

Calcolo di Lambek, logica lineare e linguistica

Claudia Casadio – Università “Gabriele D’Annunzio” Chieti e Pescara –
casadio@unich.it

1. Introduzione

La logica lineare (Girard, 1987) rappresenta un’area di ricerca nuova ed interessante, sia dal punto di vista teorico, che dal punto di vista delle sue possibili applicazioni, in particolare nell’ambito delle discipline computazionali e linguistiche. Per quanto riguarda la dimensione del linguaggio, si tratta di un territorio per molti aspetti inesplorato, come è stato sottolineato da J. Y. Girard (1995), che si presta allo scambio interdisciplinare ed alla produzione di nuovi modelli interpretativi. La versione della logica lineare chiamata logica non commutativa (Abrusci, 1991; Abrusci e Ruet, 1999) nasce dall’esigenza di sviluppare una logica sensibile non solo al numero e all’uso delle risorse, ma anche al loro ordine, proprietà distintiva delle espressioni e dei contesti linguistici; questa logica ammette sia una formulazione classica, che una formulazione intuizionista. È stato lo studio delle proprietà della formulazione intuizionista che ha portato alla scoperta delle sue relazioni con il calcolo per il linguaggio naturale o *Syntactic Calculus* introdotto nel 1958 dal matematico canadese J. Lambek (1958): il Calcolo di Lambek è stato, ed è tuttora, oggetto di studio nell’ambito della logica, dell’informatica e della linguistica teorica.¹

L’aspetto forse più promettente degli studi recenti in questo campo è rappresentato dalla diretta corrispondenza individuata tra i due connettivi del Calcolo di Lambek, *left slash*: “\” e *right slash*: “/”, che presiedono alla generazione dei tipi sintattici, e le due implicazioni: *post-implicazione* “—o” e *retro-implicazione* “o—” della logica lineare non-commutativa, a loro volta formulabili nei termini dei connettivi *par*: “ \wp ”, *times*: “ \otimes ” della logica lineare moltiplicativa.² In particolare, è stato dimostrato che il Calcolo di Lambek corrisponde al frammento moltiplicativo della logica lineare non-commutativa intuizionista senza esponenziali (Abrusci, 1991; Casadio e Lambek, 2001; Abrusci, 2002).

¹Per un quadro degli studi sul Calcolo di Lambek e le sue recenti applicazioni a livello logico e linguistico si richiamano Van Benthem (1988), Van Benthem (1991), Morrill (1994), Morrill (2010) e Moortgat (1997)

²Precisamente, nel calcolo PNCL (*Pure Non-commutative Classical Linear propositional logic* (Abrusci, 1991)) la post-implicazione lineare “—o” corrisponde all’operazione “\” del Calcolo di Lambek, e la retro-implicazione lineare “o—” corrisponde alla operazione “/”. Sulla base della definizione delle due implicazioni in PNCL, valgono le seguenti equivalenze: $A \text{—o} B = A^\perp \wp B$ e $B \text{o—} A = B \wp \perp A$, dove $()^\perp$ (*post-negazione*) e $\perp()$ (*retro-negazione*) sono le due negazioni ammesse dalla logica lineare non-commutativa.

La logica lineare non-commutativa introduce una prospettiva dinamica in base a cui la comunicazione linguistica è analizzata come un processo che può svilupparsi ed essere afferrato da diversi punti di vista. Questa prospettiva dinamica consente di considerare sotto una nuova luce le grammatiche logiche per il linguaggio naturale, come la grammatica di Montague o le grammatiche categoriali che, richiamandosi al principio fregeano di composizionalità del significato, assegnano un ruolo basilare alla relazione funzione-argomento nella classificazione ed organizzazione delle risorse linguistiche (Van Benthem, 1991; Moortgat, 1997; Morrill, 2010). Le grammatiche dei tipi logici, assumendo l'impostazione di teoria delle dimostrazioni in base a cui i processi sintattici (e semantici) di una lingua possano essere rappresentati nei termini di inferenze logiche, si propongono di ottenere l'insieme delle espressioni ben formate, ed in particolare delle frasi, come risultato di appropriate dimostrazioni. In questo quadro, ha particolare rilievo il metodo delle reti di dimostrazione (*proof nets*), introdotti nell'ambito della logica lineare per fornire una rappresentazione geometrica dei processi logici messi in atto da una dimostrazione. L'impiego dei *proof nets* e la concezione ad essi associata di una geometria dell'interazione, hanno rappresentato una scelta metodologica importante che consente di afferrare e comprendere in modo immediato le proprietà logiche delle dimostrazioni, fornendo inoltre uno strumento proficuo per l'analisi formale delle lingue naturali.

2. La distinzione funzione argomento

Le grammatiche dei tipi logici si ispirano a due fondamentali orientamenti della analisi semantica del linguaggio naturale: la teoria delle categorie di significato formulata all'inizio del secolo da Edmund Husserl e la teoria composizionale del significato di Gottlob Frege.³ Le *Bedeutungskategorien* di Husserl richiamano la concezione aristotelica delle parti del discorso, ma in un nuovo quadro teorico che rimanda alle ricerche sul significato di Bolzano e Frege. Nell'analisi di Husserl il discorso si struttura in parti che esibiscono due fondamentali aspetti relativamente al modo in cui si costituisce il loro significato: alcune parti, come nomi o enunciati, sono significative di per sé stesse, ovvero sono dotate di un significato indipendente. Altre parti, al contrario, non sono significative di per sé stesse, ma hanno bisogno di essere completate per dare luogo ad una espressione dotata di significato; si tratta delle parole sincategorematiche, come *e*, *o*, *è*, la cui funzione linguistica dipende dal significato dell'intero enunciato (o contesto) che contribuiscono a determinare. La connessione dei significati, che è alla base dell'articolazione linguistica, è il risultato della combinazione di parti incomplete con parti adeguate a completarle.

³Il tema delle categorie di significato è analizzato nelle *Ricerche Logiche* di Husserl, in particolare nella Terza e nella Quarta Ricerca (Husserl, 1913); per una analisi della teoria del significato di Husserl si veda Casadio (2011). Per la teoria del significato di Frege si vedano Frege (1892) e Frege (1893/1903) e per la corrispondenza con Husserl Frege (1976).

La distinzione tra significati indipendenti o completi vs. significati non indipendenti o incompleti è molto vicina al modo in cui Frege caratterizza la nozione di funzione come espressione 'insatura' che richiede di essere completata da argomenti appropriati per dare luogo ad un certo valore. Frege riconosce, inoltre, una relazione diretta tra espressioni complete (nomi propri o enunciati) e funzioni, dal momento che una funzione, insatura, può sempre essere ricavata da una espressione completa, mediante l'individuazione di una sua parte come variabile (Frege, 1893/1903). Raccogliendo in un unico quadro teorico questi risultati, come avviene nell'ambito delle ricerche sviluppate dalla Scuola Logica Polacca,⁴ otteniamo alcuni principi che svolgono un ruolo basilare nell'analisi semantica del linguaggio naturale: (i) vi sono parti del discorso che sono incomplete; (ii) tali parti del discorso corrispondono a funzioni che possono essere rappresentate, mediante appropriate notazioni algebriche o logiche, come categorie funtoriali; (iii) l'articolazione linguistica è il risultato di processi di saturazione delle categorie delle funzioni con gli argomenti appropriati, dando luogo a valori che sono entità complete, in particolare le proposizioni.

