Chapter 1

Polarity Items in Resource Logics. A comparison.

Raffaella Bernardi[1]

Abstract.

Polarity Items (PIs) are linguistic expressions known for being polarity sensitive. In the paper we show how resource logics can account for these phenomena. By doing so we investigate the relation between unary operators of Multimodal Lambek calculus and the binary ones of Multiplicative Linear Logic. We show that working with unary operators provides precisely the expressiveness required to deal with PIs.

1 Resource Logics

Resource logic is a family of logics which differ from each other in the structural rules they are characterized by. The name indicates that they are sensitive to the number or/and structure of the resources they work with. Quite obviously, this is a desirable property when reasoning with linguistic signs. As is well known, the non associative Lambek calculus (NL), firstly presented in [Lambek,1961] is the basic fragment of resource logics relevant for linguistic application [Abrusci and Casadio,1994]. It consists of logic rules for the binary operators, product and implications, where the latter are the residuals of the former. In other words, the implications are connected to the product operator by means of a basic algebraic relation and their logic rules are simply obtained from this relation. However, NL is not powerful enough when reasoning with natural language. For example, it cannot account for structural variations from language to language, or for the grammaticality of complex sentences where phenomena like discontinuous dependencies or coordination are involved. For these reasons a more powerful system is required. We will look at two possible ways of achieving this goal, one which increases the expressiveness of the logical side of the system using unary operators besides the binary ones; and one which extends

[1] I am grateful to Richard Oehrle for his helpful suggestions on this work.

the power of the structural component allowing to permute and change the order of the composed structures. Following the first solution we obtain Multimodal (non associative) Lambek Calculus (MMNL) [Moortgat,1997], whereas adding permutation and associativity we reach Multiplicative Linear Logic (MLL) [Danos and Regnier,1989].

The aim of this paper is to show the advantages of having unary operators at our disposal when reasoning with linguistic structures. In particular, we will look at expressions, known as polarity items, which are sensitive to the monotonicity properties of the context they occur in [Ladusaw,1979] and which require some structural reasoning when part of complex sentences, for example in relative clauses. There are several reasons for choosing them as a case-study. First of all Polarity Items (PIs) are a pervasive phenomenon of language, with extensive lexical variation and with similarity with other context-sensitive expressions, viz. anaphors. Therefore, they can be used when investigating how languages differ from each other with respect to structural properties. Moreover, their distribution is grammatically governed while being semantically and pragmatically motivated [Krifka,1995]. Hence, assuming a proof-theoretical approach on the issue combines the two levels, giving a syntactic account which incorporates the semantic and pragmatic properties involved.

In [Fry,1999] MLL is used as a meaning constructor within a formal grammar, viz. Lexical Functional Grammar (LFG), to account for PIs distribution. In [Bernardi,1999] the same phenomenon is presented using MMNL. By comparing the two systems we will answer the question why unary operators are needed when applying resource logics to natural languages.

The paper is structured as follows. We first briefly present NL, then we show the result of extending the system with the unary operators and with structural rules. Finally, we discuss how PIs can be accounted for in such systems.

2 Logical Inferences

The basic idea behind the application of resource logics to natural language is to assign formulas of a logical language to linguistic structures in such a way that reasoning about the former corresponds to reasoning about the latter. When assuming this perspective two important points have to be kept in mind: (i) linguistic items can require logically related types, in other words expressions can be of the ‘same types’ while differing on some specific properties; and (ii) composition of linguistic structures is order sensitive. The ingredients needed to meet these requirements are (a) a logical language able to express both compositions and sub-categorization of types; (b) a close connection between the linguistic structures and logical types; and (c) a way to locally control structural composition. We introduce these requirements step by step starting from the logical language containing both unary and binary operators, showing how the logical types and the linguistic structures can be connected, and finally discussing how the order composition can be controlled.

The logical language is recursively built from the unary operators, $\Diamond$ and $\Box^{\downarrow}$, and the binary ones, $\backslash$, $\bullet$ and $/$, as follows.

