Reasoning with Polarity in Categorial Type Logic

Raffaella Bernardi

Contents

1. The Problem

- In formal linguistic literature, one finds examples of theories based on **classifications** of items which belong to the same semantic type but which differ in their syntactic distribution. For example,
  - **generalized quantifiers** have been classified considering their different ways of scope taking [Beghelli and Stowell'97];
  - **wh-phrases** can be divided considering their sensitivity to different weak-islands strength [Szablosci and Zwarts'97];
  - **adverbs** differ in their order relations [Ernst'01];
  - **polarity items** have been distinguished by the sort of licensors they require for grammaticality [Wouden'94,Giannakidou'97].
- In all these cases, the described classifications are based on semantically motivated **subset relations** holding within the domain of interpretation of the involved items.

2. Examples

a. Three referees read an abstract. [Three > An, An > Three].
b. Three referees read few abstracts. [Three > Few, *Few > Three].

a. Who didn't Fido see?
b. *How didn't Fido behave?

a. Usually I speak with anybody I meet;
b. *Yesterday I spoke with anybody I met;
c. It is not the case that I speak with anybody I meet.

'Yesterday' is a veridical function;
'Usually' is nonveridical;
'Anybody' requires a nonveridical function;
$\mathsf{AV} \subseteq \mathsf{NV}$;
e.g. 'It is not the case' is antiveridical.

3. Our Proposal

- **Proposal**: To exploit the logical tools of the Categorial Type Logic (CTL) framework to encode the subset relations as derivability relations among the types of the involved items.
- **Aim**: to show how CTL can contribute to the study of linguistic typologies.
- **Framework**: CTL belongs to the same Formal Grammar family of Classical Categorial Grammar (CG), Combinatory Categorial Grammar (CCG) and Lambek Calculus (NL). Schematically, they differ as following.

CG & CCG	NL	CTL
Categories	Formulas	ditto
Type forming operators $(\backslash, /)$	Logical operators $(\backslash, \bullet, /)$	$(\backslash_i, \bullet_i, /_i, \Diamond_i, \Box_i^{\downarrow}, \ldots)$
Rule schemata	Inference Rules	ditto
Parsing	Deduction	ditto

4. Categorial Type Logic

- CTL provides a modular architecture to study **constants** and **variation** of grammatical composition:
  - **base logic** grammatical invariants, universals of form/meaning assembly;
  - **structural module** non-logical axioms (postulates), lexically anchored options for structural reasoning.
- Up till now, research on the constants of the base logic has focussed on binary operators. E.g.
  - Lifting theorem: $A \vdash (B/A)\backslash B$;

  While unary operators have been used to account for structural variants.
- We will show how **unary operators** can be used
  - to account for linguistic classification encoding the **subset relations** among items of the same semantic type, and
  - to account for **cross-linguistic** differences.

5. Residuation

In [Areces, Bernardi and Moortgat 01] the base logic ($\mathsf{NL}(\Diamond, \cdot^{\mathbf{0}})$) consisting of residuated and Galois connected operators has been studied.

Let $\mathcal{A}_i = (A_i, \sqsubseteq_{A_i})$ be partially ordered sets. A pair of functions (f, g) such that $f : A_1 \to A_2$ and $g : A_2 \to A_1$ forms a **residuated pair** if

$$[RES_1] \quad \forall x \in A_1, y \in A_2 \begin{pmatrix} f(x) \sqsubseteq_{A_2} y \\ \text{iff} \\ x \sqsubseteq_{A_1} g(y) \end{pmatrix}$$

Similarly, a triple of functions (f, g, h) such that $f : A_1 \times A_2 \to A_3$, $g : A_1 \times A_3 \to A_2$, $h : A_3 \times A_2 \to A_1$ forms a **residuated triple** if

$$[RES_2] \quad \forall x \in A_1, y \in A_2, z \in A_3 \begin{pmatrix} x \sqsubseteq_{A_1} h(z, y) \\ \text{iff} \\ f(x, y) \sqsubseteq_{A_3} z \\ \text{iff} \\ y \sqsubseteq_{A_2} g(x, z) \end{pmatrix}$$

6. Monotonicity of Residuated Operators

Saying that (f, g) is a residuated pair is equivalent to the conditions $i)$ and $ii)$,

i. f and g are $[\uparrow]$-functions.

$$\text{ii.}\ \ \forall y \in A_2, x \in A_1 \left(\begin{array}{c} f(g(y)) \sqsubseteq_{A_2} y \\ \text{and} \\ x \sqsubseteq_{A_1} g(f(x)) \end{array} \right)$$