Poichè una funzione è identificata dalle categorie dei suoi argomenti e dalla categoria del valore che risulta per quegli argomenti, la notazione algebrica introdotta da Kazimierz Ajdukiewicz⁵ si è dimostrata particolarmente adeguata nell'ambito dei modelli orientati ad utilizzare funzioni per analizzare formalmente il linguaggio naturale. Tale notazione rappresenta i simboli di funzione (funtori) mediante indici frazionari. L'indice frazionario associato ad un funtore prende al denominatore gli indici degli argomenti della funzione corrispondente e al numeratore l'indice del valore assunto dalla funzione. Si ottiene così la notazione standard di una grammatica categoriale in cui A/B è il simbolo di una funzione che per un argomento di tipo B assume un valore di tipo A :

$$\frac{\text{funtore} : A/B \quad \text{argomento} : B}{A}$$

corrispondente all'operazione di applicazione funzionale che in una grammatica a costituenti immediati è rappresentata dalla regola di riscrittura (regola di cancellazione o eliminazione):

$$A/B \quad B \rightarrow A$$

Ad esempio, un verbo transitivo come *dare*, è analizzato come un'espressione incompleta corrispondente ad una funzione che richiede due argomenti: l'oggetto diretto: SN_1 e l'oggetto indiretto: SN_2 . L'applicazione della categoria che corrisponde a tale funzione, alle categorie degli argomenti, genera come valore la categoria dell'espressione risultante, il predicato verbale:

⁴L'approfondimento e la formalizzazione della teoria del significato di Husserl ha caratterizzato la Scuola Logica Polacca in un ambito di ricerche dedicate ai fondamenti della matematica ed all'analisi formale del linguaggio. Si veda in proposito McCall e Ajdukiewicz (1967).

⁵Si veda Ajdukiewicz (1935), il noto saggio dedicato alla connessione sintattica, comparso sul primo numero della rivista *Studia Philosophica*.

dare: $(SV/SN_2)/SN_1$, un libro: SN_1 , a Maria: $SN_2 \rightarrow$
 dare un libro a Maria: SV .

In base alla distinzione tra espressioni complete ed incomplete, Frege divide i segni linguistici in due categorie: (i) la categoria dei nomi propri, a cui appartengono le espressioni complete (che non hanno posti per argomento); (ii) la categoria dei nomi di funzione, a cui appartengono le espressioni incomplete (che hanno uno o più posti per argomento). La prima classe include i nomi di oggetti (nomi propri, descrizioni definite), gli enunciati (come nomi dei particolari oggetti che sono i valori di verità) ed, in generale, i nomi dei corsi di valori delle funzioni.

La nozione di funzione può essere estesa a qualsiasi segno linguistico in cui un costituente è lasciato indeterminato: una funzione è ottenuta rimuovendo una o più occorrenze di un'espressione completa (un nome proprio) da un'espressione completa. Si hanno così gli ingredienti essenziali per definire un insieme di tipi logici per il linguaggio naturale a partire da una base finita costituita dai due tipi N (nomi) ed S (enunciati), come nel calcolo formulato da Geach (1972), in cui, rimuovendo una espressione di tipo $:N$ da un enunciato di tipo $:S$, si ottiene il tipo $:S/N$ di una funzione di primo livello corrispondente ad un predicato:

Socrate $:N$ sogna $:S/N \rightarrow$ Socrate sogna $:S$

mentre, rimuovendo una espressione di tipo $:S/N$ da un enunciato, si ottiene una funzione di secondo livello di tipo $:S/(S/N)$, come nel caso dei sintagmi quantificazionali:

Ogni uomo $:S/(S/N)$ sogna $:S/N \rightarrow$ Ogni uomo sogna $:S$

Queste sono le assunzioni su cui sono state fondate le grammatiche categoriali di Ajdukiewicz (Ajdukiewicz, 1935), Bar-Hillel (Bar-Hillel, 1953; Hillel et al., 1960) e il calcolo sintattico di Lambek (Lambek, 1958).

3. Calcolo di Lambek

Lambek presenta il suo calcolo sintattico in due noti articoli: *The mathematics of sentence structure* del 1958 (Lambek, 1958) e *On the calculus of syntactic types* del 1961 (Lambek, 1961). Il calcolo fornisce una procedura effettiva per determinare le frasi di una lingua (naturale o formale) e lo studio delle sue proprietà logiche e matematiche ha dimostrato che può essere applicato in modo proficuo a diversi ambiti di indagine dalla linguistica, alla analisi computazionale alla traduzione automatica.

Il Calcolo di Lambek (L) è un sistema deduttivo che definisce un insieme di tipi logici o sintattici ed è chiuso rispetto alle tre operazioni binarie $A \bullet B$ (prodotto), C/B (C over B), $A \setminus C$ (A under C), in cui M è un sistema moltiplicativo (monoide) con identità I (per ogni $A, B, C \in M$):

- a. $A \bullet B = \{x.y \in M \mid x \in A \wedge y \in B\}$
- b. $C/B = \{x \in M \mid \forall y \in B, x.y \in C\}$
- c. $A \setminus C = \{y \in M \mid \forall x \in A, x.y \in C\}$

I teoremi elencati in (3) sono dimostrabili in L sulla base degli assiomi in (2) e delle regole di inferenza in (1) (per ogni $A, B, C \in M$):

- (1) a. $A \bullet B \rightarrow C \leftrightarrow A \rightarrow C/B$ (*implicazione o divisione a destra*)
 b. $A \bullet B \rightarrow C \leftrightarrow B \rightarrow A \backslash C$ (*implicazione o divisione a sinistra*)
 c. $A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C$ (*transitività*)
- (2) a. $X \rightarrow X$ (*identità*)
 b. $(A \bullet B) \bullet C \leftrightarrow A \bullet (B \bullet C)$ (*associativa*)
- (3) a. $(A/B) \bullet B \rightarrow A$ (*cancellazione a destra*)
 b. $A \bullet (A \backslash B) \rightarrow B$ (*cancellazione a sinistra*)
 c. $B \rightarrow (A/B) \backslash A$ (*type raising*)
 d. $B \rightarrow A/(B \backslash A)$ (*type raising*)
 e. $(A \backslash B)/C \leftrightarrow A \backslash (B/C)$ (*associatività*)
 f. $(A/B)/C \leftrightarrow A/(C \bullet B)$ (*def. \bullet*)
 g. $(A/B) \bullet (B/C) \rightarrow (A/C)$ (*composizione*)
 h. $A/B \rightarrow (A/C)/(B/C)$ (*Geach law*)

Il sistema moltiplicativo M non è necessariamente associativo: se si ammette la proprietà associativa (2b) come in Lambek (1958), M è un semigruppato che consente di formare liberamente costituenti a destra o a sinistra (*unbracketed strings*); se non viene ammessa la proprietà associativa si ottiene il sistema più restrittivo formulato in Lambek (1961) e studiato in numerose applicazioni linguistiche (si vedano in proposito Moortgat (1997), Morrill (2010) e Morrill (1994)).

