From Logic to Natural Language via Residuation

Raffaella Bernardi
KRDB, Free University of Bolzano
co-work with Rajeev Goré, Natasha Kurotnina and Michael Moortgat

Contents

1. Logic & Language

Aim to find the universal **core** of all natural languages and their variations

How Using logic to:

- formally define **grammaticality** of sentences and understand how syntactic structures are built
- formally define the **meaning** of sentences and understand how semantic structures are built
- model **syntax-semantic** interface

1.1. Natural Language: syntax

- **Syntax**: "setting out things together", in our case things are words. The main question addressed here is "*How do words compose together to form a grammatical sentence (s) (or fragments of it)?*"
- **Categories**: words are said to belong to *classes*/categories. The main categories are nouns (*n*), verbs (*v*), adjectives (*adj*), determiners (*det*) and adverbs (*adv*).
- **Constituents**: Groups of categories may form a single *unit or phrase* called constituents. The main phrases are noun phrases (*np*), verb phrases (*vp*), prepositional phrases (*pp*). Noun phrases for instance are: "she"; "Michael"; "Rajeev Goré"; "the house"; "a young two-year child".

  Structure: $[[\text{Michael}]_{np}\ [[\text{bought}]_{v}\ [[\text{the}]_{det}\ [\text{house}]_{n}]_{np}]_{vp}]_{s}$
- **Dependency**: Categories are interdependent, for example

Ryanair services $[\text{Pescara}]_{np}$	Ryanair flies $[\text{to Pescara}]_{pp}$
*Ryanair services $[\text{to Pescara}]_{pp}$	*Ryanair flies $[\text{Pescara}]_{np}$

  the verbs services and flies determine which category can/must be juxtaposed. If their constraints are not satisfied the structure is ungrammatical.

1.2. Natural language: semantics

The meaning of sentences is its truth value.

Model Given the domain (of entities) $\{a, b, c, d\}$, and the interpretation below

$$
\begin{array}{lcll}
[\![\texttt{man}]\!] & = & \{a, b, c\}; & \\
[\![\texttt{dog}]\!] & = & \{d\}; & \\
[\![\texttt{fat}]\!] & = & \{a, b, c, d\}; & \\
[\![\texttt{run}]\!] & = & \{a, b\}; & iv \\
[\![\texttt{knows}]\!] & = & \{\langle c, b\rangle, \langle b, c\rangle, \langle a, b\rangle, \langle b, a\rangle\}; & tv \\
[\![\texttt{every man}]\!] & = & \{X | [\![\texttt{man}]\!] \subseteq [\![X]\!]\} & \\
 & = & \{\{a, b, c\}, \{a, b, c, d\}\}. &
\end{array}
$$

The meaning representation for a sentence can be built from the meaning representations of its parts and is based on its syntactic structure.

1.3. Natural language: syntax-semantics

Local Scope: A single linguistic sentence can legitimately have different meaning representations assigned to it. For instance,

- "I saw the man with the telescope" (two syntactic structures!)

  a. John [saw [a man [with the telescope]$_{pp}$]$_{np}$]$_{vp}$ $\exists x.\texttt{Man}(x) \wedge \texttt{Saw}(j,x) \wedge \texttt{Has}(x,t)$

  b. John [[saw [a man]$_{np}$]$_{vp}$ [with the telescope]$_{pp}$]$_{vp}$ $\exists x.\texttt{Man}(x) \wedge \texttt{Saw}(j,x) \wedge \texttt{Has}(j,t)$

- Mary showed each boy an apple.

  a. Then she mixed the apples up and had each boy guess which was his.

  b. The apple was a MacIntosh.

  The sentence has two possible meaning representations:

  a. $\forall y(\texttt{Boy}(y) \rightarrow \exists x(\texttt{Apple}(x) \wedge \texttt{Show}(m,y,x)))$

  b. $\exists x(\texttt{Apple}(x) \wedge \forall y((\texttt{Boy}(y) \rightarrow \texttt{Show}(m,y,x))))$

  but only one syntactic structure: [Mary [[showed [each boy]] [an apple]]] (non-local scope)

1.4. Long distance dependencies

Interdependent constituents need not be juxtaposed, but may form long-distance dependencies, manifested by gaps

- What cities does Ryanair service [. . .]?

The constituent what cities depends on the verb service, but is at the front of the sentence rather than at the object position.

Such distance can be large,

- Which flight do you want me to book [. . .]?
- Which flight do you want me to have the travel agent book [. . .]?

Both non local scope construal and long distance dependencies are challenging phenomena for formal analysis of natural language.