Given $\mathsf{ATOM} = \{np, n, s\}$, the well formed formulas are,

$$\begin{aligned}\mathsf{FORM} \quad := \quad & \mathsf{ATOM} \mid \Diamond\mathsf{FORM} \mid \Box^{\downarrow}\mathsf{FORM} \mid \mathsf{FORM} \bullet \mathsf{FORM} \mid \\ & \mathsf{FORM}/\mathsf{FORM} \mid \mathsf{FORM}\backslash\mathsf{FORM}\end{aligned}$$

where the $\bullet$ is the product operator, the $\backslash$ and $/$ are the implication ones; whereas the $\Diamond$ and $\Box^{\downarrow}$ are the residual unary operators.

As we said, the logical behavior of these operators is given by the residuation law,

$$(1) \quad \Diamond A \Rightarrow B \quad \text{iff} \quad (2) \quad A \Rightarrow \Box^{\downarrow} B$$

$$(3) \quad A \Rightarrow C/B \quad \text{iff} \quad (4) \quad A \bullet B \Rightarrow C \quad \text{iff} \quad (5) \quad B \Rightarrow A\backslash C$$

where, for example, the implication from (1) to (2) can be read as: if assuming that an expression has type $\Diamond A$ we can prove that it also has type B, then from the assumption that an expression has type A we can infer that it has type $\Box^{\downarrow} B$.

Since resource logics mainly differ from each other in the structural level, we will need to focus the attention on this part when comparing MMNL and MLL. For this reason in our presentation we will use a proof format in which both the logical types and the structural part are displayed. As mentioned in (b) the two sides have to be closely connected. Hence, the fact that in MMNL the logic types are built on both binary and unary operators will be reflected on the structural level, as well. Structures are built as follows.

Given the set of formulas FORM, the allowed structures are,

$$\mathsf{STRUCT} := \mathsf{FORM} \mid \mathsf{STRUCT} \circ \mathsf{STRUCT} \mid \langle \mathsf{STRUCT} \rangle$$

where $\circ$ and $\langle\cdot\rangle$ are the structural connectives corresponding to $\bullet$ and $\Diamond$, respectively.

Combing these three elements (the logical and structural languages and the residuation law) with regards to the binary operators, NL can be presented in natural deduction sequent format as follows.

Logical Rules

$$\frac{\Gamma \vdash A/B \quad \Delta \vdash B}{(\Gamma \circ \Delta) \vdash A}\,[/\mathrm{E}] \qquad \frac{(\Gamma \circ B) \vdash A}{\Gamma \vdash A/B}\,[/\mathrm{I}]$$

$$\frac{\Gamma \vdash B \quad \Delta \vdash B\backslash A}{(\Gamma \circ \Delta) \vdash A}\,[\backslash\mathrm{E}] \qquad \frac{(B \circ \Gamma) \vdash A}{\Gamma \vdash B\backslash A}\,[\backslash\mathrm{I}]$$

$$\frac{\Delta \vdash A \bullet B \quad \Gamma[(A \circ B)] \vdash C}{\Gamma[\Delta] \vdash C}\,[\bullet\mathrm{E}] \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma \circ \Delta \vdash A \bullet B}\,[\bullet\mathrm{I}]$$

where $\Gamma \vdash A$ means that the structure Γ is proved to be of type A, and $\Gamma[A]$ denotes that Γ contains a specific formula A. To facilitate the reading in the examples structural formulas will be replaced by the linguistic items they represent, e.g. instead of $(np \circ np\backslash s) \vdash s$ we will write (mary $\circ$ works) $\vdash s$, since 'mary' has type np and 'works' has type $np\backslash s$.

