as the reader can check.

$$\begin{array}{lll} T: & \Box A \rightarrow A & \Diamond\Box A \rightarrow A \\ 4: & \Box A \rightarrow \Box\Box A & \Diamond\Box A \rightarrow \Diamond\Box\Diamond\Box A \end{array}$$

For modally controlled structural rules, one can add restricted versions of the global rules relativized to $\Diamond$ contexts, cf. the modal version of the Permutation rule below.

$$\frac{\Gamma[((\Delta_2)^\diamond, \Delta_1)] \Rightarrow A}{\overline{\Gamma[(\Delta_1, (\Delta_2)^\diamond)] \Rightarrow A}}$$

3 Controlling resource management

What kind of expressivity do the unary connectives $\Diamond, \Box$ add to the categorial languageΓ We have seen in §1 and §2.1 that the binary connectives (or in general: n-ary connectives for $2 \leq n$) offer the logical vocabulary to talk about the composition of linguistic resources. With respect to grammatical composition the unary connectives play the role of *control* devices. Resource control can be approached from two perspectives:

> THE STATIC PERSPECTIVE. How can one control the *structural* properties of well-formed configurations of linguistic resources with respect to parameters such as word-order, constituent structure, dependency structureΓ
>
> THE DYNAMIC PERSPECTIVE. How can one control the way the linguistic resources are used in the *process* of putting together a well-formed structural configurationΓ

Structural control via $\Diamond, \Box$ is studied in [32]. We summarize the results in §3.1. In §3.2 we show how one can dynamically control the use of assumptions in the process of proof search via $\Diamond, \Box$ decoration.

3.1 Structural control

The structural parameters of precedence (word-order), dominance (constituent structure) and dependency generate a cube of resource-sensitive categorial type logics. From the pure logic of residuation **NL**, one obtains the familiar systems **L**, **NLP** and **LP** in terms of Associativity, Commutativity, and their combination. These systems occupy the upper plane of Figure 5. Each of these systems has a dependency variant, where the product is split up into a left-headed $\bullet_l$ and a right-headed $\bullet_r$ version. The dependency dimension was introduced in [36] where it is argued that the asymmetry between heads and the material they govern should be considered as orthogonal with respect to the function-argument asymmetry. The dependency systems decorate the vertices at the lower plane of Figure 5.

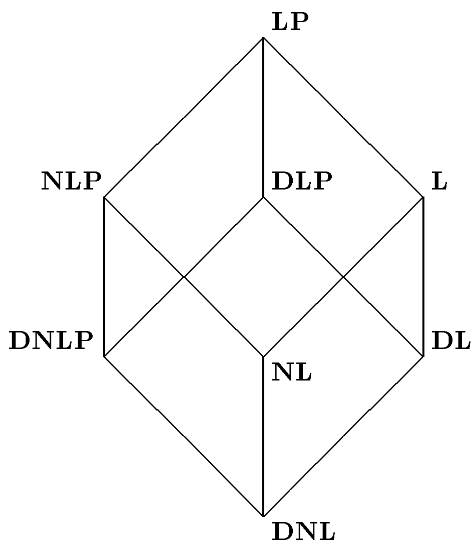


Figure 5: Resource-sensitive logics: precedence, dominance, dependency

In [32] it is shown that the $\Diamond, \Box$ connectives provide a theory of systematic communication between these systems. The communication is two-way: it is shown how one can fully recover the structural discrimination of a weaker logic from within a system with a more liberal resource management regime, and how one can reintroduce the structural flexibility of a stronger logic within a system with a more articulate notion of structure-sensitivity.

Consider a pair of logics $\mathcal{L}_0, \mathcal{L}_1$ where $\mathcal{L}_0$ is a 'southern' neighbour of $\mathcal{L}_1$. Let us write $\mathcal{L}\Diamond$ for the system $\mathcal{L}$ extended with the unary operators $\Diamond, \Box$ with their minimal residuation logic. For the 12 edges of the cube of Fig 5, one can define embedding translations $(\cdot)^\flat : \mathcal{F}(\mathcal{L}_0) \mapsto \mathcal{F}(\mathcal{L}_1\Diamond)$ which impose the structural discrimination of $\mathcal{L}_0$ in $\mathcal{L}_1$ with its more liberal resource management, and $(\cdot)^\sharp : \mathcal{F}(\mathcal{L}_1) \mapsto \mathcal{F}(\mathcal{L}_0\Diamond)$ which license relaxation of structure sensitivity in $\mathcal{L}_0$ in such a way that one fully recovers the flexibility of the the coarser $\mathcal{L}_1$. The embedding translations decorate critical subformulae in the target logic with the operators $\Diamond, \Box$. The translations are defined on the product $\bullet$ of the source logic: their action on the implicational formulas is fully determined by the residuation laws. For the $\cdot^\flat$ type of embedding, the modal decoration has the effect of blocking a structural rule that would be applicable otherwise. For the $\cdot^\flat$ direction, the modal decoration gives access to a controlled version of a structural rule which is unavailable in its 'global' (non-decorated) version.

We illustrate the two-way structural control with the pair **NL** and **L**. Let us subscript the connectives in **NL** with 0 and those of **L** with 1. The **L** system has an associative resource management which is insensitive to constituent bracketing. Extending **L** with the operators $\Diamond, \Box$ we can recover control over associativity by means of the translation $\cdot^\flat : \mathcal{F}(\mathbf{NL}) \mapsto \mathcal{F}(\mathbf{L}\Diamond)$ below.

$$
(\dagger) \qquad \begin{array}{c}
p^\flat = p \\
(A \bullet_0 B)^\flat = \Diamond(A^\flat \bullet_1 B^\flat) \\
(A/_0 B)^\flat = \Box A^\flat /_1 B^\flat \\
(B\backslash_0 A)^\flat = B^\flat \backslash_1 \Box A^\flat
\end{array}
$$

One can then prove the following embedding result.

$$
\mathbf{NL} \vdash A \to B \quad \text{iff} \quad \mathbf{L}\Diamond \vdash A^\flat \to B^\flat
$$

Consider next the other direction of communication: suppose one wants to obtain the structural flexibility of **L** from within the system **NL** with its rigid constituent sensitivity. For the (‡) embedding translation $\cdot^\sharp : \mathcal{F}(\mathbf{L}) \mapsto \mathcal{F}(\mathbf{NL}\Diamond)$ one can use the same decoration schema as for (†). This time, one achieves the desired embedding result by means of a modally controlled version of the structural rule of Associativity, relativized to the critical $\Diamond$ decoration. Compare the global version (A) with its image under $(\cdot)^\sharp$, $(A_\diamond)$.

$$\mathcal{L}_1 : \quad A \bullet_1 (B \bullet_1 C) \longleftrightarrow (A \bullet_1 B) \bullet_1 C \quad (A)$$

$$\mathcal{L}_0 : \quad \Diamond(A \bullet_0 \Diamond(B \bullet_0 C)) \longleftrightarrow \Diamond(\Diamond(A \bullet_0 B) \bullet_0 C) \quad (A_\diamond)$$

$$(\ddagger) \qquad \begin{array}{c} p^\sharp = p \\ (A \bullet_1 B)^\sharp = \Diamond(A^\sharp \bullet_0 B^\sharp) \\ (A/_1B)^\sharp = \Box A^\sharp/_0 B^\sharp \\ (B\backslash_1 A)^\sharp = B^\sharp\backslash_0 \Box A^\sharp \end{array}$$