Il calcolo sintattico L (generato dal sistema moltiplicativo M) può essere pensato come un sistema universale di regole che si applica ad un dizionario specificato per una certa lingua (naturale o formale)⁶ definendo appropriati insiemi di parole (o stringhe di parole) chiamati tipi sintattici, in cui S è il tipo degli enunciati dichiarativi o frasi della lingua, N è il tipo dei nomi come *Mary, John*. Partendo da questi due tipi di base, applicando le regole del calcolo L , possono essere derivati i tipi dei vari costituenti di un enunciato (elencati nel Lessico B), come NP (sintagma nominale), VP (sintagma verbale), PP (sintagma preposizionale), ecc., nei cui termini si può dimostrare se una certa sequenza di tipi sintattici, assegnata ad una stringa di parole della lingua, è una frase grammaticale della lingua.

⁶Facciamo qui riferimento alla lingua inglese, daremo quindi alcuni esempi relativi alla lingua italiana.

- a. $\mathbf{B} = [\mathbf{N}, \mathbf{NP}, \mathbf{S}]$
- b. if $\mathbf{Mary} \rightarrow \mathbf{NP}$ and $\mathbf{Mary works} \rightarrow \mathbf{S}$,
then $\mathbf{works} \rightarrow \mathbf{NP} \backslash \mathbf{S}$;
- c. if $\mathbf{apple} \rightarrow \mathbf{N}$ and $\mathbf{an apple} \rightarrow \mathbf{NP}$,
then $\mathbf{an} \rightarrow \mathbf{NP} / \mathbf{N}$;
- d. if $\mathbf{Mary ate an apple} \rightarrow \mathbf{S}$ and $\mathbf{Mary} \rightarrow \mathbf{NP}$,
then $\mathbf{ate an apple} \rightarrow \mathbf{NP} \backslash \mathbf{S}$;
if $\mathbf{an apple} \rightarrow \mathbf{NP}$, then $\mathbf{ate} \rightarrow (\mathbf{NP} \backslash \mathbf{S}) / \mathbf{NP}$.

Di seguito presentiamo la derivazione di alcune frasi della lingua italiana, usando la procedura inferenziale in base a cui ogni applicazione di una regola del calcolo L è indicata dal simbolo scritto a fianco della linea orizzontale.

CALCOLO SINTATTICO L: *Applicazione a destra / vs. a sinistra *

- a. *Maria mette una tovaglia nuova sul tavolo.*
- b. **mette (una tovaglia nuova) (sul tavolo)**

$$\frac{\frac{\frac{(\mathbf{VP}/\mathbf{PP})/\mathbf{NP} \quad \mathbf{NP}}{\mathbf{VP}/\mathbf{PP}} \quad / \quad \mathbf{PP}}{\mathbf{VP}} \quad /}{\mathbf{VP}} \quad \backslash$$
- c. **Maria (mette una tovaglia nuova sul tavolo)**

$$\frac{\mathbf{NP} \quad \mathbf{VP} = \mathbf{NP} \backslash \mathbf{S}}{\mathbf{S}} \quad \backslash$$
- a. *Gianni corre velocemente sotto la pioggia.*
- b. **corre velocemente (sotto la pioggia)**

$$\frac{\frac{\mathbf{VP} \quad \mathbf{VP} \backslash \mathbf{VP}}{\mathbf{VP}} \quad \backslash \quad \mathbf{VP} \backslash \mathbf{VP}}{\mathbf{VP}} \quad \backslash$$
- c. **Gianni (corre velocemente sotto la pioggia)**

$$\frac{\mathbf{NP} \quad \mathbf{VP} = \mathbf{NP} \backslash \mathbf{S}}{\mathbf{S}} \quad \backslash$$

CALCOLO SINTATTICO L: *Type Raising o Expansion: EXP*

- a. *Molte ragazze partono domani.*
- b. **(Molte ragazze) (partono domani)**

$$\frac{\frac{\mathbf{NP}}{\mathbf{S}/(\mathbf{NP} \backslash \mathbf{S})} \quad \mathbf{EXP} \quad \mathbf{NP} \backslash \mathbf{S}}{\mathbf{S}} \quad /$$

4. Logica Lineare Non-commutativa

Il Calcolo di Lambek L ha una traduzione naturale nella logica lineare non-commutativa (*Non-commutative Linear Logic*: NLL), un sistema della logica lineare che si ottiene escludendo le regole strutturali di indebolimento (*weakening*): $A \vdash A, A$; contrazione (*contraction*): $A, A \vdash A$; scambio (*exchange*): $A, B \vdash B, A$ (vedi Abrusci (1991), Abrusci (2002) e Abrusci (2014)). Precisamente, il Calcolo di Lambek corrisponde al frammento moltiplicativo MNLL e può ricevere sia una formulazione intuizionista che una formulazione classica, come è dimostrato in Abrusci (1991). Il sistema completo NLL ha il seguente alfabeto (Abrusci, 1991; Abrusci e Ruet, 1999):

Variabili Proporzionali :	P, Q, \dots
Costanti Logiche (<i>bottom, one, true, zero</i>):	$\perp, 1, T, 0$
Connettivi binari :	moltiplicativi (<i>times, par</i>): \otimes, \wp additivi (<i>with, plus</i>): $\&, \oplus$
Connettivi unari (<i>of course, why not</i>):	$!, ?$
Simbolo di sequente :	\vdash
Simboli ausiliari :	$(,), \dots$

NLL e il suo frammento moltiplicativo MNLL sono sistemi classici perchè i sequenti possono avere più di una conclusione e valgono leggi logiche classiche come le leggi *De Morgan* e una particolare formulazione delle leggi della *Negazione*. Si ha la seguente definizione di sequente nel calcolo ad una parte (*one-sided sequent calculus*):

$\vdash \Gamma$ è un sequente sse Γ è una sequenza finita di formule di NLL (MNLL) .