1.5. Formal Grammar

A grammar is a formal device to recognize a language. This task is achieved via

- **Categorization**: a lexicon assigning words to categories. (re-writing rules from non-terminal to terminals)
- **Composition**: rules specifying ways of categorizing phrases. (re-writing rules from non-terminal to non-terminals)

Expressions that cannot be recognized by the grammar are **ungrammatical**.

Example Given the start symbol S, the terminal symbols a, b, and the rules below:

Rules

Rule 1	$S \rightarrow A\ B$	Rule 2	$S \rightarrow A\ S\ B$
Rule 3	$A \rightarrow a$	Rule 4	$B \rightarrow b$

the above grammar recognizes the string $aabb$. It can also be used to obtain its structure/parse tree

1.6. CFG for Natural Language

```
Categorization            Composition
    NP --> john           S --> NP VP
    IV --> walks          VP --> IV
    TV --> knows          VP --> TV NP
    DTV --> gives         VP --> DTV NP NP
    Adj --> poor          N --> Adj N
```

1.7. Logical Grammar

We want to find the Logic that properly models natural language syntax-semantics interface.

- We consider syntactic categories to be logical formulas
- As such, they can be atomic or complex (not just plain A, B, a, b etc.).
- They are related by means of the derivability relation ($\Rightarrow$)
- To recognize that a string/structure is of a certain category reduces to prove the formulas corresponding to the structure and the category are in a derivability relation $\Gamma \Rightarrow A$

The slogan is:

> "Parsing as deduction"

1.8. Function/Implication and NL

We have seen that words (and phrases) can be interpreted as sets of entities or set of properties, etc.. Alternatively, one can assume a functional perspective and interpret, for example, "student" as a function from individual (entities) to truth values, $student(monika) = 1$, $student(rajeev) = 0$.

The shift from the set-theoretical to the functional perspective is made possible by the fact that the **sets and their characteristic functions amount to the same thing**:

> if f_X is a function from Y to $\{0, 1\}$, then $X = \{y \mid f_X(y) = 1\}$. In other words, the assertion '$y \in X$' and '$f_X(y) = 1$' are equivalent.

E.g. `run`: $D_e \to D_t$; `know`: $D_e \to (D_e \to D_t)$; `every man`: $(D_e \to D_t) \to D_t$

Hence, we need to "represent" functions and be able to "reason" on (compose) them.

2. Pure logic of Residuation

The minimum we need to speak about functions is $\rightarrow$ that is governed by the principle below.

$$(a) \quad p, q \Rightarrow r \quad \text{iff} \quad p \Rightarrow q \rightarrow r$$

But linguistic structures are:

- not commutative, hence we need to have a right ($A\backslash B$ –if A then B) and a left implication (B/A – B if A).
- not associativity –we cannot freely change their bracketing.
- sensitive to the occurrence of words (we cannot freely reduce or add them), hence no contraction and weakening is allowed.

Hence, the **minimum logic** we need is the **logic of residuation** expressed in (a).

2.1. Residuation

Let $\langle C, \leq_3 \rangle$ be a third partially ordered set, a triple of functions (f, g, h) such that $f : A \times B \longrightarrow C$, $g : A \times C \longrightarrow B$, $h : C \times B \longrightarrow A$ forms a residuated triple if

$$[RES_2] \quad \forall x \in A, y \in B, z \in C \begin{pmatrix} x \leq_1 h(z, y) \\ \text{iff} \\ f(x, y) \leq_3 z \\ \text{iff} \\ y \leq_2 g(x, z) \end{pmatrix}$$

For instance

$$[RES_2] \quad \forall x \in A, y \in B, z \in C \begin{pmatrix} x \leq_1 \frac{z}{y} \\ \text{iff} \\ x \times y \leq_3 z \\ \text{iff} \\ y \leq_2 \frac{z}{x} \end{pmatrix}$$

Similarly, we can speak of n-ary residuated operators.

2.2. Residuation: Tonicity and Composition

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

i)Tonicity: $f(+,+), g(-,+)$ and $h(+,-)$

where $+$ means, it preserve the order of its argument (upward monotonic).
e.g. $f(a, b) \leq f(c, d)$ if $a \leq c$ and $c \leq d$

where $-$ means, it reverses the order of its argument (downward monotonic).
e.g. $g(c, b) \leq f(a, d)$ if $a \leq c$ and $c \leq d$

$$ii)\text{Composition}: \quad \forall x \in A, y \in B, z \in C \begin{pmatrix} f(x, g(x, z)) \leq_3 z \\ \text{and} \\ y \leq_2 g(x, f(x, y)) \\ \text{and} \\ f(h(z, y), y) \leq_3 z \\ \text{and} \\ x \leq_1 h(f(x, y), y) \end{pmatrix}$$