The embedding theorem then takes the following form.

$$\mathbf{L} \vdash A \rightarrow B \quad \text{iff} \quad \mathbf{NL}\Diamond + A_\diamond \vdash A^\sharp \rightarrow B^\sharp$$

The derivations below illustrate the strategies of imposing structural control or relaxing structure sensitivity with the Geach rule, the characteristic theorem which differentiates **L** from **NL**. On the left, we try to derive the $\cdot^\flat$ translation of the Geach rule in $\mathbf{L}\Diamond$. The resource management regime is associative — still the derivation fails because of the structural $(\cdot)^\diamond$ decoration which makes the C resource inaccessible for the functor $\Box B/_1 C$. On the right one finds a successful derivation of the $\cdot^\sharp$ translation in $\mathbf{NL}\Diamond$. Although the resource management regime in this case does not allow free rebracketing, the $\Diamond$ decoration gives access to the modal version of the structural rule.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{FAIL}}{((((\Box A/_1B, \Box B/_1C)^\bullet)^\diamond, C)^\bullet)^\diamond \Rightarrow A}}{(((\Box A/_1B, \Box B/_1C)^\bullet)^\diamond, C)^\bullet \Rightarrow \Box A}\,\Box R}{((\Box A/_1B, \Box B/_1C)^\bullet)^\diamond \Rightarrow \Box A/_1C}\,/_1R}{(\Box A/_1B, \Box B/_1C)^\bullet \Rightarrow \Box(\Box A/_1C)}\,\Box R}{\Box A/_1B \Rightarrow \Box(\Box A/_1C)/_1(\Box B/_1C)}\,/_1R$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{C \Rightarrow C \quad \dfrac{B \Rightarrow B}{(\Box B)^\diamond \Rightarrow B}\,\Box L}{((\Box B/_0C, C)^\bullet)^\diamond \Rightarrow B}\,/_0L \quad \dfrac{A \Rightarrow A}{(\Box A)^\diamond \Rightarrow A}\,\Box L}{((\Box A/_0B, ((\Box B/_0C, C)^\bullet)^\diamond)^\bullet)^\diamond \Rightarrow A}\,/_0L}{((((\Box A/_0B, \Box B/_0C)^\bullet)^\diamond, C)^\bullet)^\diamond \Rightarrow A}\,A_\diamond}{(((\Box A/_0B, \Box B/_0C)^\bullet)^\diamond, C)^\bullet \Rightarrow \Box A}\,\Box R}{((\Box A/_0B, \Box B/_0C)^\bullet)^\diamond \Rightarrow \Box A/_0C}\,/_0R}{(\Box A/_0B, \Box B/_0C)^\bullet \Rightarrow \Box(\Box A/_0C)}\,\Box R}{\Box A/_0B \Rightarrow \Box(\Box A/_0C)/_0(\Box B/_0C)}\,/_0R$$

$$\mathbf{L}\Diamond \not\vdash (A/_0B)^\flat \Rightarrow ((A/_0C)/_0(B/_0C))^\flat \qquad \mathbf{NL}\Diamond + (A_\diamond) \vdash (A/_1B)^\sharp \Rightarrow ((A/_1C)/_1(B/_1C))^\sharp$$

3.2 Procedural control

The type of resource control discussed in §3.1 concerns the static aspects of well-formed structural configurations of resources such as represented by the term structure of the antecedent of a derivable sequent. In the present section we show how one can dynamically control the use of assumptions in the process of proof search via $\Diamond, \Box$ decoration. Our concrete objective is to logically enforce the König-Hepple-Hendriks goal-directed head-driven search regime for sequent proof search by means of an appropriate $\Diamond, \Box$ decoration.

In the literature on automated deduction, it is well known that cut-free Gentzen proof search is still suboptimal from the efficiency perspective: there may be different (cut-free!) derivations leading to one and the same proof term. Restricting ourselves to the implicational fragment, the spurious non-determinism in the search space has two causes ([48]): (i) permutability of [L] and [R] inferences, and (ii) permutability of [L] inferences among themselves, i.e. non-determinism in the choice of the active formula in the antecedent. A so-called *goal directed* (or: uniform) search regime performs the non-branching [R] inferences before the [L] inferences (re (i)), whereas *head driven* search commits the choice of the antecedent active formula in terms of the goal formula (re (ii)). Such optimized search regimes have been proposed in the context of Linear Logic programming in [25, 2]. In the categorial setting, goal-directed head-driven proof search for product-free **L** was introduced in [27] and worked out in [23] who provided a proof of the safeness (no proof terms are lost) and non-redundancy (each proof term has a unique derivation) of the method. We present the Hepple regime in the format of Hendriks [20] with Curry-Howard semantic term labelling.

$$[\mathrm{Ax}/\star\mathrm{L}]\frac{}{x : p^\star \Rightarrow x : p} \qquad \frac{\Gamma, u : B^\star, \Gamma' \Rightarrow t : p}{\Gamma, u : B, \Gamma' \Rightarrow t : p^\star}[\star\mathrm{R}]$$

$$[/\mathrm{R}]\frac{\Delta, x : B \Rightarrow t : A^\star}{\Delta \Rightarrow \lambda x.t : A/B^\star} \qquad \frac{\Delta \Rightarrow u : B^\star \quad \Gamma, x : A^\star, \Gamma' \Rightarrow t : C}{\Gamma, s : A/B^\star, \Delta, \Gamma' \Rightarrow t[x/su] : C}[/\mathrm{L}]$$

$$[\backslash\mathrm{R}]\frac{x : B, \Delta \Rightarrow t : A^\star}{\Delta \Rightarrow \lambda x.t : B\backslash A^\star} \qquad \frac{\Delta \Rightarrow u : B^\star \quad \Gamma, x : A^\star, \Gamma' \Rightarrow t : C}{\Gamma, \Delta, s : B\backslash A^\star, \Gamma' \Rightarrow t[x/su] : C}[\backslash\mathrm{L}]$$

The **L*** calculus eliminates the spurious non-determinism of the original presentation **L** by annotating sequents with a procedural control operator '*'. Goal sequents $\Gamma \Rightarrow t : A$ in **L** are replaced by **L*** goal sequents $\Gamma \Rightarrow t : A^\star$. With respect to the first cause of spurious ambiguity (permutability of [L] and [R] inferences), the control part of the [R] inferences forces one to remove all connectives from the succedent until one reaches an atomic succedent. At that point, the '*' control is transmitted from succedent to antecedent: the [*R] selects an active antecedent formula the head of which ultimately, by force of the control version of the Axiom sequent [*L], will have to match the (now atomic) goal type. The [L] implication inferences initiate a '*' control derivation on the minor premise, and transmit the '*' active declaration from conclusion to major (right) premise. The effect of the flow of control information is to commit the search to the target type selected in the [*R] step. This removes the second source of spurious ambiguity: permutability of [L] inferences. It can be shown that the **L*** regime eliminates spurious ambiguity. Syntactically, derivability in **L** and **L*** coincide. Semantically, the set of **L*** proof terms forms a subset of the set of **L** terms. But, modulo logical equivalence, no readings are lost moving from **L** to **L***. Moreover, the **L*** system has the desired one-to-one correspondence between readings and proofs.