Una proprietà distintiva della logica lineare non-commutativa è la presenza di due negazioni:

- *post*-negazione di $A = A^\perp$
- *retro*-negazione di $A = \perp A$

Esse sono introdotte mediante una definizione metalinguistica (Abrusci, 1991; Abrusci, 2002). Su questa base, l'insieme delle formule di NLL è definito induttivamente (per la corrispondenza con il Calcolo di Lambek, limitiamo la definizione al frammento moltiplicativo MNLL):

- ciascuna variabile proposizionale è una formula di MNLL ;
- se A è una formula di MNLL, $A^\perp, \perp A$ sono formule di MNLL ;
- se A e B sono formule di MNLL, $A \otimes B$ è una formula di MNLL ;
- se A e B sono formule di MNLL, $A \wp B$ è una formula di MNLL ;
- nient'altro è una formula di MNLL .

Formule che contengono le due implicazioni “ \multimap ” and “ \multimap ” (corrispondenti alle operazioni “ \backslash ”, “ $/$ ” del Calcolo di Lambek) possono essere tradotte in formule di MNLL contenenti il connettivo \wp e una delle due negazioni. Se A e B sono formule di MNLL,

- $A \multimap B$ è tradotta dalla formula $A^\perp \wp B$ di MNLL ;
- $B \multimap A$ è tradotta dalla formula $B \wp^\perp A$ di MNLL .

Queste traduzioni richiamano la definizione del connettivo di *implicazione* della logica proposizionale nei termini della *disgiunzione* e della *negazione*, e svolgono un ruolo fondamentale nella costruzione delle dimostrazioni assicurando la transizione dai tipi logici implicazionali del Calcolo di Lambek a tipi classici definiti nei termini dei connettivi duali *par*: \wp e *times*: \otimes , e delle due negazioni lineari. Una proprietà caratteristica dei sistemi NLL, MNLL è l’architettura simmetrica in base a cui le leggi classiche di De Morgan valgono per insiemi duali di connettivi ed in particolare per i connettivi \wp e \otimes , che qui riportiamo :

Leggi De Morgan

$$\begin{aligned} (A \wp B)^\perp &= B^\perp \otimes A^\perp, \\ \perp(A \wp B) &= \perp B \otimes \perp A, \\ (A \otimes B)^\perp &= B^\perp \wp A^\perp, \\ \perp(A \otimes B) &= \perp B \wp^\perp \perp A. \end{aligned}$$

La distribuzione della negazione sulle due sottoformule ha l’effetto di cambiare il connettivo nel suo duale invertendo l’ordine delle formule:

$$\begin{aligned} (A^\perp \wp B)^\perp &= B^\perp \otimes A^{\perp\perp} & (B \wp^\perp \perp A)^\perp &= A \otimes B^\perp, \\ \perp(A^\perp \wp B) &= \perp B \otimes A & \perp(B \wp^\perp \perp A) &= \perp\perp A \otimes \perp B, \\ (A^\perp \otimes B)^\perp &= B^\perp \wp A^{\perp\perp} & (B \otimes \perp A)^\perp &= A \wp B^\perp, \\ \perp(A^\perp \otimes B) &= \perp B \wp^\perp A & \perp(B \otimes \perp A) &= \perp\perp A \wp^\perp \perp B. \end{aligned}$$

La negazione lineare è una operazione involutiva: nel calcolo dei sequenti a due parti (*two-sided sequent calculus*), la retro (post)-negazione di una formula di MNLL che occorre in un sequente, ha l’effetto di spostare tale formula sul lato opposto del simbolo di sequente. Se la formula è una premessa, la sua retro (post)-negazione sarà una conclusione, e viceversa; questo scambio di ruoli, che esprime la dualità premesse-conclusioni, rispetta l’ordine mostrato qui sotto:

$$\begin{aligned} \frac{A, \Gamma \vdash \Delta}{\Gamma \vdash A^\perp, \Delta} ()^\perp & \quad \frac{\Gamma \vdash A, \Delta}{\perp A, \Gamma \vdash \Delta} \perp() \\ \frac{\Gamma, B \vdash \Delta}{\Gamma \vdash \Delta, \perp B} \perp() & \quad \frac{\Gamma \vdash \Delta, B}{\Gamma, B^\perp \vdash \Delta} ()^\perp \end{aligned}$$

Dalla definizione delle due negazioni in MNLL segue che la legge logica della doppia negazione vale solo per le negazioni simmetriche: $\perp(A^\perp)$, $(\perp A)^\perp$, mentre le formule negate due volte sullo stesso lato non possono essere ridotte a formule positive:

Leggi della Doppia Negazione

$$\begin{aligned} \perp(A^\perp) &= (\perp A)^\perp = A, \\ \perp\perp(A) &\neq (A)^\perp\perp \neq A. \end{aligned}$$

Il seguente teorema è dimostrabile nel calcolo dei sequenti a una parte di MNLL:

Se il sequente $\vdash \Gamma, A$ è dimostrabile in MNLL, anche i sequenti $\vdash A^\perp\perp, \Gamma$ e $\vdash \Gamma, \perp\perp A$ sono dimostrabili in MNLL.

In base a questo teorema nel sistema MNLL si danno due leggi cicliche della negazione (*cyclic negation laws*) che consentono a una formula negata due volte sullo stesso lato di essere spostata, indietro: $()^\perp\perp$ o in avanti: $\perp\perp()$, in un sequente:

Leggi Cicliche della Negazione

$$\frac{\vdash \Gamma, A}{\vdash A^\perp\perp, \Gamma} ()^\perp\perp \quad \frac{\vdash A, \Gamma}{\vdash \Gamma, \perp\perp A} \perp\perp()$$

Il calcolo dei sequenti ad una parte per MNLL include una regola *Assioma*, due regole *Cut*, due regole per la congiunzione moltiplicativa \otimes , e una regola per la disgiunzione moltiplicativa: \wp .⁷

ASSIOMA

$$\frac{}{\vdash A^\perp, A} \text{id}$$

REGOLE CUT

$$\frac{\vdash \Gamma, A, \Gamma' \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta, \Gamma'} \text{cut1} \quad \frac{\vdash \Gamma, A \vdash \Delta, A^\perp, \Delta'}{\vdash \Delta, \Gamma, \Delta'} \text{cut2}$$

REGOLE \otimes

$$\frac{\vdash \Gamma, A, \Gamma' \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta, \Gamma'} \otimes 1 \quad \frac{\vdash \Gamma, A \vdash \Delta, B, \Delta'}{\vdash \Delta, \Gamma, A \otimes B, \Delta'} \otimes 2$$

REGOLA \wp

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \wp B} \wp$$

⁷Qui sono presentate le regole per le formule A^\perp , le regole corrispondenti per le formule $\perp A$ sono analoghe. Le due regole per *Cut* e \otimes servono per rendere conto del contesto.