3. Non-associative Lambek Calculus (NL)

NL logical and structural language

$$\begin{array}{ll} \mathsf{FORM} ::= & \mathsf{ATOM} \mid \mathsf{FORM} \otimes \mathsf{FORM} \mid \mathsf{FORM}/\mathsf{FORM} \mid \mathsf{FORM}\backslash\mathsf{FORM} \\ \mathsf{X} ::= & \mathsf{FORM} \mid \mathsf{X}, \mathsf{X} \end{array}$$

Remark In sequent calculi we need both logical and structural language, the re-write rule below establish the connection between $\otimes$ and its structural proxy ,:

$$\frac{A, B \Rightarrow C}{A \otimes B \Rightarrow C}$$

Proof Theory For each logical operator $(*)$, Gentzen Sequents Calculi consist of a logical rule introducing the $*$ on the left ([$*$L)]) and on the right ([$*$R)]) of the $\Rightarrow$.

Let $\Delta, \Gamma, \ldots$ and $A, B, \ldots$ stand for structures and formulas, respectively.

$$\frac{A, \Delta \Rightarrow B}{\Delta \Rightarrow A\backslash B} (\backslash \mathrm{R}) \qquad [RES_2] \quad \forall x \in A, y \in B, z \in C \begin{pmatrix} f(x, y) \leq_3 z \\ \text{if} \\ y \leq_2 g(x, z) \end{pmatrix}$$

This rule encodes half of the residuation condition holding between $\backslash$ and , i.e. the structural proxy of $\otimes$.

3.1. Non-associative Lambek Calculus (Cont'd)

The other half of the residuation condition is compiled in the [\L] and [/L].

$$\frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[A/B,\Delta] \Rightarrow C} \qquad \frac{\Delta \Rightarrow B \quad \Gamma[A] \Rightarrow C}{\Gamma[\Delta, B\backslash A] \Rightarrow C}$$

The **composition** property is an instantiation of the rules above, e.g.

$$\left(\begin{array}{c} f(x, g(x,z)) \leq_3 z \\ \text{is} \\ (A/B) \otimes B \Rightarrow A \end{array} \right)$$

where $\Delta = B$, $C = A$ and Γ is empty.

3.2. (Binary) Residuated System: **NL**

$$\frac{}{A \Rightarrow A} \text{ (axiom)}$$

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

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

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

Tonicity			
upward mon.	$+/$	$+\otimes+$	$\backslash+$
downward mon.	$/-$		$-\backslash$

3.3. Logical Grammar: Lexicon

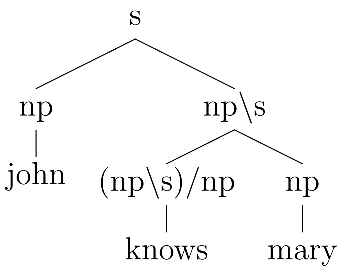
```
CFG Lexicon                Rules                            S
    NP --> john            S --> NP VP                    /     \
    IV --> walks           VP --> IV                     /       VP
    TV --> knows           VP --> TV NP                 /      /    \
    NP --> mary                                        NP    TV     NP
                                                       john knows mary
```

NL Lexicon (Categorization):

John, Mary: np walks: $np\backslash s$ knows: $(np\backslash s)/np$

3.4. Logical Grammar: Rules (Composition)

NL Rules (Composition): (/L) and (\L)



$$\dfrac{np \Rightarrow np \qquad \dfrac{np \Rightarrow np \quad s \Rightarrow s}{np, (np\backslash s) \Rightarrow s}\,(\backslash L)}{\underbrace{np}_{\text{john}}, \underbrace{((np\backslash s)/np)}_{\text{knows}}, \underbrace{np}_{\text{mary}} \Rightarrow s}\,(/L)$$

3.5. Advantages and Limits

Advantages

- it identifies in the **residuation principle the core of natural language structure**.
- it reduces cross-linguistic variations to variations w.r.t. structural rules and lexicon.
- it captures the **syntax-semantics** interface in a clear way: NL corresponds to λ-calculus (**Curry-Howard correspondence**). Hence, meaning representation is built as their by-product by simply by labeling the derivations with the corresponding λ-terms.

Limits It does not account for **non local scope construal** and **long distance dependencies**.

4. Going on research: Bi-Lambek & Grishin

Aim We want to extend the expressivity of NL to overcome the undergeneration problem (avoiding overgeneration) by shopping in the algebraic structure it lives in.