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

i. f is a $[\uparrow, \uparrow]$-function, g is an $[\downarrow, \uparrow]$-function and h is an $[\uparrow, \downarrow]$-function.

$$\text{ii.}\ \ \forall x \in A_1, y \in A_2, z \in A_3 \left(\begin{array}{c} f(x, g(x, z)) \sqsubseteq_{A_3} z \\ \text{and} \\ y \sqsubseteq_{A_2} g(x, f(x, y)) \\ \text{and} \\ f(h(z, y), y) \sqsubseteq_{A_3} z \\ \text{and} \\ x \sqsubseteq_{A_1} h(f(x, y), y) \end{array} \right)$$

7. Residuated Operators in CTL

The connectives $(\backslash, \bullet, /)$ of NL in [Lambek 58, 61] form a **residuated triple** of operators, i.e.

$$[RES_2] \quad \forall A, B, C \in \mathsf{TYPE} \begin{pmatrix} A \vdash C/B \\ \text{iff} \\ A \bullet B \vdash C \\ \text{iff} \\ B \vdash A\backslash C \end{pmatrix}$$

Similarly, the $\Diamond, \Box^{\downarrow}$ connectives introduced in [Moortgat & Kurtonina 95] form a residuated pair,

$$[RES_1] \quad \forall A, B \in \mathsf{TYPE} \begin{pmatrix} \Diamond A \vdash B \\ \text{iff} \\ A \vdash \Box^{\downarrow} B \end{pmatrix}$$

8. Galois Connections

Let $\mathcal{A}_i = (A_i, \sqsubseteq_{A_i})$ be a partially ordered set. Consider a pair of functions $f : A_1 \to A_2$ and $g : A_2 \to A_1$. The pair (f, g) is called a Galois connection if

$$[\text{GC}] \qquad \forall x \in A_1, y \in A_2 \left(\begin{array}{c} y \sqsubseteq_{A_2} f(x) \\ \text{iff} \\ x \sqsubseteq_{A_1} g(y) \end{array} \right)$$

The equivalent formulation of this property in terms of the monotonicity behavior and a composition rule is given by

i. f and g are $[\downarrow]$-functions;

$$\text{ii.} \qquad \forall x \left(\begin{array}{c} x \sqsubseteq f(g(x)) \\ \text{and} \\ x \sqsubseteq g(f(x)) \end{array} \right)$$

Remark 1 Galois connected operators have been also studied in the context of Linear Logic where they are intended to exhibit negation-like behavior. This means that the Galois properties have to be mixed with extra features guaranteeing, for example, a double negation law $f(g(A)) = A = g(f(A))$.

9. Galois Connections in CTL

$\mathsf{NL}(\diamond, \cdot^{\mathbf{0}})$ is obtained by the base logic of residuation $\mathsf{NL}(\diamond)$ by adding the unary Galois connected operators:

$$[\mathrm{GC}] \quad \forall A, B \in \mathsf{TYPE} \begin{pmatrix} A \vdash {}^{\mathbf{0}}B \\ \text{iff} \\ B \vdash A^{\mathbf{0}} \end{pmatrix}$$

Remark 3 $\mathsf{NL}(\diamond)$ is the pure calculus of residuation. But, note that $\backslash$ and $/$ form a **Galois** connection when their positive argument is fixed (i.e. $\backslash B, B/$).

$$[\mathrm{GC}] \quad \forall A, B \in \mathsf{TYPE} \begin{pmatrix} A \vdash B/(A\backslash B) \\ \text{and} \\ A \vdash (B/A)\backslash B \end{pmatrix}$$

This fact is derivation is known as **lifting theorem** and has found interesting linguistic applications.

Remark 4 Similarly, one could consider Dual Galois as well (Goré 98).

10. Models

$\mathsf{NL}(\Diamond, \cdot^{\mathbf{0}})$ has been interpreted using Kripke Models.

Frames $F = \langle W, R_0^2, R_\Diamond^2, R_\bullet^3 \rangle$

W: 'signs', resources, expressions

$R_\bullet^3$: 'Merge', grammatical composition

$R_\Diamond^2$: 'feature checking', structural control

R_0^2: 'feature checking', structural control

Models $\mathcal{M} = \langle F, V \rangle$

Valuation $V : \mathsf{TYPE} \mapsto \mathcal{P}(W)$: types as sets of expressions