In a logical approach to natural language an important goal is to assign types which can be logically related to each other. In particular, when reasoning with PIs phenomena, the following logical implications, discussed in [Moortgat,1997], will be used:

Theorems

1a. Application:	$A \bullet (A\backslash B) \Rightarrow B$
	$(A/B) \bullet B \Rightarrow A$
2a. Co-application:	$A \Rightarrow B\backslash(B \bullet A)$
	$A \Rightarrow (A \bullet B)/B$
3a. Monotonicity $\backslash, /$:	*if* $A \Rightarrow B$ *and* $C \Rightarrow D$, *then* $(D\backslash A) \Rightarrow (C\backslash B)$
	if $A \Rightarrow B$ *and* $C \Rightarrow D$, *then* $(A/D) \Rightarrow (B/C)$

where, for example, the first theorem means that if a linguistic structure has type $A \bullet (A\backslash B)$, then it has type B as well.

If besides the logical rules of the binary operators we consider the unaries', we have MMNL. The logic rules for these operators obtained by the residuation law above are,

Logical Rules

$$\frac{\Delta \vdash \Diamond A \quad \Gamma[\langle A \rangle] \vdash B}{\Gamma[\Delta] \vdash B}\,[\Diamond\mathrm{E}] \qquad \frac{\Gamma \vdash A}{\langle \Gamma \rangle \vdash \Diamond A}\,[\Diamond\mathrm{I}]$$

$$\frac{\Gamma \vdash \Box^{\downarrow} A}{\langle \Gamma \rangle \vdash A}\,[\Box^{\downarrow}\mathrm{E}] \qquad \frac{\langle \Gamma \rangle \vdash A}{\Gamma \vdash \Box^{\downarrow} A}\,[\Box^{\downarrow}\mathrm{I}]$$

When speaking of PIs we are going to use the following logical relations among types headed by unary operators,

Theorems

$$\begin{array}{lll} \text{1b. Application:} & \Diamond\Box^{\downarrow}A \Rightarrow A \\ \text{2b. Co-application:} & A \Rightarrow \Box^{\downarrow}\Diamond A \end{array}$$

The other logic system we are interested in, MLL, is obtained from NL allowing the structural rules of associativity and permutation at a global level,

$$\frac{\Gamma[\Delta_1 \circ (\Delta_2 \circ \Delta_3)] \vdash C}{\Gamma[(\Delta_1 \circ \Delta_2) \circ \Delta_3] \vdash C}\,[\text{Ass}] \quad \frac{\Gamma[(\Delta_2 \circ \Delta_1)] \vdash C}{\Gamma[(\Delta_1 \circ \Delta_2)] \vdash C}\,[\text{Per}]$$

Having these structural properties, the equivalences below among logical types hold.

Theorems

$$A/B \Leftrightarrow B\backslash A \qquad\qquad A \bullet B \Leftrightarrow B \bullet A$$

Due to these theorems, in a commutative and associative system the two implications collapse into the linear one, $\multimap$, and the product is replaced by the tensor, $\otimes$. However, in order to facilitate the comparison of the systems, in the rest of the paper we will keep on using NL (and MMNL) notation when working with MLL.

Within MMNL there is a more restrictive way of introducing structural reasoning. As explained in [Moortgat,1999] and [Moortgat,1995] the residual unary operators, can be used as control devices when dealing, for example, with discontinuous or long distance phenomena. By means of them structural properties, in fact, can be allowed at a local level being lexically anchored. For a formal analysis of the general theory of resource control in term of unary operators see [Kurtonina and Moortgat,1996]. For example a restrictive version of the permutation rule is,

$$\frac{\Gamma[(\langle\Delta_2\rangle \circ \Delta_1)] \vdash C}{\Gamma[(\Delta_1 \circ \langle\Delta_2\rangle)] \vdash C}\,[\text{Per}]$$

We will come back to this point in the last part of this paper.