Uniform proof search: modal control. The control operators $\Diamond, \Box$ make it possible to enforce the König-Hepple-Hendriks uniform head-driven search regime via a modal translation. Our proposal is a variant on the "lock-and-key" method of [35]: we force a particular execution strategy for successful proof search by decorating formulae with the $\Box$ ('lock') and $\Diamond$ ('key') control operators. We use the base residuation logic for $\Diamond, \Box$, plus weak distributivity principles $K1, K2$ for the interaction between the unary and the binary families. For convenience we repeat the Gentzen transformation of the $K1, K2$ structural postulates from our discussion above.

	STRUCTURAL POSTULATE	STRUCTURAL RULE
$K1:$	$\Diamond(A \bullet B) \rightarrow \Diamond A \bullet B$	$\dfrac{\Gamma[((\Delta_1)^\diamond, \Delta_2)] \Rightarrow A}{\Gamma[((\Delta_1, \Delta_2))^\diamond] \Rightarrow A}\ K1$
$K2:$	$\Diamond(A \bullet B) \rightarrow A \bullet \Diamond B$	$\dfrac{\Gamma[(\Delta_1, (\Delta_2)^\diamond)] \Rightarrow A}{\Gamma[((\Delta_1, \Delta_2))^\diamond] \Rightarrow A}\ K2$

To establish the equivalence with **L*** search, we can use the sugared presentation of **L** where Associativity is compiled away so that *binary* punctuation $(\cdot, \cdot)$ can be omitted (but not the unary $(\cdot)^\diamond$!). This gives the following compiled format for $K1, K2$:

$$\frac{\Gamma, (A)^\diamond, \Gamma' \Rightarrow B}{(\Gamma, A, \Gamma')^\diamond \Rightarrow B}\ K'$$

TRANSLATION: FORMULAE, SEQUENTS. We define the translation mapping first on the formula level, and then extend it to the level of **L*** sequents, where we have to distinguish marked and unmarked formulae. On the formula level, define mappings $(\cdot)^1, (\cdot)^0 : \mathcal{F}(/, \backslash) \mapsto \mathcal{F}(/, \backslash, \diamondsuit, \Box)$, for antecedent and succedent formula occurrences respectively.

$$\begin{array}{rl} (p)^1 = p & (p)^0 = \Box p \\ (A/B)^1 = (A)^1/(B)^0 & (A/B)^0 = (A)^0/\Box(B)^1 \\ (B\backslash A)^1 = (B)^0\backslash(A)^1 & (B\backslash A)^0 = \Box(B)^1\backslash(A)^0 \end{array}$$

The formulae of a sequent $\Gamma \Rightarrow A$ in **L*** are partitioned by the '*' annotation in a set of marked formulae — a singleton, since there is only one '*' per sequent — and a set of unmarked formulae. We extend the translation mapping taking this difference into account. The antecedent and succedent translation functions $(\cdot)_1, (\cdot)_0$ below are defined in terms of $(\cdot)^1, (\cdot)^0$, but they act in a different way on marked and on unmarked formulae.

$$(A_1, \ldots, A_n)_1 = \overline{A_1}, \ldots, \overline{A_n} \quad \text{where} \quad \overline{A} = \begin{cases} (A)^1 & \text{if } A \text{ is '*' marked} \\ \Box(A)^1 & \text{otherwise} \end{cases}$$

$$(A)_0 = \begin{cases} (A)^0 & \text{if } A \text{ is '*' marked} \\ A & \text{otherwise} \end{cases}$$

We now have the following proposition.

$$\mathbf{L}^\star \vdash \Gamma \Rightarrow A^\star \qquad \text{iff} \qquad \mathbf{L}\diamondsuit\mathbf{K}' \vdash (\Gamma)_1 \Rightarrow (A)_0$$

EQUIVALENCE OF L^* AND $L\diamondsuit K'$. The $(\Rightarrow)$ direction of the equivalence can be proved by straightforward induction on the length of derivations in **L***. For the more delicate $(\Leftarrow)$ direction, we have to show that $\mathbf{L}\diamondsuit K'$ does not derive *more* than **L***. We give a case analysis of the choice points in the top-down (backward-chaining) unfolding of the search space, and show that $\mathbf{L}\diamondsuit K'$ can make no moves that would lead the search out of the space defined by the translation mapping.

Below we juxtapose the **L*** rules and axiom and their counterpart in $\mathbf{L}\diamondsuit K'$. We treat only one implication. For the $\mathbf{L}\diamondsuit K'$ version, we interleave the proof unfolding with the evaluation of the translation mapping. As an auxiliary notion, we have functions ACTIVE and LOCKED which for a sequent return the set of formulae matching the input condition for a logical rule (ACTIVE), and those which no logical rule is applicable to (LOCKED).

Proof search starts with an **L*** goal sequent $\Gamma \Rightarrow (A)^\star$. The goal type A is either atomic or complex, the antecedent is of length 1 or greater than 1. Consider first the case of a complex goal type and $1 \leq |\Gamma|$. On the left the **L*** $[/R]$ rule, on the right the corresponding derivation in $\mathbf{L}\diamondsuit K'$. Both (†) and (‡) stand in a feeding relation with themselves. For the roots of the derivations, we have ACTIVE(†)$= \{A/B\}$, ACTIVE(‡)$= \{(A/B)^0\}$; for the leaves, ACTIVE(†)$= \{A\}$, ACTIVE(‡)$= \{(A)^0\}$. Note that all antecedent formulae in (‡) have main connective $\Box$ as a result of the translation mapping. The $\Box$ connective acts as a lock: embedded connectives in these formulae will only be accessible after the removal of $\Box$.

$$(\dagger) \quad \frac{\Gamma, B \Rightarrow (A)^\star}{\Gamma \Rightarrow (A/B)^\star}\,/R \qquad \frac{\dfrac{\Box(\Gamma)^1, \Box(B)^1 \Rightarrow (A)^0}{\Box(\Gamma)^1 \Rightarrow (A)^0/\Box(B)^1}\,/R}{\Box(\Gamma)^1 \Rightarrow (A/B)^0}\,(\cdot)^0 \quad (\ddagger)$$