Un sequente $\vdash \Gamma$ è dimostrabile in MNLL se e solo se c'è una dimostrazione di $\vdash \Gamma$ in MLL. La nozione di *proof net* svolge un ruolo fondamentale in logica lineare (Girard, 1987): i *proof nets* sono grafi associati a dimostrazioni.⁸

Dal momento che una dimostrazione di un sequente corrisponde ad una deduzione da premesse $\Gamma = A_1, \dots, A_n$ a conclusioni $\Delta = B_1, \dots, B_n$, essa può essere rappresentata mediante un grafo orientato π di occorrenze di formule (i *vertici* di π), in cui ciascun vertice in π è la *premessa* di al massimo un link in π e la *conclusione* di esattamente un link in π . Girard (1987) definisce una classe di grafi con queste proprietà come la classe delle *proof structures* della logica lineare moltiplicativa (MLL) e mostra che ad ogni dimostrazione D di un sequente $\vdash \Gamma$ in MLL può essere associata una *proof structure* (commutativa) $\Phi(D)$ le cui conclusioni sono esattamente le formule in Γ . I *proof nets* per MLL sono quindi individuati come un particolare sottoinsieme delle *proof structures* di MLL, quelle che non presentano *short trips*.⁹

Nella logica lineare non-commutativa MNLL, che risulta della esclusione della regola strutturale dello scambio (*exchange*), i *proof nets* non-commutativi hanno una buona proprietà geometrica: sono grafi planari, in cui i links non si intersecano. I *proof nets* in MNLL sono costruiti per mezzo dei seguenti *links*¹⁰:

LINK ASSIOMA



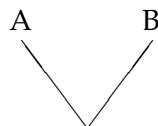
LINK CUT

(due premesse, nessuna conclusione)



LINK \otimes

(due premesse, una conclusione)



$A \otimes B$

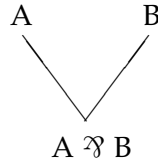
LINK \wp

(due premesse, una conclusione)

⁸*Proof nets* per il Calcolo di Lambek sono analizzati in Abrusci (1991) e Moortgat (1997). Le proprietà dei *proof nets* non-commutativi sono studiate in particolare in Abrusci e Ruet (1999) e Abrusci (2014).

⁹Per la spiegazione di queste nozioni e le relative dimostrazioni si rimanda a Girard (1987).

¹⁰Di nuovo i links sono dati per formule A^\perp ; i corrispondenti links con formule ${}^\perp A$ seguono per dualità e sono analoghi.



Come si vede dalle regole per i connettivi di MNLL, il *link- \otimes* connette due formule che appartengono a contesti separati, mentre il *link- \wp* connette due formule che appartengono allo stesso contesto.

5. Calcolo di Lambek e logica lineare

I tipi sintattici del Calcolo di Lambek L possono essere tradotti in MNLL mediante una procedura che sostituisce ciascuna formula con / o \ in una formula con una appropriata sequenza di \wp e negazioni $()^\perp, {}^\perp()$, iniziando dalle formule semplici $A \setminus B$ e A / B :

$$\begin{aligned}
 A \setminus B &\text{ è tradotta in } A^\perp \wp B, \\
 B / A &\text{ è tradotta in } B \wp^\perp A.
 \end{aligned}$$

Su questa base si possono ottenere tipi più complessi:

Verbo Intransitivo:	IV	$NP \setminus S$	$NP^\perp \wp S$
Verbo Transitivo:	TV	$(NP \setminus S) / NP$	$NP^\perp \wp S \wp^\perp NP$
Verbo Ditransitivo:	DTV	$((NP \setminus S) / PP) / NP$	$NP^\perp \wp S \wp^\perp PP \wp^\perp NP$
Sintagma Quantificazionale:	Q_1	$S / (NP \setminus S)$	$S \wp ({}^\perp S \otimes NP)$
Sintagma Quantificazionale:	Q_2	$(S / NP) \setminus S$	$(NP \otimes S^\perp) \wp S$
Avverbio Intransitivo:	IVA	$(NP \setminus S) \setminus (NP \setminus S)$	$(S^\perp \otimes NP^\perp) \wp NP^\perp \wp S$

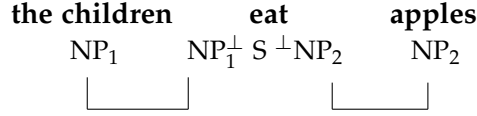
In L, il tipo IV dei verbi intransitivi e il tipo TV dei verbi transitivi sono introdotti nel calcolo dei sequenti a due parti (a sinistra) o a una parte (a destra), mediante le due implicazioni / e \:

$$\begin{array}{lll}
 \text{IV: } run & NP \setminus S \vdash NP \setminus S & run \vdash NP \setminus S \\
 \text{TV: } eats & (NP_1 \setminus S) / NP_2 \vdash (NP_1 \setminus S) / NP_2 & eats \vdash (NP_1 \setminus S) / NP_2
 \end{array}$$

In MNLL questi tipi sono introdotti come sequenti del calcolo ad una parte, in cui la virgola corrisponde al " \wp ":

$$\begin{aligned}
 \text{IV: } &\vdash run^\perp, NP^\perp, S \\
 \text{TV: } &\vdash eat^\perp, NP^\perp, S, {}^\perp NP
 \end{aligned}$$

La definizione dei tipi del Calcolo di Lambek nei termini di \wp e delle due negazioni $()^\perp, {}^\perp()$, ha l'effetto di rappresentare i posti per argomento di una categoria funzionale (come IV o TV) secondo un ordine lineare, che permette di collegarli simultaneamente con gli argomenti appropriati; le modalità di tali connessioni possono essere adeguatamente rappresentate mediante i *proof nets* non-commutativi di MNLL, come mostrato in questo esempio:



Questo grafo planare ha la buona proprietà di rappresentare simultaneamente le due derivazioni, diverse ma equivalenti, in base a cui la stringa corrispondente al verbo transitivo TV: *eats* viene combinata con i suoi due argomenti: il soggetto NP_1 e il complemento oggetto NP_2 :

Figura 1. MNLL: *Derivazioni di TV*

$$\frac{\frac{\frac{\vdash \text{the} - \text{children}^\perp, \text{NP} \quad \vdash \text{eat}^\perp, \text{NP}^\perp, \text{S}, \perp \text{NP}}{\vdash \text{eat}^\perp, \text{the} - \text{children}^\perp, \text{S}, \perp \text{NP}} \text{ cut} \quad \vdash \text{apples}^\perp, \text{NP}}{\vdash \text{apples}^\perp, \text{eat}^\perp, \text{the} - \text{children}^\perp, \text{S}} \text{ cut}}{\text{the} - \text{children}, \text{eat}, \text{apples} \vdash \text{S}}$$

$$\frac{\frac{\frac{\vdash \text{the} - \text{children}^\perp, \text{NP} \quad \frac{\frac{\vdash \text{eat}^\perp, \text{NP}^\perp, \text{S}, \perp \text{NP} \quad \vdash \text{apples}^\perp, \text{NP}}{\vdash \text{apples}^\perp, \text{eat}^\perp, \text{NP}^\perp, \text{S}} \text{ cut}}{\vdash \text{apples}^\perp, \text{eat}^\perp, \text{the} - \text{children}^\perp, \text{S}} \text{ cut}}{\text{the} - \text{children}, \text{eat}, \text{apples} \vdash \text{S}}$$

Le due derivazioni equivalenti sono ottenute ciascuna mediante due *cuts* operati in ordine inverso, prima il verbo transitivo è applicato al soggetto e poi all'oggetto, e viceversa. Poiché in MNLL vale il teorema di *cut-elimination*, come provato in Abrusci (1991), derivazioni *cut-free* possono essere date per ogni espressione linguistica che si ottiene con *cuts* semplici, o *cuts* complessi che possono essere così ridotti. In questo senso il calcolo dei sequenti ad una parte per MNLL rappresenta una procedura efficiente per ottenere dimostrazioni mediante *cuts*.