Ingredients

- (n-ary) Residuated operators
- (n-ary) Dual Residuated operators
- (n-ary) Galois Operators
- Connection between the different families of operators

Receipt

- increase the expressivity step by step to grasp the minimal logic needed.

4.1. Dual Residuation

Recall Let $\langle C, \leq_3 \rangle$ be a third partially ordered set, a triple of functions (f, g, h) such that $f : A \times B \longrightarrow C$, $g : A \times C \longrightarrow B$, $h : C \times B \longrightarrow A$ forms a residuated triple if

$$[RES_2] \quad \forall x \in A, y \in B, z \in C \begin{pmatrix} x \leq_1 h(z, y) \\ \text{iff} \\ f(x, y) \leq_3 z \\ \text{iff} \\ y \leq_2 g(x, z) \end{pmatrix}$$

Similarly a triple of functions (f, g, h) forms a dual residuated triple if

$$[DRES_2] \quad \forall x \in A, y \in B, z \in C \begin{pmatrix} h(z, y) \leq_1 x \\ \text{iff} \\ z \leq_3 f(x, y) \\ \text{iff} \\ g(x, z) \leq_2 y \end{pmatrix}$$

4.2. Bi-Lambek

Language

$$\begin{aligned}\mathsf{FORM} ::= \; & \mathsf{ATOM} \mid \mathsf{FORM} \otimes \mathsf{FORM} \mid \mathsf{FORM}/\mathsf{FORM} \mid \mathsf{FORM}\backslash\mathsf{FORM} \\ & \mathsf{FORM} \oplus \mathsf{FORM} \mid \mathsf{FORM} \oslash \mathsf{FORM} \mid \mathsf{FORM} \obslash \mathsf{FORM} \\ \mathsf{X} ::= \; & \mathsf{FORM} \mid \mathsf{X}, \mathsf{X}\end{aligned}$$

Composition

$$A \otimes (A\backslash B) \Rightarrow B \qquad B \Rightarrow A \oplus (A \obslash B)$$

Tonicity

Tonicity						
upward mon.	$+/$	$+ \otimes +$	$\backslash +$	$+ \oslash$	$+ \oplus +$	$\obslash +$
downward mon.	$/-$		$-\backslash$	$\oslash -$		$- \obslash$

Problem No communication between the two families of operators. The expressivity of each logic does not increase.

4.3. Grishin: Inequalities

Grishin identifies a class of system obtained from given algebraic systems by **adding certain inequalities to the axioms**. In particular, he looks at associative Lambek calculus (L) and its bi-counterpart (bi-L) enriched with neutral elements. The generalization proceeds as below.

- We have 6 binary operations (3 res, 3 dual-res, w), hence 12 cases ($w?, ?w$).
- These 12 operators are divided into (i) left vs. right based on where they live w.r.t. to $\leq$ ($\Rightarrow$); and (ii) upward ($|w| = 0$) vs. downward ($|w| = 1$) monotonic based on the monotonicity of their argument (the ?).
- Grishin gives 6 inequality schema, $a^\mu x = awx$ if $\mu = w?$, and $a^\mu x = xwa$ if $\mu =?w$.

1.	$\forall a, b, c(a^\mu, b^\lambda c \leq b^\lambda a^\mu c)$	4.	$\forall a, b, c((a^{\lambda^\perp} b)^{\mu^{*\perp}} c \leq_{\lvert\mu^*\rvert} b^{\mu^{*\perp}} a^\lambda c)$
2.	$\forall a, b, c(b^\lambda a^{\mu^\perp} c \leq_{\lvert\mu\rvert} a^{\mu^\perp} b^\lambda c)$	5.	$\forall a, b, c(a^{\lambda^{*\perp}} b^\mu c \leq_{\lvert\lambda^*\rvert} b^{\mu^\perp} a^{\lambda^{*\perp}} c)$
3.	$\forall a, b, c(a^{\lambda^\perp} b^\mu c \leq_{\lvert\lambda\rvert} b^\mu a^{\lambda^\perp} c)$	6.	$\forall a, b, c((c^{\mu^{*\perp}} b)^\mu a \leq (c^{\lambda^{*\perp}} a)^\lambda b)$

μ	$?w$	$w?$
μ^*	$w?$	$?w$

μ	$\otimes ?$	$? \backslash$	$?/$	$\oplus ?$	$? \oslash$ (reversed)	$? \oslash$
$\mu^\perp$	$\backslash ?$	$?/$	$? \otimes$	$\oslash ?$ (reversed)	$\oslash ?$	$? \oplus$