Consider now the case where the recursion on succedent implications bottoms out, i.e. where we reach the '*' marked atomic head of the goal formula. In **L*** the only applicable rule in this situation is $[\star R]$ which transmits the '*' marking from succedent to antecedent. $[\star R]$ is non-deterministic: any antecedent formula B can receive the '*' marking. In $\mathbf{L}\diamondsuit K'$ the active atom is realized as $(p)^0 = \Box p$. The only applicable rule here is $[\Box R]$ which, by residuation, realizes $\Box$ as $(\cdot)^\diamond$ on the antecedent. $[\Box R]$ can only be followed by $[K']$, which non-deterministically pushes $(\cdot)^\diamond$ to an arbitrary antecedent formula $\Box(B)^1$. At that point $[\Box L]$ becomes applicable, which through the elimination of $\Box$ shifts $(B)^1$ from LOCKED to ACTIVE. Again, roots and leaves of the (†) (‡) derivations agree on ACTIVE and LOCKED.

$$(\dagger)\quad \dfrac{\Gamma,(B)^\star,\Gamma' \Rightarrow p}{\Gamma,B,\Gamma' \Rightarrow (p)^\star}\,\star R \qquad \dfrac{\dfrac{\dfrac{\dfrac{\Box(\Gamma)^1,(B)^1,\Box(\Gamma')^1 \Rightarrow p}{\Box(\Gamma)^1,(\Box(B)^1)^\diamond,\Box(\Gamma')^1 \Rightarrow p}\,\Box L}{(\Box(\Gamma)^1,\Box(B)^1,\Box(\Gamma')^1)^\diamond \Rightarrow p}\,K'}{\Box(\Gamma)^1,\Box(B)^1,\Box(\Gamma')^1 \Rightarrow \Box p}\,\Box R}{\Box(\Gamma)^1,\Box(B)^1,\Box(\Gamma')^1 \Rightarrow (p)^0}\,(\cdot)^0 \quad (\ddagger)$$

Next we analyse the possible antecedent configurations for the different choices of active formula. The active formula is either atomic or complex, and the context is either empty or non-empty. Let us consider these in turn, starting with the non-empty context case. If the active formula is atomic, the derivation fails, in $\mathbf{L*}$ and in $\mathbf{L}\Diamond K'$. If the active formula is complex (i.e. of the form B/C or $C\backslash B$), the only applicable rule, in $\mathbf{L*}$ and in $\mathbf{L}\Diamond K'$, is $[/L]$ $([\backslash L])$. The derivation branches, initiating uniform head-driven search for the negative subtype of the goal formula in the left premise, and declaring the positive subtype active in the right premise. Roots and leaves of the derivations in $\mathbf{L*}$ and in $\mathbf{L}\Diamond K'$ agree on the ACTIVE, LOCKED partitioning.

$$\dfrac{\Delta \Rightarrow (B)^\star \quad \Gamma,(A)^\star,\Gamma' \Rightarrow p}{\Gamma,(A/B)^\star,\Delta,\Gamma' \Rightarrow p}\,/L \qquad \dfrac{\dfrac{\Box(\Delta)^1 \Rightarrow (B)^0 \quad \Box(\Gamma)^1,(A)^1,\Box(\Gamma')^1 \Rightarrow p}{\Box(\Gamma)^1,(A)^1/(B)^0,\Box(\Delta)^1,\Box(\Gamma')^1 \Rightarrow p}\,/L}{\Box(\Gamma)^1,(A/B)^1,\Box(\Delta)^1,\Box(\Gamma')^1 \Rightarrow p}\,(\cdot)^1$$

Finally, consider the base cases of the recursion. Below the correspondence when the $\mathbf{L*}$ Axiom sequent, i.e. $[\star L]$ is reached.

$$\dfrac{}{(p)^\star \Rightarrow p}\,\star L \qquad \dfrac{p \Rightarrow p}{(p)^1 \Rightarrow p}\,(\cdot)^1$$

For the sake of completeness, one should add the case of the trivial initial sequent $p \Rightarrow (p)^\star$, though the issue of spurious ambiguity hardly arises here. Below the $\mathbf{L*}$ form and its $\mathbf{L}\Diamond K'$ counterpart.

$$\dfrac{\dfrac{}{(p)^\star \Rightarrow p}\,\star L}{p \Rightarrow (p)^\star}\,\star R \qquad \dfrac{\dfrac{\dfrac{p \Rightarrow p}{(\Box p)^\diamond \Rightarrow p}\,\Box L}{\Box p \Rightarrow \Box p}\,\Box R}{\Box(p)^1 \Rightarrow (p)^0}\,(\cdot)^1,(\cdot)^0$$

ILLUSTRATION: GEACH. Without the constraint on uniform head-driven search, there are two $\mathbf{L}$ sequent derivations for the Geach transition. They produce the same proof term.

$$\dfrac{\dfrac{\dfrac{\dfrac{c \Rightarrow c \quad b \Rightarrow b}{b/c,c \Rightarrow b}\,/L \quad a \Rightarrow a}{a/b,b/c,c \Rightarrow a}\,/L}{a/b,b/c \Rightarrow a/c}\,/R}{a/b \Rightarrow (a/c)/(b/c)}\,/R \qquad \dfrac{\dfrac{\dfrac{c \Rightarrow c \quad \dfrac{b \Rightarrow b \quad a \Rightarrow a}{a/b,b \Rightarrow a}\,/L}{a/b,b/c,c \Rightarrow a}\,/L}{a/b,b/c \Rightarrow a/c}\,/R}{a/b \Rightarrow (a/c)/(b/c)}\,/R$$

GEACH: UNIFORM HEAD-DRIVEN SEARCH. Of these two, only the first survives in the $\mathbf{L*}$ regime.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(c)^\star \Rightarrow c}\,\star L}{c \Rightarrow (c)^\star}\,\star R \quad \dfrac{}{(b)^\star \Rightarrow b}\,\star L}{(b/c)^\star,c \Rightarrow b}\,/L}{b/c,c \Rightarrow (b)^\star}\,\star R \quad \dfrac{}{(a)^\star \Rightarrow a}\,\star L}{(a/b)^\star,b/c,c \Rightarrow a}\,/L}{a/b,b/c,c \Rightarrow (a)^\star}\,\star R}{a/b,b/c \Rightarrow (a/c)^\star}\,/R}{a/b \Rightarrow ((a/c)/(b/c))^\star}\,/R \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(c)^\star \Rightarrow c}\,\star L}{c \Rightarrow (c)^\star}\,\star R \quad \dfrac{\text{FAIL}}{a/b,(b)^\star \Rightarrow a}}{a/b,(b/c)^\star,c \Rightarrow a}\,/L}{a/b,b/c,c \Rightarrow (a)^\star}\,\star R}{a/b,b/c \Rightarrow (a/c)^\star}\,/R}{a/b \Rightarrow ((a/c)/(b/c))^\star}\,/R$$

Uniform head-driven search: modal control. We interleave the proof unfolding and the unpacking of the $(\cdot)^1, (\cdot)^0$ translations.