3 Polarity Items

Polarity Items are linguistic expressions which depend on the monotonicity properties of their context for grammaticality. In this paper we will focus on Negative Polarity Items (NPIs), i.e. items which can only be felicitously used in a context with a 'negative' flavor, see [Giannikidou,1997] for a detailed discussion. This phenomenon has been studied from both syntactic and semantical perspectives. The first line started with Klima [Klima,1964] who claimed that NPIs have to be "in construction with" a trigger, where the triggers are either overt negation or affective element, e.g. *surprise*. The semantic approach traces back to Fauconnier [Fauconnier,1978] and Ladusaw [Ladusaw,1979] who characterized the trigger as downward-entailing operators, i.e. functions which reverse the order among their arguments. In our approach this semantic property of the trigger will be stored in the lexicon; consequentially when the NPIs are "constructed with" their trigger, the composition is proved to be grammatically correct, otherwise the proof fails. In English NPIs are expressions such as *yet*, *at all*, *anything*, and their triggers can be items such as *nobody*, *rarely*, as shown below.

Linguistic data

(ia.)	Somebody left.	(iiia.)	*Kim says *anything at all.*
(ib.)	**Nobody** left.	(iiib.)	Kim **rarely** says *anything at all.*
(iia.)	*Somebody left *yet.*	(iva.)	**Nobody** says *anything.*
(iib.)	**Nobody** left *yet.*	(ivb.)	**Nobody rarely** says *anything.*

As is clear from these examples, in a type logical approach to natural language a trigger as *nobody* has to be of a similar type as a non-licensing general quantifier, e.g. *somebody*, so that sentences in (i) are proved to be grammatically correct in a similar way. However, the former differs from the latter for being a downward-entailing operator, and this difference motivates the contrast between (iia) and (iib). Taking into account this fact in the lexicon means that the type assigned to *somebody* has to be a 'sub-type' of the one assigned to *nobody.* The sentences in (iii) and (iv) show that the same trigger can license more than one NPI, and that more triggers can occur in a context where only one NPI is present. These facts are predicted when assuming a proof-theoretical approach as it is explained in the rest of this paper.

3.1 Logic Types for NPI

In this section we compare the results obtained when using MLL and MMNL to account for NPIs. The two proposals have been presented in [Fry,1999] and [Bernardi,1999][2], respectively. The two formalizations follow the general idea of deriving the grammaticality of a sequence of words by means of logical rules starting from axioms stored in a lexicon, i.e. from the assignments of

[2]In [Bernardi,1999] MMNL is employed both for checking the features agreement among NPIs and their licensers and marking the linguistic parsed structures so to derive inferences from them. In order to satisfy this double task the information, here stored in the logic types as $\Box^{\downarrow}$, is passed to the structural side of the sequents as $\langle\cdot\rangle$.

types to linguistic items. As we have shown in the previous section, the two logic systems used differ in their structural properties. Moreover, the logical constants of MLL are only the binary ones. Therefore, when comparing the linguistic applications of these logical systems we have to look at the logic types stored in the lexicon, and at the way structures can be composed. These will be our leading criteria in our comparison.

In [Fry,1999] MLL is used as a meaning constructor for Lexical Functional Grammar (LFG). We focus the attention on this part skipping the others aspects of LFG (c-structure, f-structure, etc.). For reason of space we present the system only briefly, giving the main idea behind the formalization.

Polarity is considered as a resource represented by a proposition, ℓ, which is required by the NPI and produced by the licenser. Let B/A and $B\backslash A$ be the standard logic types of the licenser and the NPI, respectively, the resource is added to them in the following way:

Logic types:

Licenser: $B/((A \bullet \ell)/\ell)$ NPI: $(B \bullet \ell)\backslash(A \bullet \ell)$

From these types the system can correctly parse the linguistic data given above. Strings with more than one licenser (ivb.) or no NPI consuming the produced resource (ib.), are correctly parsed thanks to the fact that the licenser's type introduces the resource hypothetically and "cleans it up within itself" if it is not consumed. The NPI's type, on the other hand, has the resource in both its antecedent and its succedent, therefore it simply "borrows" it, without really consuming the resource. This correctly predicts that multiple NPIs can be licensed by the same trigger (iiib.).

As intuitively explained above, the logic type for the licenser has to account for at least all cases derived by the non-licensing version, i.e. they have to be ordered by a logical derivability relation. More precisely, the type enriched with the polarity resources has to derive the one without them:

$$B/((A \bullet \ell)/\ell) \Rightarrow B/A \tag{1.1}$$

The reader will better understand this resource exchange by means of the example below.