6. Alcuni esempi linguistici

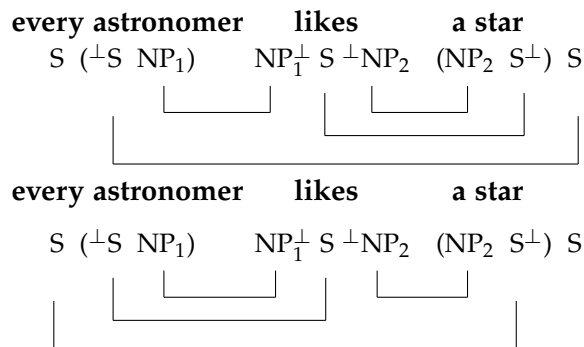
Consideriamo ora alcune applicazioni linguistiche che risultano dalla traduzione del Calcolo di Lambek nel frammento moltiplicativo della logica lineare non-commutativa MNLL. In primo luogo, prendiamo in esame alcuni esempi relativi al campo d'azione (*scope*) dei quantificatori (Montague, 1976; Benthem e Ter Meulen, 1996), mostrando come si possa rappresentarlo mediante *proof net* non-commutativi. Quindi viene proposta l'analisi, sempre avvalendosi della geometria dei *proof net*, di alcuni casi di contesti subordinati, conosciuti nella letteratura come dipendenze illimitate (*unbounded dependencies*) (Van Benthem, 1991; Moortgat, 1997; Morrill, 2010; Morrill, 1994).

6.1. Quantificatori

Poichè i *proof nets* in MNLL hanno la buona proprietà di rappresentare simultaneamente tutte le derivazioni equivalenti di una data stringa di parole, un *proof net* può essere associato all'insieme di tutte le dimostrazioni equivalenti di una data stringa di parole (espressione o frase). Le ambiguità reali corrisponderanno quindi a *proof net* differenti, come nel caso delle ambiguità di campo d'azione esemplificate da enunciati con quantificatori come 'Every astronomer likes a star' ('Ogni astronomo ama una stella'), che ricevono due rappresentazioni diverse corrispondenti al *narrow scope* (campo d'azione ridotto) vs. *wide scope* (campo d'azione ampio) di ciascun quantificatore (vedi Montague (1976)).

Nel seguente esempio, i sintagmi quantificazionali ricevono il tipo D_1 e D_2 per le rispettive occorrenze pre-verbali vs. post-verbali e i due *proof net* sono grafi planari in forma normale, con *links* paralleli:

LESSICO $D_1 : \vdash \text{every astronomer}^\perp, (S \wp (\perp S \otimes NP_1))$
 $D_2 : \vdash \text{a star}^\perp, ((NP_2 \otimes S^\perp) \wp S)$
 TV : $\vdash \text{likes}^\perp, (NP_1^\perp \wp S \wp \perp NP_2)$



La lettura *narrow scope* del quantificatore subordinato D_2 , associata alla lettura *wide scope* del quantificatore principale D_1 : 'every astronomer is such that he likes a star' ('ogni astronomo è tale che egli ama una stella'), è analizzata in MNLL come il processo che inizia nel contesto del predicato verbale e termina nel quantificatore principale, in cui occorre il tipo S non connesso da alcun link ad altri tipi.

D'altro lato, la lettura *wide scope* del quantificatore subordinato D_2 , associata alla lettura *narrow scope* del quantificatore principale D_1 , *there is a star such that every astronomer likes it* (c'è una stella tale che ogni astronomo la ama), è analizzata in MNLL come il processo che va dal soggetto principale, il quantificatore D_1 , al complemento verbale con il quantificatore D_2 , in cui occorre il tipo S non connesso ad altri tipi.

Ai due *proof net* corrispondono le dimostrazioni in Figura 2, nel calcolo dei sequenti a una parte per MNLL. Vediamo dunque che i *proof net* della logica non-commutativa rappresentano un metodo efficiente per inferire le proprietà di campo d'azione dei quantificatori, a partire dall'informazione fornita dalle assegnazioni di tipi sintattici agli elementi del lessico e dalle operazioni associate ai connettivi di MNLL (Casadio e Lambek, 2002).

LESSICO	NP :	$\vdash Elisa^\perp, NP_1$
	DTV :	$\vdash write^\perp, (NP_1^\perp \wp S \wp \perp PP \wp \perp NP_2)$
	AUX :	$\vdash did^\perp, (Q \wp \perp INF \wp \perp NP_1)$
	WH :	$\vdash whom^\perp, (\bar{Q} \wp (\perp\perp NP_i \otimes \perp Q))$
	P :	$\vdash to^\perp, (PP \wp \perp NP_3)$

Su questa base, i verbi transitivi DTV, come *give* o *write*, che prendono un oggetto diretto DO e un oggetto indiretto IO, hanno la seguente struttura argomentale:

SUBJ	DTV	DO	IO
NP_1	$(NP_1^\perp S \perp PP \perp NP_2)$	NP_2	PP
└───┘		└───┘	
	└──────────┘		

Essa è esemplificata dalla seguente dimostrazione nel calcolo dei sequenti ad una parte:

Figura 3. MNLL: *Dipendenza frasale*

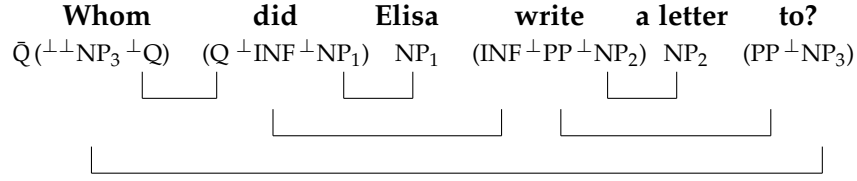
$$\frac{\frac{\frac{\frac{\vdash Elisa^\perp, NP_1 \quad \vdash wrote^\perp, NP_1^\perp, S, \perp PP, \perp NP_2}{\vdash wrote^\perp, Elisa^\perp, S, \perp PP, \perp NP_2} \text{ cut} \quad \vdash a \text{ letter}^\perp, NP_2}{\vdash a \text{ letter}^\perp, wrote^\perp, Elisa^\perp, S, \perp PP} \text{ cut} \quad \vdash to \text{ John}^\perp, PP}{\vdash to \text{ John}^\perp, a \text{ letter}^\perp, wrote^\perp, Elisa^\perp, S} \text{ cut}}{\text{Elisa, wrote, a letter, to John} \vdash S} (i)^\perp$$