$$
\begin{array}{cl}
\text{to be cont'd} & \\ \cline{1-1}
(a)^1/(b)^0, \Box(b/c)^1, \Box(c)^1 \Rightarrow a & \\ \cline{1-1}
(a/b)^1, \Box(b/c)^1, \Box(c)^1 \Rightarrow a & (\cdot)^1 \\ \cline{1-1}
(\Box(a/b)^1)^\diamond, \Box(b/c)^1, \Box(c)^1 \Rightarrow a & \Box L \\ \cline{1-1}
(\Box(a/b)^1, \Box(b/c)^1, \Box(c)^1)^\diamond \Rightarrow a & K' \\ \cline{1-1}
\Box(a/b)^1, \Box(b/c)^1, \Box(c)^1 \Rightarrow \Box a & \Box R \\ \cline{1-1}
\Box(a/b)^1, \Box(b/c)^1, \Box(c)^1 \Rightarrow (a)^0 & (\cdot)^0 \\ \cline{1-1}
\Box(a/b)^1, \Box(b/c)^1 \Rightarrow (a)^0/\Box(c)^1 & /R \\ \cline{1-1}
\Box(a/b)^1, \Box(b/c)^1 \Rightarrow (a/c)^0 & (\cdot)^0 \\ \cline{1-1}
\Box(a/b)^1 \Rightarrow (a/c)^0/\Box(b/c)^1 & /R \\ \cline{1-1}
\Box(a/b)^1 \Rightarrow ((a/c)/(b/c))^0 & (\cdot)^0
\end{array}
$$

Consider first the interaction of [/R] rules and selection of the active antecedent type. Antecedent types all have $\Box$ as main connective. The $\Box$ acts as a *lock*: a $\Box A$ formula can only become active when it is *unlocked* by the key $\Diamond$ (or $(\cdot)^\diamond$ in structural terms). The key becomes available only when the head of the goal formula is reached: through residuation, $[\Box R]$ transmits $\Diamond$ to the antecedent, where it selects a formula via $[K']$.

Transmission of the active formula. There is only *one* key $\Diamond$ by residuation on the $\Box$ of the goal formula. As soon as it is used to unlock an antecedent formula, that formula has to remain active and connect to the Axiom sequent.

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\dfrac{\dfrac{\dfrac{c \Rightarrow c}{(c)^1 \Rightarrow c}\,(\cdot)^1}{(\Box(c)^1)^\diamond \Rightarrow c}\,\Box L}{\Box(c)^1 \Rightarrow \Box c}\,\Box R}{\Box(c)^1 \Rightarrow (c)^0}\,(\cdot)^0
\qquad
\dfrac{b \Rightarrow b}{(b)^1 \Rightarrow b}\,(\cdot)^1
}{(b)^1/(c)^0, \Box(c)^1 \Rightarrow b}\,/L
}{(b/c)^1, \Box(c)^1 \Rightarrow b}\,(\cdot)^1
}{(\Box(b/c)^1)^\diamond, \Box(c)^1 \Rightarrow b}\,\Box L
}{(\Box(b/c)^1, \Box(c)^1)^\diamond \Rightarrow b}\,K'
}{\Box(b/c)^1, \Box(c)^1 \Rightarrow \Box b}\,\Box R
}{\Box(b/c)^1, \Box(c)^1 \Rightarrow (b)^0}\,(\cdot)^0
\qquad
\dfrac{a \Rightarrow a}{(a)^1 \Rightarrow a}\,(\cdot)^1
}{(a)^1/(b)^0, \Box(b/c)^1, \Box(c)^1 \Rightarrow a}\,/L
$$

Below, we show how the wrong identification of the antecedent head leads to failure. The key to unlock $\Box(a/b)^1$ has been spent on the wrong formula. As a result, the implication in $(a/b)^1$ cannot become active. Compare with the failure of the corresponding **L*** derivation above.

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{c \Rightarrow c}{(c)^1 \Rightarrow c}\,(\cdot)^1
}{(\Box(c)^1)^{\Diamond} \Rightarrow c}\,\Box L
}{\Box(c)^1 \Rightarrow \Box c}\,\Box R
}{\Box(c)^1 \Rightarrow (c)^0}\,(\cdot)^0
\qquad
\dfrac{\text{FAILS}}{\Box(a/b)^1, (b)^1 \Rightarrow a}\,\dagger
}{\Box(a/b)^1, (b)^1/(c)^0, \Box(c)^1 \Rightarrow a}\,/L
}{\Box(a/b)^1, (b/c)^1, \Box(c)^1 \Rightarrow a}\,(\cdot)^1
}{\Box(a/b)^1, (\Box(b/c)^1)^{\Diamond}, \Box(c)^1 \Rightarrow a}\,\Box L
}{(\Box(a/b)^1, \Box(b/c)^1, \Box(c)^1)^{\Diamond} \Rightarrow a}\,K'
}{\Box(a/b)^1, \Box(b/c)^1, \Box(c)^1 \Rightarrow \Box a}\,\Box R
$$

4 Conclusion

This paper is a technical investigation of the architecture of mixed categorial type logics. The *raison d'être* for such an exercise derives from new applications of these logics to problems of grammatical analysis — without the linguistic motivation many of the logical issues addressed above would simply not arise. The reader who is interested in linguistic applications can turn to [21, 22], where one finds a multimodal analysis of comparatives and ellipsis, [28, 29] for a head-adjunction treatment of clitics in French, or [47] for a logical encoding of linear precedence constraints and word order domains in terms of $\Diamond, \Box$.

Appendix. Cut elimination for $\Diamond, \Box$

In this Appendix, we establish the equivalence of the axiomatic and Gentzen presentations of the logic for $\Diamond, \Box$, and we show that the Gentzen presentation allows Cut Elimination. We start with the pure residuation logic for $\Diamond, \Box$ and then consider the addition of structural postulates/rules $K, T, 4$.

Equivalence of the axiomatic and Gentzen presentations. To compare the presentations of Fig 2 and Fig 4 we extend the formula representation Δ° of a structural configuration Δ to the language $\mathcal{F}(/, \bullet, \backslash, \Diamond, \Box)$ in the obvious way: $(\Delta_1, \Delta_2)^\circ = \Delta_1^\circ \bullet \Delta_2^\circ$, $(\Delta)^\circ = \Diamond\Delta^\circ$, $A^\circ = A$. The sequent presentation for the language $\mathcal{F}(/, \bullet, \backslash, \Diamond, \Box)$ can then be shown to be equivalent to the combinator axiomatisation in the sense that every combinator $f : A \to B$ gives a proof of $A \Rightarrow B$, and every proof of a sequent $\Gamma \Rightarrow B$ gives a combinator $f : \Gamma^\circ \to B$.

From combinators to sequents. To obtain the Gentzen rules [$\Diamond$L], [$\Diamond$R], [$\Box$L], [$\Box$R] from combinator deductions, we use the Isotonicity of $\Diamond, \Box$ (cf. Fig 3) in addition to the residuation inferences μ, μ^{-1}. Given the formula equivalent Γ° for sequent terms Γ, μ gives [$\Box$R]$^\circ$, [$\Diamond$L]$^\circ$ makes premise and conclusion identical, and Isotonicity for $\Diamond$ gives [$\Diamond$R]$^\circ$. The only non-trivial case is [$\Box$L]. Consider first the case where the context Γ is empty. The combinator derivation of [$\Box$L]$^\circ$ is given below.

$$
\dfrac{\dfrac{f : A \to B}{(f)^{\Box} : \Box A \to \Box B}}{\mu((f)^{\Box}) : \Diamond\Box A \to B}
\qquad\qquad
\dfrac{A \Rightarrow B}{(\Box A)^{\Diamond} \Rightarrow B}\,\Box L
$$

Next the case where the context Γ in the [$\Box$L] premise $\Gamma[A] \Rightarrow B$ is non-empty. Let g be $\Gamma[A]^\circ \to B$. Let $\pi(g)$ be a sequence of μ, β, γ residuation inferences isolating A on the left of the arrow. Then we obtain the formula equivalent of the conclusion of [$\Box$L] via the deduction $\pi^{-1}(\mu(\Box(\pi(g))))$.