	$\varepsilon = 0$	$\varepsilon = 1$
$x \leq_\varepsilon y$	$x \leq y$	$y \leq x$

4.4. Grishin: Classes of inequalities

- Grishin proves that these 6 inequalities (of formulas) are mutually equivalent (interderivable) given residuation (and dual-residuation), when both $|\lambda| = 0$ and $|\mu| = 0$ (upward monotonic).
- The 6 mutually equivalent formulas identify classes of equivalent postulates.
- Out of the 12 cases of operators the combination of the upward monotonic ones (viz. 4 left $\{\otimes ?, ?\otimes, \obslash ?, ?\oslash\}$ and 4 right $\{\oplus ?, ?\oplus, \backslash ?, ?/\}$) gives 16 classes of 6 mutually equivalent postulates, namely:

  1. 4: associativity of res. operators (II) and of dual-res. (III);
  2. 4: 3-commutativity of res. operators (II') and of dual-res. (III');
  3. 4: mixed associativity of res. & dual-res operators (I and IV);
  4. 4: mixed commutativity of res. & dual-residuation (I' and IV').

  Each group of 4 classes consists of 2 classes and their symmetric ($\sim$) cases –e.g. $(\backslash)^{\sim} = /$ and $(\obslash)^{\sim} = \oslash$.
  The N' are obtained by keeping the μ and switching to the $(\lambda)^{\sim}$ of the N.

4.5. Remarks: inequalities strength

- Commutativity follows from II' and III' (3-commutativity), e.g. postulate 3. $a \otimes (b \otimes c) \leq b \otimes (c \otimes a)$, take $c = 1$, $a \otimes (b \otimes 1) \leq b \otimes (1 \otimes a) = a \otimes b \leq b \otimes a$.

- Class IV is weaker than the other classes (???).

  1. Class IV (mix. ass. of res. and dual res) is provable from the having $a \backslash b =_{def} \neg a \oplus b$, residuation, classes I and III.
     If $a \backslash b = \neg a \oplus b$, postulate 2. $a \backslash (c \oplus b) \leq (a \backslash c) \oplus b$ is a valid statement, viz. $\neg a \oplus (c \oplus b) \leq (\neg a \oplus c) \oplus b$, and so do the other equivalent postulates.

4.6. Remarks: displayable equalities

displayable inequality: in each side of the $\leq$, the formula is built out of operators living on the same side of the $\Rightarrow$ in Display Logic.

- Each of the classes formed by taking both $|\mu|$ and $|\lambda|$ as 0 (upw. mon) contains one displayable inequality (two if they are mixed —one for each side of $\Rightarrow$):

  [ass. and 3-com] In group II and $(\text{II})^{\sim}$ (its symmetric), and in II' and $(\text{II'})^{\sim}$ (resp. III and $(\text{III})^{\sim}$, and III' and $(\text{III'})^{\sim}$) they are the postulates 3. (resp. 2.).

  [mix-ass. and mix-com] In group I and IV (resp. I' and IV') they are the postulates 2. and 3. (Similarly, for the symmetric cases).

- Equalities of these postulates are obtained by combining two classes:

  by II plus $(\text{II})^{\sim}$ the inequalities 3. become: $a \otimes (c \otimes b) = (a \otimes c) \otimes b$.

  by II' plus $(\text{II'})^{\sim}$ the inequalities 3. become: $a \otimes (b \otimes c) = b \otimes (c \otimes a)$.

  Similarly, for the $\oplus$ by III plus $(\text{III})^{\sim}$ and III' plus $(\text{III'})^{\sim}$

  by I plus IV (resp. I' and IV') 2. become: $a \oplus (c/b) = (a \oplus c)/b$, (resp. $b \oplus (c \backslash a) = a \backslash (c \oplus b)$) and 3. become: $a \oslash (c \otimes b) = (a \oslash c) \otimes b$ (resp. $a \oslash (b \otimes c) = b \otimes (a \oslash c)$).

  (Similarly, for the symmetric cases.)

5. Where we are and where we are going

- **Hierarchy** A Residuated Logics for linguistic analysis.
- **Completness** It has been proved for Bi-NL + Groups IV and IV' (Kurtonina, Moortgat and Goré)
- **Proof System**
  - Display Logic (of course).
  - Sequent Calculus: but we are still checking whether cut is admissible.
  - Sequent Calculus based on de Groote'99 approach (context with a hole)
- **Complexity** de Groote's approach could be used to show that Bi-NL (plus Group IV ...) is decidable in polynomial time. (started)
- **Curry-Howard Correspondence** to be done!
- **Galois** to be done. (started.)
- **Unary** Unary Residuated operators (Kurtonina Moortgat 95); Unary Galois (Areces, Bernardi, Moortgat'00). Still to be done: communication. (started.)