A classical downward-entailing operator is the general quantifier *nobody*. General quantifiers have type: $s/(np\backslash s)$. Following the schema we obtain, `nobody`: $s/((iv \bullet \ell)/\ell)$, where $iv = (np\backslash s)$. As for the NPI, let's consider the adverb *yet*, of type $iv\backslash iv$. Applying the schema we have: `yet`: $(\ell \bullet iv)\backslash(\ell \bullet iv)$.

Example Nobody left yet

$$
\dfrac{\text{nobody} \vdash s/((iv \bullet \ell)/\ell) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{left} \vdash iv \quad [x \vdash \ell]^1}{\text{left} \circ x \vdash iv \bullet \ell}[\bullet\text{I}] \quad \text{yet} \vdash (iv \bullet \ell)\backslash(iv \bullet \ell)}{(\text{left} \circ x) \circ \text{yet} \vdash iv \bullet \ell}[\backslash E]}{\text{left} \circ (x \circ \text{yet}) \vdash iv \bullet \ell}[\text{Ass}]}{\text{left} \circ (\text{yet} \circ x) \vdash iv \bullet \ell}[\text{Per}]}{(\text{left} \circ \text{yet}) \circ x \vdash iv \bullet \ell}[\text{Ass}]}{\text{left} \circ \text{yet} \vdash (iv \bullet \ell)/\ell}[/\text{I}]^1}{}}{\text{nobody} \circ (\text{left} \circ \text{yet}) \vdash s}[/\ \text{E}]
$$

If we look carefully at the logic type assigned to the licenser using equation 1.1, we realize that it is simply a consequence of the theorems 2a-3a discussed above when presenting NL.

$$
\dfrac{B \Rightarrow B \quad A \Rightarrow (A \bullet \ell)/\ell}{B/(A \bullet \ell)/\ell \Rightarrow B/A}[\text{Mon}]
$$

where $A \Rightarrow (A \bullet \ell)/\ell$ is the Co-application theorem.

A natural question which arises is whether we really need the structural rules of associativity and permutation used in the proof to account for NPIs. In what follows we investigate what happens if we replace the theorem 2a of the binary operators with the corresponding theorem of the unary ones, 2b.

$$
\dfrac{B \Rightarrow B \quad A \Rightarrow \Box^{\downarrow}\Diamond A}{B/\Box^{\downarrow}\Diamond A \Rightarrow B/A}[\text{Mon}]
$$

The theorem gives a new derived type for NPI's licensers in which polarity resource is represented as $\Box^{\downarrow}\Diamond$, e.g. `nobody`: $s/\Box^{\downarrow}\Diamond iv$, and the one for NPIs, is changed consequentially, e.g. `yet`: $\Box^{\downarrow}\Diamond iv\backslash\Box^{\downarrow}\Diamond iv$.

Example Nobody left yet.

$$
\dfrac{\text{nobody} \vdash s/\Box^{\downarrow}\Diamond iv \qquad \dfrac{\dfrac{\dfrac{\text{left} \vdash iv}{\langle\text{left}\rangle \vdash \Diamond iv}[\Diamond\text{I}]}{\text{left} \vdash \Box^{\downarrow}\Diamond iv}[\Box^{\downarrow}\text{I}] \quad \text{yet} \vdash \Box^{\downarrow}\Diamond iv\backslash\Box^{\downarrow}\Diamond iv}{\text{left} \circ \text{yet} \vdash \Box^{\downarrow}\Diamond iv}[\backslash\text{E}]}{\text{nobody} \circ (\text{left} \circ \text{yet}) \vdash s}[/\text{E}]
$$

Comparing the two derivations we see a first desirable consequence brought by the switch from the binary to the unary operators: no structural rules are needed to prove the grammaticality of simple example as the one above. This fact is important for two different reasons. First of all as anticipated in the introduction, natural languages differ from each other with respect to the structural properties they might require. Following this insight, a cross-linguistic studies on NPIs using MMNL might bring evidence to the well known claim of the categorial type logical tradition, namely that the pure calculus of residuation is the logic underlying the universally valid linguistic properties, whereas the structural rules account for variants. Moreover, the fact that structures cannot be globally modified once composed, makes the use of a formal grammar as LFG superfluous. Both the semantic and syntactic composition are handled within the resource logic, avoiding the double formalism.