From sequents to combinators. To obtain the combinators **id**, $f \circ g$ (Transitivity), $\mu(f)$, $\mu^{-1}(g)$ (Residuation) from sequent derivations, we use the Cut rule. Once we have established the equivalence of the combinator and the sequent presentation, we prove Cut Elimination for the latter. [Ax] gives **id**, $f \circ g$ is a special case of Cut. The crucial new cases $\mu(f)$, $\mu^{-1}(g)$ follow.

$$\dfrac{\dfrac{A \Rightarrow A}{(A)^\diamond \Rightarrow \Diamond A}\,\Diamond R \quad f : \Diamond A \Rightarrow B}{\dfrac{(A)^\diamond \Rightarrow B}{\mu(f) : A \Rightarrow \Box B}\,\Box R}\,(cut) \qquad \dfrac{g : A \Rightarrow \Box B \quad \dfrac{B \Rightarrow B}{(\Box B)^\diamond \Rightarrow B}\,\Box L}{\dfrac{(A)^\diamond \Rightarrow B}{\mu^{-1}(g) : \Diamond A \Rightarrow B}\,\Diamond L}\,(cut)$$

Cut elimination: principal cuts. We now extend the Cut Elimination result to the new connectives $\Diamond, \Box$. We proceed by induction on the complexity of the cut formula, and distinguish principal cuts, where the cut formula is active in both cut premises, from permutation conversions, where this is not the case.

Below the new cases of principal cuts, with cut formula $\Diamond A$ and $\Box A$. Replacement of a cut on $\Diamond A$ ($\Box A$) by a cut on A of smaller degree.

$$\dfrac{\dfrac{\Delta \Rightarrow A}{(\Delta)^\diamond \Rightarrow \Diamond A}\,\Diamond R \quad \dfrac{\Gamma[(A)^\diamond] \Rightarrow B}{\Gamma[\Diamond A] \Rightarrow B}\,\Diamond L}{\Gamma[(\Delta)^\diamond] \Rightarrow B}\,(cut) \quad \rightsquigarrow \quad \dfrac{\Delta \Rightarrow A \quad \Gamma[(A)^\diamond] \Rightarrow B}{\Gamma[(\Delta)^\diamond] \Rightarrow B}\,(cut)$$

$$\dfrac{\dfrac{(\Delta)^\diamond \Rightarrow A}{\Delta \Rightarrow \Box A}\,\Box R \quad \dfrac{\Gamma[A] \Rightarrow B}{\Gamma[(\Box A)^\diamond] \Rightarrow B}\,\Box L}{\Gamma[(\Delta)^\diamond] \Rightarrow B}\,(cut) \quad \rightsquigarrow \quad \dfrac{(\Delta)^\diamond \Rightarrow A \quad \Gamma[A] \Rightarrow B}{\Gamma[(\Delta)^\diamond] \Rightarrow B}\,(cut)$$

Cut elimination: permutation conversions. The new cases where the active formula in the left or right premise is different from the Cut formula allow for the usual elimination strategy: permutation of the Cut rule and the logical rule. The Cut is moved upwards , and becomes of lower degree. Below the left premise antecedent cases for $\Diamond A$ and $\Box A$.

$$\dfrac{\dfrac{\Gamma[(A)^\diamond] \Rightarrow B}{\Gamma[\Diamond A] \Rightarrow B}\,\Diamond L \quad \Delta[B] \Rightarrow C}{\Delta[\Gamma[\Diamond A]] \Rightarrow C}\,(cut) \quad \rightsquigarrow \quad \dfrac{\dfrac{\Gamma[(A)^\diamond] \Rightarrow B \quad \Delta[B] \Rightarrow C}{\Delta[\Gamma[(A)^\diamond]] \Rightarrow C}\,(cut)}{\Delta[\Gamma[\Diamond A]] \Rightarrow C}\,\Diamond L$$

$$\dfrac{\dfrac{\Delta[A] \Rightarrow B}{\Delta[(\Box A)^\diamond] \Rightarrow B}\,\Box L \quad \Gamma[B] \Rightarrow C}{\Gamma[\Delta[(\Box A)^\diamond]] \Rightarrow C}\,(cut) \quad \rightsquigarrow \quad \dfrac{\dfrac{\Delta[A] \Rightarrow B \quad \Gamma[B] \Rightarrow C}{\Gamma[\Delta[A]] \Rightarrow C}\,(cut)}{\Gamma[\Delta[(\Box A)^\diamond]] \Rightarrow C}\,\Box L$$

Permutation conversion: right premise antecedent. Active type $\Diamond A$ or $\Box A$ in the antecedent of the right Cut premise. Notation: $\Gamma[\Delta_1, \Delta_2]$ for a structure Γ with substructures Δ_1, Δ_2, not necessarily sisters.

$$\dfrac{\Delta \Rightarrow A \quad \dfrac{\Gamma[A, (B)^\diamond] \Rightarrow C}{\Gamma[A, \Diamond B] \Rightarrow C}\,\Diamond L}{\Gamma[\Delta, \Diamond B] \Rightarrow C}\,(cut) \quad \rightsquigarrow \quad \dfrac{\dfrac{\Delta \Rightarrow A \quad \Gamma[A, (B)^\diamond] \Rightarrow C}{\Gamma[\Delta, (B)^\diamond] \Rightarrow C}\,(cut)}{\Gamma[\Delta, \Diamond B] \Rightarrow C}\,\Diamond L$$

$$\dfrac{\Delta \Rightarrow A \quad \dfrac{\Gamma[A, B] \Rightarrow C}{\Gamma[A, (\Box B)^\diamond] \Rightarrow C}\,\Box L}{\Gamma[\Delta, (\Box B)^\diamond] \Rightarrow C}\,(cut) \quad \rightsquigarrow \quad \dfrac{\dfrac{\Delta \Rightarrow A \quad \Gamma[A, B] \Rightarrow C}{\Gamma[\Delta, B] \Rightarrow C}\,(cut)}{\Gamma[\Delta, (\Box B)^\diamond] \Rightarrow C}\,\Box L$$

Permutation conversion: right premise succedent. Active type $\Diamond A$ or $\Box A$ in the succedent of the right Cut premise.