La dimostrazione e tutte quelle ad essa equivalenti corrispondono a questo semplice proof net:

Elisa	wrote	a letter	to John
NP_1	$(NP_1^\perp S \perp PP \perp NP_2)$	NP_2	PP
└───┘		└───┘	
	└──────────┘		

La domanda relativa all'oggetto diretto DO si ottiene assegnando al pronome-*wh* il tipo: $(\bar{Q} \wp (\perp\perp NP_i \otimes \perp Q))$; in base a questo assegnamento, la formula $(\perp\perp NP_3 \otimes \perp Q)$ è duale della formula $(Q \wp \perp NP_3)$ che risulta dalla composizione dei costituenti della clausola con cui il sintagma-*wh* è in relazione, per NP_3 argomento della preposizione *to*. La relazione di dipendenza si ottiene eseguendo il *cut* tra il \wp -link nella formula $(Q \wp \perp NP_3)$ della clausola principale e il \otimes -link nella formula $(\perp\perp NP_3 \otimes \perp Q)$ del sintagma-*wh*. Nel *proof net* questo *cut* risulta nel lungo link che connette il tipo a sinistra $\perp\perp NP_3$ al tipo $\perp NP_3$ nella parte finale della dipendenza frasale.¹²

¹²La formula con due negazioni nel tipo del sintagma-*wh* è collegata alla posizione definita da Chomsky come 'traccia' del movimento che l'argomento ha compiuto per raggiungere la parte iniziale della frase interrogativa, assumendo la forma di sintagma quantificazionale; vedi Chomsky (1995).



Per dimostrare il sequente $Whom, did, Elisa, write, a\text{-}letter, to \vdash \bar{Q}$, deriviamo il sequente che include l'argomento $\perp NP_2$ a cui il sintagma nominale *a-letter* si applica:

$$\frac{\frac{\vdash did^\perp, Q, \perp INF, \perp NP_1 \quad \vdash Elisa^\perp, NP_1}{\vdash Elisa^\perp, did^\perp, Q, \perp INF} \text{ cut} \quad \vdash write^\perp, INF, \perp PP, \perp NP_2}{\vdash write^\perp, Elisa^\perp, did^\perp, Q, \perp PP, \perp NP_2} \text{ cut}$$

Poi applichiamo gli argomenti NP_2 e PP :

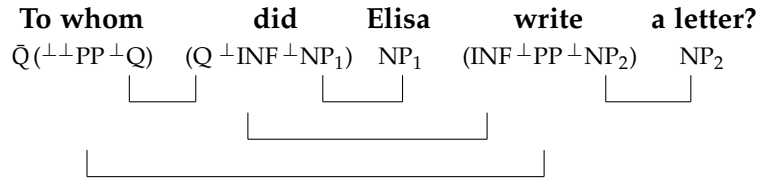
$$\frac{\frac{\vdash write^\perp, Elisa^\perp, did^\perp, Q, \perp PP, \perp NP_2 \quad \vdash a\text{-}letter^\perp, NP_2}{\vdash a\text{-}letter^\perp, write^\perp, Elisa^\perp, did^\perp, Q, \perp PP} \text{ cut} \quad \vdash to^\perp, PP, \perp NP_3}{\vdash to^\perp, a\text{-}letter^\perp, write^\perp, Elisa^\perp, did^\perp, Q, \perp NP_3} \text{ cut}$$

Infine applichiamo il pronome relativo *Whom* alla formula risultante:

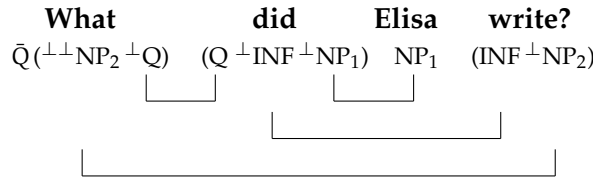
$$\frac{\vdash Whom^\perp, \bar{Q}, (\perp\perp NP_3 \otimes \perp Q) \quad \vdash to^\perp, a\text{-}letter^\perp, write^\perp, Elisa^\perp, did^\perp, (Q \wp \perp NP_3)}{\vdash to^\perp, a\text{-}letter^\perp, write^\perp, Elisa^\perp, did^\perp, Whom^\perp, \bar{Q}} \text{ cut} \quad (i)^\perp$$

$Whom, did, Elisa, write, a\text{-}letter, to \vdash \bar{Q}$

Il caso in cui la preposizione *to* occorre all'inizio della frase insieme al pronome-*wh* segue lo stesso schema: la differenza è che il tipo $(\bar{Q} \wp (\perp\perp PP \otimes \perp Q))$ è assegnato al composto lessicale *to-whom* e il \wp -link collega il posto per argomento di tipo $\perp PP$ dell'oggetto indiretto nel DTV e il tipo $\perp\perp PP$ nel sintagma-*wh*:



La forma interrogativa dell'oggetto diretto DO si ottiene assegnando il tipo $(\bar{Q} \wp (\perp\perp NP_2 \otimes \perp Q))$ al pronome-*wh*, in cui la formula $(\perp\perp NP_2 \otimes \perp Q)$ risulta duale della formula $(Q \wp \perp NP_2)$, che corrisponde alle clausola in cui manca l'oggetto diretto cui si riferisce il pronome relativo:



7. Conclusioni

In questo lavoro abbiamo cercato di mettere in luce, con l'aiuto di alcuni esempi, il ruolo innovativo che può essere assolto dalla logica lineare non commutativa nello studio delle proprietà formali delle lingue naturali. In particolare, abbiamo cercato di mostrare che le grammatiche dei tipi logici possono essere riformulate e considerate sotto una nuova luce nei termini delle rappresentazioni geometriche offerte dalle reti di dimostrazione (*proof nets*) della logica lineare non commutativa. Queste strutture, come risultato dell'interazione delle due negazioni e dei connettivi duali \wp , \otimes , costituiscono un contesto dinamico per la rappresentazione del linguaggio naturale, nel quale una sequenza di formule (un enunciato) può essere considerato da diversi punti di vista.

In un articolo del 1999 Lambek (1999) ha presentato la formulazione di un nuovo calcolo per l'analisi logica delle lingue naturali, da lui definito calcolo dei pregruppi o *Pregroup Grammar*. Non entreremo qui nei dettagli di questo calcolo che rappresenta un'alternativa innovativa del Syntactic Calculus del 1958; per gli aspetti teorici e le applicazioni linguistiche si vedano in particolare: Lambek (2008), Casadio e Lambek (2001), Casadio e Lambek (2002) e Casadio e Lambek (2008). Come il calcolo sintattico era basato sulla operazione di *residuation* e su monoidi residuati, il calcolo dei pregruppi si basa sulle nozioni di *compact closed category* e aggiunto, riformulando in questo contesto le proprietà della logica lineare non-commutativa moltiplicativa (Casadio e Lambek, 2002; Lambek, 2004; Lambek, 2001; Lambek, 2008).