11. The Base Logic $\mathbf{NL}(\Diamond, \cdot^0)$

Transitivity/Reflexivity of the derivability relation, plus

$$
\begin{array}{llcl}
(\textsc{res-l}) & A \bullet B \vdash C & \text{iff} & A \vdash C/B \\
(\textsc{res-r}) & A \bullet B \vdash C & \text{iff} & B \vdash A\backslash C \\[1em]
(\textsc{res-1}) & \Diamond A \vdash B & \text{iff} & A \vdash \Box^{\downarrow} B \\[1em]
(\textsc{gal}) & A \vdash {}^{\mathbf{0}}B & \text{iff} & B \vdash A^{\mathbf{0}}
\end{array}
$$

Soundness/Completeness

$$A \vdash B \;\; \text{is provable} \quad \text{iff} \quad \forall F, V, \;\; V(A) \subseteq V(B)$$

See [Areces, Bernardi & Moortgat 2001], also for Gentzen presentation, cut elimination and decidability.

12. Some Useful Derived Properties

(Iso/Anti)tonicity $A \vdash B$ implies

$$\begin{array}{ccc}
\Diamond A \vdash \Diamond B & \text{and} & \Box^{\downarrow} A \vdash \Box^{\downarrow} B \\
{}^{\mathbf{0}}B \vdash {}^{\mathbf{0}}A & \text{and} & B^{\mathbf{0}} \vdash A^{\mathbf{0}} \\
A/C \vdash B/C & \text{and} & C\backslash A \vdash C\backslash B \\
C/B \vdash C/A & \text{and} & B\backslash C \vdash A\backslash C \\
A \bullet C \vdash B \bullet C & \text{and} & C \bullet A \vdash C \bullet B
\end{array}$$

Compositions

$$\begin{array}{cc}
\Diamond\Box^{\downarrow} A \vdash A & A \vdash \Box^{\downarrow}\Diamond A \\
A \vdash {}^{\mathbf{0}}(A^{\mathbf{0}}) & A \vdash ({}^{\mathbf{0}}A)^{\mathbf{0}} \\
(A/B) \bullet B \vdash A & A \vdash (A \bullet B)/B \\
B \bullet (B\backslash A) \vdash A & A \vdash B\backslash(B \bullet A)
\end{array}$$

$\Box$

13. Linguistic Applications

When looking at linguistic applications $\mathsf{NL}(\Diamond, \cdot^{\mathbf{0}})$ offers:

- new derivability relations;
- downward entailment relations.

We will show how

- the new patterns can be used to model the licensing relation accounting for polarity item distribution.

14. Polarity Items Typology

- A **classification** of Polarity Items (PIs) has been described in [Zwarts 1995, Giannakidou 1997] where PIs are considered sensitive to (non-)veridicality.
- In other words, polarity items (syntactic) **distribution** depends on some semantic features, viz. (non-)veridicality, of their licensors.
- Though (non-)veridicality is an **invariant** among natural language expressions, PIs show **different** behavior cross-linguistically. E.g.
  - "Possibly" differs from its Greek counterpart: though they have the same meaning, the Greek version licenses PIs, whereas the English one does not.
- PIs are an interesting phenomena from a **cross-linguistic** perspective: languages differ in the distributional properties of PIs, rather than in their structural occurrence.

15. Non-veridical Contexts

Definition [(Non-)veridical functions] Let f be a boolean function with a boolean argument, a definition of (non-)veridical functions can be given starting from the following basic case: $f \in (t \to t)$

- f is said to be **veridical** iff $[\![f(x)]\!] = 1$ entails $[\![x]\!] = 1$ (e.g. 'yesterday');
- f is said to be **non-veridical** iff $[\![f(x)]\!] = 1$ does not entail $[\![x]\!] = 1$ (e.g. 'usually');
- f is said to be **anti-veridical** iff $[\![f(x)]\!] = 1$ entails $[\![x]\!] = 0$ (e.g. 'It is not the case').

AV functions form a proper subset of the NV ones, $\mathsf{AV} \subset \mathsf{NV}$

□

16. Polarity Items Classification

Based on these distinctions of (non-)veridical contexts, PIs have been classified as follow:

- **positive polarity items** (PPIs) can occur in veridical contexts (V) ('some N');
- **affective polarity items** (APIs) cannot occur in V, i.e. they must occur in non-veridical contexts (NV), (e.g. 'any N');
- **negative polarity items** (NPIs) cannot occur in NV, i.e. they must occur in anti-veridical contexts (AV) (e.g. 'say a word').