$$\dfrac{\Gamma \Rightarrow A \quad \dfrac{(\Delta[A])^\diamond \Rightarrow B}{\Delta[A] \Rightarrow \Box B}\,\Box R}{\Delta[\Gamma] \Rightarrow \Box B}\,(cut) \quad \rightsquigarrow \quad \dfrac{\dfrac{\Gamma \Rightarrow A \quad (\Delta[A])^\diamond \Rightarrow B}{(\Delta[\Gamma])^\diamond \Rightarrow B}\,(cut)}{\Delta[\Gamma] \Rightarrow \Box B}\,\Box R$$

$$\dfrac{\Delta \Rightarrow A \quad \dfrac{\Gamma[A] \Rightarrow B}{(\Gamma[A])^\diamond \Rightarrow \Diamond B}\,\Diamond R}{(\Gamma[\Delta])^\diamond \Rightarrow \Diamond B}\,(cut) \quad \rightsquigarrow \quad \dfrac{\dfrac{\Delta \Rightarrow A \quad \Gamma[A] \Rightarrow B}{\Gamma[\Delta] \Rightarrow B}\,(cut)}{(\Gamma[\Delta])^\diamond \Rightarrow \Diamond B}\,\Diamond R$$

Structural rules from structural postulates. The discussion so far concerns the pure residuation logic for $\Diamond, \Box$. Let us now extend the equivalence between axiomatic and Gentzen style presentation to the structural postulates and rules. To obtain the sequent rules $T, 4, K$ from axiomatic derivations, it is enough to consider the case where the context Γ is empty, as we have seen above. The following deductions give the formula equivalent of the structural rules $T, 4, K$. We leave $K1, K2$ to the reader.

$$\dfrac{4 : \Diamond\Diamond\Delta^\circ \to \Diamond\Delta^\circ \quad f : \Diamond\Delta^\circ \to A}{f \circ 4 : \Diamond\Diamond\Delta^\circ \to A} \quad \rightsquigarrow \quad \dfrac{[(\Delta)^\diamond]^\circ \Rightarrow A}{[((\Delta)^\diamond)^\diamond]^\circ \Rightarrow A}\,4$$

$$\dfrac{T : \Delta^\circ \to \Diamond\Delta^\circ \quad f : \Diamond\Delta^\circ \to A}{f \circ T : \Delta^\circ \to A} \quad \rightsquigarrow \quad \dfrac{[(\Delta)^\diamond]^\circ \Rightarrow A}{\Delta^\circ \Rightarrow A}\,T$$

$$\dfrac{K : \Diamond(\Delta_1^\circ \bullet \Delta_2^\circ) \to \Diamond\Delta_1^\circ \bullet \Diamond\Delta_2^\circ \quad f : \Diamond\Delta_1^\circ \bullet \Diamond\Delta_2^\circ \to A}{f \circ K : \Diamond(\Delta_1^\circ \bullet \Delta_2^\circ) \to A} \quad \rightsquigarrow \quad \dfrac{[((\Delta_1)^\diamond, (\Delta_2)^\diamond)]^\circ \Rightarrow A}{[((\Delta_1, \Delta_2))^\diamond]^\circ \Rightarrow A}\,K$$

Structural postulates from structural rules. Derivation of the structural postulates via Gentzen proofs is straightforward.

$$\dfrac{\dfrac{A \Rightarrow A}{(A)^\diamond \Rightarrow \Diamond A}\,\Diamond R}{A \Rightarrow \Diamond A}\,T \qquad \dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{(A)^\diamond \Rightarrow \Diamond A}\,\Diamond R}{((A)^\diamond)^\diamond \Rightarrow \Diamond A}\,4}{(\Diamond A)^\diamond \Rightarrow \Diamond A}\,\Diamond L}{\Diamond\Diamond A \Rightarrow \Diamond A}\,\Diamond L \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{(A)^\diamond \Rightarrow \Diamond A}\,\Diamond R \quad \dfrac{B \Rightarrow B}{(B)^\diamond \Rightarrow \Diamond B}\,\Diamond R}{((A)^\diamond, (B)^\diamond) \Rightarrow \Diamond A \bullet \Diamond B}\,\bullet R}{((A, B))^\diamond \Rightarrow \Diamond A \bullet \Diamond B}\,K}{((A \bullet B))^\diamond \Rightarrow \Diamond A \bullet \Diamond B}\,\bullet L}{\Diamond(A \bullet B) \Rightarrow \Diamond A \bullet \Diamond B}\,\Diamond L$$

Cut elimination: structural rules. We extend the cut elimination algorithm to logics with a structural rule package from $\mathcal{P}(\{T, 4, K\})$. Recall that in the case of connectives the proof of the Cut Elimination theorem is by induction on the complexity of Cut inferences, measured in terms of the number of connectives in the cut formula. The structural rules do not involve decomposition of formulae, so we need an additional complexity measure here.

Following [12, 5], let the *trace* of a cut formula A be the sum of the lengths of the paths in the derivations of the cut premises connecting the two occurrences of A with the point of their first introduction in the proof. The cut elimination steps involving structural rules now assimilate to the permutation cases: if a structural rule feeds the cut inference, we can interchange the order of application of the cut and the structural rule, leading to a situation with decreased *trace*, as the inductive hypothesis requires. Two examples are given below.

$$\dfrac{\dfrac{\Delta[(\Delta')^\diamond] \Rightarrow A}{\Delta[\Delta'] \Rightarrow A}\,T \qquad \Gamma[A] \Rightarrow B}{\Gamma[\Delta[\Delta']] \Rightarrow B}\,(cut) \quad\rightsquigarrow\quad \dfrac{\dfrac{\Delta[(\Delta')^\diamond] \Rightarrow A \quad \Gamma[A] \Rightarrow B}{\Gamma[\Delta[(\Delta')^\diamond]] \Rightarrow B}\,(cut)}{\Gamma[\Delta[\Delta']] \Rightarrow B}\,T$$

$$\dfrac{\Delta \Rightarrow A \qquad \dfrac{\Gamma[(\Gamma'[A])^\diamond] \Rightarrow B}{\Gamma[((\Gamma'[A])^\diamond)^\diamond] \Rightarrow B}\,4}{\Gamma[((\Gamma'[\Delta])^\diamond)^\diamond] \Rightarrow B}\,(cut) \quad\rightsquigarrow\quad \dfrac{\dfrac{\Delta \Rightarrow A \quad \Gamma[(\Gamma'[A])^\diamond] \Rightarrow B}{\Gamma[(\Gamma'[\Delta])^\diamond] \Rightarrow B}\,(cut)}{\Gamma[((\Gamma'[\Delta])^\diamond)^\diamond] \Rightarrow B}\,4$$

References

[1] Abrusci, M., C. Casadio and M. Moortgat (eds) (1994) *Linear Logic and Lambek Calculus.* Proceedings 1st Rome Workshop, June 1993. OTS/DYANA, Utrecht, Amsterdam.

[2] Andreoli, J.-M. (1992) 'Logic programming with focusing proofs in linear logic'. *Journal of Logic and Computation*, 2(3).

[3] Barry, G. and G. Morrill (eds) (1990) *Studies in Categorial Grammar.* CCS, Edinburgh.