Un aspetto caratteristico del calcolo dei pregruppi è la definizione di categoria che ammette due aggiunti, un aggiunto di destra (*right adjoint*) e un aggiunto di sinistra (*left adjoint*) in modo analogo alle due negazioni della logica lineare non-commutativa, precisamente:

$$\begin{array}{ll} \text{post-negazione di } A = A^\perp & \text{vs. aggiunto di destra di } A = A^\top \\ \text{retro-negazione di } A = {}^\perp A & \text{vs. aggiunto di sinistra di } A = A^\lrcorner \end{array}$$

Per via della semplificazione ottenuta nei processi di computazione, il calcolo dei pregruppi è stato applicato in breve tempo a numerose lingue naturali: inglese, italiano, francese, tedesco, polacco, persiano, giapponese e molte altre (Casadio e Lambek, 2008; Lambek, 2010; Sadrzadeh, 2008). Le proprietà del calcolo sono attualmente oggetto di studio, ma è importante sottolineare la caratteristica fondamentale di questi sistemi: l'attenzione verso l'ordine delle risorse che vengono prese in considerazione. Tale sensibilità all'ordine, che risulta dalla soppressione della regola strutturale dello scambio, si riflette nella presenza di due negazioni (due aggiunti) e nella conseguente architettura simmetrica e planare delle reti di dimostrazione (*proof net*).

Bibliografia

Abrusci, V. M. (1991). «Phase semantics and sequent calculus for pure noncommutative classical linear propositional logic». In: *The Journal of Symbolic Logic* 56.4, pp. 1403–1451.

- Abrusci, V. M. (2002). «Classical conservative extensions of Lambek calculus». In: *Studia Logica* 71.3, pp. 277–314.
- Abrusci, V. M. (2014). «On Residuation». In: *Categories and Types in Logic, Language, and Physics*. A cura di C. Casadio, B. Coecke, M. Moortgat e P. Scott. Springer, pp. 14–27.
- Abrusci, V. M. e P. Ruet (1999). «Non-commutative logic I: the multiplicative fragment». In: *Annals of pure and applied logic* 101.1, pp. 29–64.
- Ades, A. E. e M. J. Steedman (1982). «On the order of words». In: *Linguistics and philosophy* 4.4, pp. 517–558.
- Ajdukiewicz, K. (1935). «Die syntaktische Konnexität». In: *Studia Philosophica* 1, pp. 1–27.
- Bar-Hillel, Y. (1953). «A quasi-arithmetical notation for syntactic description». In: *Language* 29, pp. 47–58.
- Benthem, J. van e A. Ter Meulen, cur. (1996). *Handbook of logic and language*. Elsevier.
- Casadio, C. e J. Lambek, cur. (2008). *Recent Computational Algebraic Approaches to Morphology and Syntax*. Polimetrica.
- Casadio, C. (2011). «Espressione e significato nelle Ricerche Logiche di Husserl». In: *Logica, Ontologia ed Etica*. Franco Angeli.
- Casadio, C. e J. Lambek (2001). «An algebraic analysis of clitic pronouns in Italian». In: *Logical Aspects of Computational Linguistics*. A cura di P. De Groote, G. Morrill e C. Retoré. Springer, pp. 110–124.
- Casadio, C. e J. Lambek (2002). «A tale of four grammars». In: *Studia Logica* 71.3, pp. 315–329.
- Chomsky, N. (1995). *The minimalist program*. Vol. 1765. Cambridge Univ Press.
- Frege, G. (1892). *Über Sinn und Bedeutung*, *Zeitschrift für Philosophie und philosophische Kritik*.
- Frege, G. (1893/1903). *Die Grundgesetze der Arithmetik*. Vol. 1. Pohle.
- Frege, G. (1976). «Frege an Husserl, Husserl an Frege». In: *Wissenschaftlicher Briefwechsel*. A cura di G. Gabriel. Meiner.
- Geach, P. T. (1972). «A program for syntax». In: *Semantics of Natural Language*. A cura di D. Davidson e G. Harman. Reidel P.C., pp. 483–497.
- Girard, J.-Y. (1987). «Linear logic». In: *Theoretical computer science*. Vol. 50, pp. 1–102.
- Girard, J.-Y. (1995). «Linear logic: its syntax and semantics». In: a cura di A. in *Linear Logic*. Cambridge University Press, pp. 1–42.
- Hillel, Y. B., C. Gaifman e E. Shamir (1960). «On categorial and phrase structure grammars». In: *Bulletin of the research council of Israel* 9.
- Husserl, E. (1913). *Logische Untersuchungen*. Halle.
- Lambek, J. (1958). «The mathematics of sentence structure». In: *American mathematical monthly* 65, pp. 154–170.
- Lambek, J. (1961). «On the calculus of syntactic types». In: *Structure of language and its mathematical aspects*. A cura di R. Jacobson. Providence.
- Lambek, J. (1999). «Type grammar revisited». In: *Logical aspects of computational linguistics*, pp. 1–27.
- Lambek, J. (2001). «Type grammars as pregroups». In: *Grammars* 4.1, pp. 21–39.

- Lambek, J. (2004). «A computational algebraic approach to English grammar». In: *Syntax* 7.2, pp. 128–147.
- Lambek, J. (2008). *From Word to Sentence: a computational algebraic approach to grammar*. Polimetrica.
- Lambek, J. (2010). «Exploring feature agreement in French with parallel pregroup computations». In: *Journal of Logic, Language and Information* 19.1, pp. 75–88.
- McCall, S. e K. Ajdukiewicz, cur. (1967). *Polish Logic, 1920-1939*. Oxford, Clarendon Press.
- Montague, R. (1976). *Formal Philosophy: Selected Papers*. Yale University Press.
- Moortgat, M. (1997). «Categorial Type Logics». In: a cura di J. van Benthem e A. ter Meulen, pp. 93–177.
- Morrill, G. (1994). «Type Logical Grammar Categorial Logic of Signs». In: Morrill, G. (2010). *Categorial grammar: Logical syntax, semantics, and processing*. Oxford: Oxford University Press.
- Sadrzadeh, M. (2008). «Pregroup analysis of Persian sentences». In: *Recent Computational Algebraic Approaches to Morphology and Syntax*. A cura di C. Casadio e J. Lambek. Polimetrica.
- Van Benthem, J. (1988). «The lambek calculus». In: *Categorial grammars and natural language structures*. A cura di Oehrle. Springer, pp. 35–68.
- Van Benthem, J. (1991). «Language in action». In: *Journal of philosophical logic* 20.3, pp. 225–263.