Schematicaly, this means that

$$\begin{array}{ll} \mathrm{AV} \circ \Delta\lceil \mathrm{NPI} \rceil & {}^*\mathrm{NV} \circ \Delta\lceil \mathrm{NPI} \rceil, \\ \mathrm{AV} \circ \Delta\lceil \mathrm{API} \rceil & \mathrm{NV} \circ \Delta\lceil \mathrm{API} \rceil, \\ {}^*\mathrm{V} \circ \Delta\lceil \mathrm{NPI} \rceil & {}^*\mathrm{V} \circ \Delta\lceil \mathrm{API} \rceil. \end{array}$$

where $\circ$ is the composition operator, $\Delta\lceil \mathrm{X} \rceil$ means that X is in the structure Δ and has wide scope in it, and * marks ungrammatical composition.

17. A Concrete Example

'Yesterday', 'usually' and 'it is not the case' are all denoted in the domain $D_t^{D_t}$, hence their (syntactic) category is s/s. However,

1. (a) *Yesterday I spoke with anybody I met. $\quad$ $*\mathrm{V} \circ \Delta\lceil \mathrm{API} \rceil$

   (b) *Yesterday I said a word. $\quad$ $*\mathrm{V} \circ \Delta\lceil \mathrm{NPI} \rceil$

2. (a) **Usually** I speak with anybody I meet. $\quad$ $\mathrm{NV} \circ \Delta\lceil \mathrm{API} \rceil$

   (b) *Usually I say a word. $\quad$ $*\mathrm{NV} \circ \Delta\lceil \mathrm{NPI} \rceil$

The type of a structure is determined by the element having wide scope, viz. in $\Delta\lceil X \rceil$ it is determined by X.

18. Syntactic Types

In order to make fine-grained distinctions in the lexical assignments, we can use unary operators.

We need to express that **it is not the case** is in AV, **usually** is in NV and **yesterday** is in V, where AV $\subset$ NV and V$\cap$NV $= \{\}$. Hence, the types of functions in AV derive the types of functions in NV, whereas this does not hold for the functions in V.

Lexicon

It is not...	$\in s/({}^{\mathbf{0}}s)^{\mathbf{0}}$	(AV)
Usually	$\in s/({}^{\mathbf{0}}(\Diamond\Box^{\downarrow}s))^{\mathbf{0}}$	(NV)
Yesterday	$\in s/\Box^{\downarrow}\Diamond s$	(V)

$$AV : s/({}^{\mathbf{0}}s)^{\mathbf{0}} \rightarrow NV : s/({}^{\mathbf{0}}(\Diamond\Box^{\downarrow}s))^{\mathbf{0}} \quad \not\leftrightarrow V : s/\Box^{\downarrow}\Diamond s$$

19. Types for PIs and their Licensors

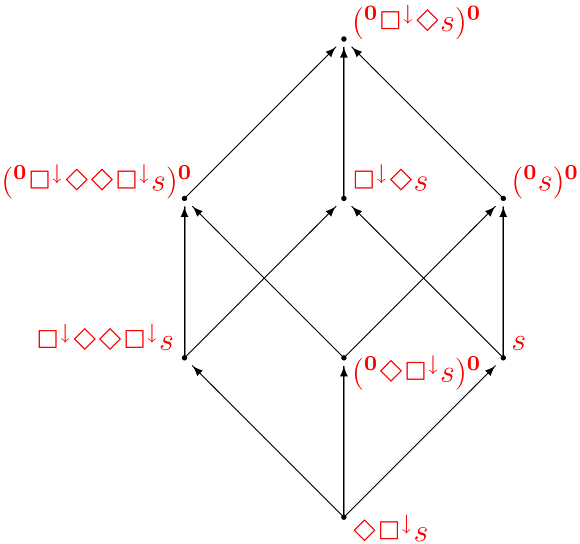
Schematically, the needed types are: $api : (^{\mathbf{0}}(\Diamond\Box^{\downarrow}s))^{\mathbf{0}}$, $npi : (^{\mathbf{0}}s)^{\mathbf{0}}$, $ppi : \Box^{\downarrow}\Diamond s$

$$\begin{array}{ccc} \mathrm{AV} \in A/npi & \mathrm{NV} \in A/api, & \mathrm{V} \in A/ppi \\ api \rightarrow npi & npi \not\rightarrow ppi & api \not\rightarrow ppi. \end{array}$$

$$\dfrac{\mathrm{AV} \vdash A/npi \quad \begin{array}{c} \Delta\lceil\mathrm{API}\rceil \vdash api \\ \vdots \\ \Delta\lceil\mathrm{API}\rceil \vdash npi \end{array}}{\mathrm{AV} \circ \Delta\lceil\mathrm{API}\rceil \vdash A} \qquad \dfrac{\mathrm{NV} \vdash A/api \quad \dfrac{\Delta\lceil\mathrm{NPI}\rceil \vdash npi}{\Delta\lceil\mathrm{NPI}\rceil \vdash api}\,*}{{}^{*}\mathrm{NV} \circ \Delta\lceil\mathrm{NPI}\rceil \vdash A}$$