[4] Benthem, J. van (1983) 'The semantics of variety in categorial grammar'. Report 83–29, Math Dept. Simon Fraser University, Burnaby, Canada.

[5] Benthem, J. van (1991) *Language in Action. Categories, Lambdas, and Dynamic Logic.* Studies in Logic, North-Holland, Amsterdam.

[6] Benthem, J. van (1984) 'Correspondence theory'. In Gabbay and Guenthner (eds) *Handbook of Philosophical Logic. Vol II.* Dordrecht, pp 167–247.

[7] Belnap, N.D. (1982) 'Display Logic'. *Journal of Philosophical Logic*, **11**, 375–417.

[8] Blyth, T.S. and M.F. Janowitz (1972) *Residuation Theory.* New York.

[9] Bucalo, A. (1994) 'Modalities in Linear Logic weaker than the exponential "of course": algebraic and relational semantics'. JoLLI **3**, 3, 211–232.

[10] Buszkowski, W. (1988) 'Generative power of categorial grammars'. In Oehrle, Bach and Wheeler (eds) *Categorial Grammars and Natural Language Structures.* Dordrecht.

[11] Došen, K. (1992) 'A brief survey of frames for the Lambek calculus'. Zeitschr. f. math. Logik und Grundlagen d. Mathematik **38**, 179–187.

[12] Došen, K. (1988,1989) 'Sequent systems and groupoid models'. *Studia Logica* **47**, 353–385, **48**, 41–65.

[13] Dunn, J.M. (1991) 'Gaggle theory: an abstraction of Galois connections and residuation, with applications to negation, implication, and various logical operators'. In Van Eijck (ed.) *Logics in AI.* JELIA Proceedings. Springer, Berlin.

[14] Dunn, M. (1993) 'Partial Gaggles Applied to Logics With Restricted Structural Rules'. In Došen and Schröder-Heister (eds) *Substructural Logics.* Oxford.

[15] Fuchs, L. (1963) *Partially-ordered Algebraic Systems.* New York.

[16] Gabbay, D. (1991) *Labelled Deductive Systems.* Draft. Oxford University Press (to appear).

[17] Gabbay, D. (1993) 'A general theory of structured consequence relations'. In Došen and Schröder-Heister (eds) *Substructural Logics.* Oxford.

[18] Girard, J.-Y., (1993) 'On the unity of logic'. Annals of Pure and Applied Logic, **59**, 201–217.

[19] Girard, J.-Y., (1994) 'Light Linear Logic: extended abstract'. Ms LMD, Marseille.

[20] Hendriks, H. (1993) *Studied Flexibility*. PhD Dissertation, ILLC, Amsterdam.

[21] Hendriks, P. (1995a) *Comparatives in Categorial Grammar*. Ph.D. Thesis, Groningen.

[22] Hendriks, P. (1995b) 'Ellipsis and multimodal categorial type logic'. In Morrill (ed) *Proceedings Formal Grammar Conference*. Barcelona.

[23] Hepple, M. (1990) *The Grammar and Processing of Order and Dependency: A Categorial Approach*. Ph.D. Dissertation, Edinburgh.

[24] Hepple, M. (1994) 'Labelled deduction and discontinuous constituency'. In [1].

[25] Hodas, J.S. and D. Miller (1994) 'Logic programming in a fragment of intuitionistic linear logic'. *Information and Computation*, **110**(2):327–365.

[26] Kandulski, W. (1988) 'The non-associative Lambek calculus'. In Buszkowski, Marciszewski and van Benthem (eds) *Categorial Grammar*. Amsterdam.

[27] König, E. (1991) 'Parsing as natural deduction'. *Proceedings of the 27th Annual Meeting of the ACL*. Vancouver.

[28] Kraak, E. (1995a) *French Clitics. A Categorial Perspective*. MA Thesis, Utrecht.

[29] Kraak, E. (1995b) 'Controlling resource management: French clitics'. In Leiss (ed) Esprit BRA Dyana-2 Deliverable R1.3.C.

[30] Kracht, M. (1993) 'Power and weakness of modal display calculus'. Ms Mathematisches Institut, Freie Universität Berlin.

[31] Kurtonina, N. (1995) *Frames and Labels. A Modal Analysis of Categorial Inference*. PhD Dissertation, OTS Utrecht, ILLC Amsterdam.

[32] Kurtonina, N. and M. Moortgat (1995) 'Structural control'. In Blackburn and de Rijke (eds) *Logic, Structures and Syntax*. Reidel, Dordrecht (to appear).

[33] Lambek, J. (1958) 'The Mathematics of Sentence Structure', *American Mathematical Monthly* **65**, 154–170.

[34] Lambek, J. (1988) 'Categorial and categorical grammar'. In Oehrle, Bach and Wheeler (eds) *Categorial Grammars and Natural Language Structures*. Dordrecht.

[35] Lincoln, P., J. Mitchell, P. Scedrov and N. Shankar (1992) 'Decision problems for propositional linear logic'. *Annals of pure and applied logic*, **56**:239–311.

[36] Moortgat, M. and G. Morrill (1991) 'Heads and phrases. Type calculus for dependency and constituent structure'. Ms OTS Utrecht.

[37] Moortgat, M. and R. Oehrle (1993) *Logical parameters and linguistic variation. Lecture notes on categorial grammar*. 5th European Summer School in Logic, Language and Information. Lisbon.

[38] Moortgat, M. and R.T. Oehrle (1994) 'Adjacency, dependency and order'. Proceedings 9th Amsterdam Colloquium, pp 447–466.

[39] Morrill, G. (1990) 'Intensionality and boundedness'. *L& P* **13**, pp 699–726.

[40] Morrill, G. (1992) 'Categorial formalisation of relativisation: pied-piping, islands and extraction sites'. Report LSI–92–23–R, Universitat Politècnica de Catalunya.

[41] Morrill, G. (1994) 'Structural facilitation and structural inhibition'. In [1].

[42] Morrill, G. (1994) *Type Logical Grammar*. Kluwer, Dordrecht.

[43] Routley, R. and R.K. Meyer (1972–73) 'The semantics of entailment I'. In H.Leblanc (ed) *Truth, Syntax, Modality*, North-Holland, Amsterdam, pp 199–243.

[44] Steedman, M. (1993) 'Categorial grammar. Tutorial overview'. *Lingua* **90**:221–258.

[45] Szabolcsi, A. (1987) 'Bound variables in syntax: are there anyΓ In Bartsch, van Benthem and van Emde Boas (eds) *Semantics and Contextual Expressions*, 294–318, Foris, Dordrecht.

[46] Venema, Y. (1993) 'Meeting strength in substructural logics'. UU Logic Preprint. To appear in *Studia Logica*.

[47] Versmissen, K. (1995) 'Word order domains in categorial grammar'. In Morrill (ed) *Proceedings Formal Grammar Conference*. Barcelona.

[48] Wallen, L.A. (1990) *Automated Deduction in Nonclassical Logics*. MIT Press, Cambridge, London.

[49] Wansing, H. (1992) 'Sequent calculi for normal modal propositional logics'. ILLC Report LP–92–12, Amsterdam.