A second advantage of using unary operators in the type assignment for the licenser and the NPIs, is that no "phantom resources", as ℓ, are used. The property of the licenser of being a downward-entailing operator is not represented using a propositional formula as if it is comparable with the linguistic categories. Having the unary operators, makes it possible to treat some items as having a specific property not shared with other items of the same linguistic family.

So far we have been describing what we gain from a 'theoretical' perspective when using unary operators, looking at simple examples. In the next section we show the advantages of working with such a lexically anchored system considering more complex cases. Our claim is that starting from MMNL we can extend its expressiveness step by step defining the precise fragment of resource logic family required. In other words, we will attempt a first answer to the question: Which are the structural properties needed to parse sentences in which the resource exchange between the licenser and the NPI happens across sentences?

3.2 Structural Reasoning

As anticipated in Section 2, within MMNL long distance phenomena (e.g. wh-movement, relative clauses) are accounted for by means of controlled structural properties. Therefore, when parsing linguistic strings containing NPIs in embedded sentences, some structural reasoning is required. The example below shows a NPI in a subordinate clause licensed by the subject of the main sentence:

Example: (v) **Nobody** thinks *anyone* left.

Due to its structural insensitivity MLL can prove the grammaticality of (v), whereas the pure calculus of residuation for unary and binary operators cannot. However, in the previous section we have shown that to work with NPIs the full power of MLL is not needed. By means of the residual unary operators of MMNL we have to "calibrate" our logic. As far as we can see, the needed property to parse sentence (v) might be the distribution of $\langle\cdot\rangle$:

$$\frac{\Gamma[\langle \Delta_1 \circ \Delta_2 \rangle] \vdash C}{\Gamma[(\Delta_1 \circ \langle \Delta_2 \rangle)] \vdash C}\ [\text{Dis}]$$

In the derivation we will distinguish bottom-up and top-down rules as Dis_1, Dis_2, respectively. Using the type-schema obtained in the previous section for MMNL we can assign the type $\Box^{\downarrow}\Diamond s/\Box^{\downarrow}\Diamond iv$ to ‘anyone’, deriving:

$$
\dfrac{\text{nobody} \vdash s/\Box^{\downarrow}\Diamond(np\backslash s) \qquad \dfrac{\dfrac{\dfrac{\begin{matrix}\vdots \\ \langle \text{anyone} \circ \text{left}\rangle \vdash \Diamond s\end{matrix} \qquad \dfrac{\dfrac{\dfrac{\text{thinks} \vdash (np\backslash s)/s \quad \mathrm{r}_1 \vdash s}{\text{thinks} \circ \mathrm{r}_1 \vdash np\backslash s}[/\mathrm{E}]}{\langle \text{thinks} \circ \mathrm{r}_1\rangle \vdash \Diamond(np\backslash s)}[\Diamond \mathrm{I}]}{\text{thinks} \circ \langle \mathrm{r}_1\rangle \vdash \Diamond(np\backslash s)}[\mathrm{Dis}_2]}{\text{thinks} \circ \langle \text{anyone} \circ \text{left}\rangle \vdash \Diamond(np\backslash s)}[\Diamond \mathrm{E}]}{\langle \text{thinks} \circ (\text{anyone} \circ \text{left})\rangle \vdash \Diamond(np\backslash s)}[\mathrm{Dis}_1]}{\text{thinks} \circ (\text{anyone} \circ \text{left}) \vdash \Box^{\downarrow}\Diamond(np\backslash s)}[\Box^{\downarrow}\mathrm{I}]}{\text{nobody} \circ (\text{thinks} \circ (\text{anyone} \circ \text{left})) \vdash s}[/\mathrm{E}]
$$