$\Box$

20. Options for Cross-Linguistic Variation



21. Greek (I)

NPI: **ipe leksi**, API: **kanenan**, FCI: **opudhipote**

1.	**Dhen** idha kanenan. (tr. I didn't see anybody)	Neg > API
2.	**Dhen** ipe leksi oli mera (tr. He didn't say a word all day)	Neg > NPI
3.	*Dhen idha opjondhipote (tr. I didn't see anybody)	*Neg > FCI
4.	Opjosdhipote fititis **bori** na lisi afto to provlima. (tr. Any student can solve this problem.)	Modal > FCI
5.	**An** dhis tin Elena [puthena/opudhipote], . . . (tr. If you see Elena anywhere, . . .)	Cond > API/FCI
6.	**An** pis leksi tha se skotoso. (tr. If you say a word, I will kill you)	Cond > NPI

22. Greek (II)

The data presented above can be summarized as follows:

$\mathsf{AV} \subseteq \mathsf{NV}$, $\mathsf{ONV} \subseteq \mathsf{NV}$, and $\mathsf{AV} \cap \mathsf{ONV} = \{\}$.

ONV: e.g. modal verbs, habituals, generics, imperatives, intensional verbs, future particle.

Greek	FCI	API	NPI	PPI
Veridical	*	*	*	Yes
Negation	*	Yes	Yes	*
Modal verb	Yes	Yes	*	Yes
Conditional	Yes	Yes	Yes	Yes

23. Italian (I)

NPI: **nessuno**, API: **mai**, FCI: **chiunque**

1.	**Non** gioco mai (tr. I don't play ever)	Neg > API
2.	**Non** ho visto nessuno (tr. I haven't seen anybody)	Neg > NPI
3.	***Non** ho visto chiunque (tr. I haven't seen anybody)	*Neg > FCI
4.	Chiunque **puó** risolvere questo problema (tr. Anybody can solve this problem)	Modal > FCI
5.	*Puoi giocare mai (tr. You can play ever)	*Modal > API
6.	*Puoi prendere in prestito nessun libro (tr. You can borrow any book)	*Modal > NPI
7.	**Se** verrai mai a trovarmi, . . . (tr. If you ever come to visit me, . . .)	Cond > API

24. Italian (II)

The data presented above can be summarized as follows:

Italian	FCI	API	NPI	PPI
Veridical	*	*	*	Yes
Negation	*	Yes	Yes	*
Modal verb	Yes	*	*	Yes
Conditional	*	Yes	*	Yes

In a similar way, one can account for the behavior of Dutch negative and positive polarity items. In [van Wouden] it is shown that in Dutch polarity items are sensitive to downward monotonicity, and

- a NPI licensed by the property of a function in DM will be grammatical also when composed with any functions belonging to a stronger set.
- if a PPI is 'allergic' to one specific property shared by the functions of a certain set, it will be ungrammatical when composed with them, but compatible with any other function in a weaker set which does not have this property.

25. Summing Up

- **Semantic** differences among items of the same class are responsible for different **syntactic** behaviors;
- In $\mathsf{NL}(\Diamond, \cdot^{\mathbf{0}})$ these differences can be encoded in the **lexicon** by means of unary operators;
- The derivability relations governing unary operators and the tonicity properties of $\backslash, /$ give precise **instructions** to encode the semantic subset relations involved;
- Starting from the lexicon, the **logical rules** prove the correct distribution of the different items;
- Cross-linguistic differences are accounted for by building different lexicon, facilitating comparisons among languages.

26. What Have We Gained?

Assuming a categorial logic perspective on linguistic typologies help

- gain a deeper understanding of the typological classifications proposed in the literature of formal linguistics;
- carry out cross-linguistic comparisons;
- clarify the consequences predicted by the typologies opening the way to further investigations, and
- discover new dependencies between linguistic phenomena.

27. Questions

- **Observation** The syntatic behavior of some linguistic phenomena is influenced by semantic properties, which cannot be accounted for simply by means of functional applications. Unary operators seem to provide the right expressivity, distinguishing functions denoted within the same domain.
- **Fact** Lambek calculus is in a Curry-Howard correspondence with (a fragment of) typed lambda calculus. The latter is based purely on functional application and the language can represent either atomic or functional expressions.
- **Question** Should the syntactic types classification have any effect on the semantic representation, and if so which are the proper interpretations of the used unary operators?