The application of the structural rules is controlled by the lexicon type assignments: they can be applied since the structures are marked by the $\langle\cdot\rangle$, which corresponds to the unary operator stored in the type assigned to ‘anyone’. The marked structure – $\langle$think $\circ$ (anyone $\circ$ left)$\rangle$ – carries the information that a NPI is occurring in it and requires for a proper licenser. Finally, this requirement is met by ‘nobody’, and the string is proved to be of type s. If we replace ‘nobody’ with a non-licensing general quantifier the proof would fail as expected, For example, given ‘somebody’, of type $s/(np\backslash s)$, as subject of the main sentence, the proof fails in applying the last rule [/E], the general quantifier does not have the property required by the NPI.

4 Further Questions

From the comparison we have drawn, unary and binary operators of MMNL and MLL, respectively, are closely linked. However, from a linguistic perspective the unary operators show desirable properties which the binary ones do not have: they account for the data in a more ‘minimal’ way. Moreover, the analysis described suggests that using the unary operators we can define the precise fragment of resource logic family. Finally, they allow to give a cross-linguistic perspective on the subject investigating how languages differ from each other with respect of structural properties when looking at PIs phenomena.

Bibliography

[Abrusci and Casadio,1994] M. Abrusci, C. Casadio (eds.). *Linear Logic and Lambek Calculus*, Proceedings 1993 Roma Workshop. ILLC, Amsterdam, 1994.

[Bernardi,1999] R. Bernardi. Monotonic Reasoning from a Proof-Theoretic Perspective. In G. Kruijff and R. Oehrle, editors, *Proceedings of Formal Grammar 1999*, pages 13–24, Utrecht, 1999.

[Danos and Regnier,1989] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.

[Fauconnier,1978] G. Fauconnier. Implication reversal in a natural language. In F. Guenthner and S. Schimidt, editors, *Formal semantics and pragmatics for natural languages.*, pages 289-301, Dordrecht:Reidel, 1978.

[Fry,1999] J. Fry. Proof nets and negative polarity licensing. In M. Dalrymple, editor, *Semantics and Syntax in Lexical Functional Grammar. The Resource Logic approach.*, chapter 3, pages 91–116. MIT Press, 1999.

[Giannikidou,1997] A. Giannikidou. *The landscape of polarity items.* PhD thesis, Groningen, 1997.

[Klima,1964] E. Klima. Negation in English. In J. Fodor and J. Katz, editors, *The structure of language.* Englewood Cliffs:Prentice-Hall, 1964.

[Krifka,1995] M. Krifka. The Semantics and Pragmatics of Polarity Items. In *Linguistic Analysis*, 25:209–257, 1995.

[Kurtonina and Moortgat,1996] N. Kurtonina and M. Moortgat. Structural Control In P. Blackburn and M. de Rijke, editors, *Logic, Structures and Syntax* . Dordrecht: Kluwer, 75-113, 1996.

[Ladusaw,1979] W. Ladusaw. *Polarity sensitivity as inherent scope relations.* PhD thesis, Texas, 1979.

[Lambek,1961] J. Lambek. On the calculus of syntactic types. In R. Jakobson, editor, *Structure of Languages and Its Mathematical Aspects.* 1961.

[Moortgat,1997] M. Moortgat. Categorial Type Logics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 93–178. The MIT Press, Cambridge, Massachusetts, Cambridge, 1997.

[Moortgat,1995] M. Moortgat. In Situ Binding: A Modal Analysis. In P. Dekker and M. Stokhof, editors, *Proceedings of the Tenth Amsterdam Colloqium*, ILLC, pages 93–178, 1995

[Moortgat,1999] M. Moortgat. Meaningful patterns. In J. Gerbrandy, M. Marx, M. de Rijke and Y. Venema, editors, *JFAK. Essays Dedicated to Johan van Benthem on the Occasion of his 50th Birthday.* , Amsterdam University Press, 1999.