Syntax and Semantics in Generalized Lambek Calculus

Michael Moortgat

LIRa seminar, January 24, 2011, Amsterdam

Abstract

Lambek's Syntactic Calculus (1961) is a logic completely without structural rules: rules affecting multiplicity (contraction, weakening) or structure (commutativity, associativity) of the grammatical resources are not considered. Originally conceived with linguistics in mind, Lambek's calculus (both the 61 and the associative 58 variant or its modern pregroup incarnation) have found many models outside linguistics: as the logic for composition of informational actions, for example, and in fields such as mathematical morphology or quantum physics.

In terms of expressivity, Lambek's calculi are strictly context-free. The context-free limitation makes itself felt in situations where syntactic and semantic composition seem to be out of sync: long distance dependencies in syntax, or the dynamics of scoping in semantics. In the talk, I discuss the Lambek-Grishin calculus, a symmetric generalization of the syntactic calculus allowing multiple conclusions. I show how its symmetry principles resolve the tension at the syntax-semantics interface.

Background reading: Symmetric categorial grammar. JPL, 38 (6) 681-710.

1. Motivation

Lambek's syntactic calculus — **(N)L**, pregroup grammar — is strictly context-free.

Expressive limitations Problematic are discontinuous dependencies: information flow between detached parts of an utterance

- extraction. Who _ stole the tarts? vs What did Alice find _ there?
- infixation. Alice thinks someone is cheating local vs non-local interpretation.

1. Motivation

Lambek's syntactic calculus — **(N)L**, pregroup grammar — is strictly context-free.

Expressive limitations Problematic are discontinuous dependencies: information flow between detached parts of an utterance

- extraction. Who _ stole the tarts? vs What did Alice find _ there?
- infixation. Alice thinks someone is cheating local vs non-local interpretation.

Stragegies for reconciling form/meaning

- **NL**$\Diamond$: controlled structural options, embedding translations; $\sim$ linear logic !,?
- Lambek-Grishin calculus **LG**, after Grishin 1983
  - symmetry: residuated, Galois connected operations and their duals
  - structural rules $\leadsto$ logical distributivity principles
  - continuation semantics: relieves the burden on syntactic source calculus

2. Lambek-Grishin calculus: fusion vs fission

Lambek-Grishin calculus **NL** has $\otimes$, left and right division $\backslash, /$ forming a residuated triple. **LG** adds a dual residuated triple: coproduct $\oplus$, right and left difference $\oslash, \obslash$.

$$A \to C/B \quad \Leftrightarrow \quad A \otimes B \to C \quad \Leftrightarrow \quad B \to A\backslash C$$
$$B \obslash C \to A \quad \Leftrightarrow \quad C \to B \oplus A \quad \Leftrightarrow \quad C \oslash A \to B$$

2. Lambek-Grishin calculus: fusion vs fission

Lambek-Grishin calculus **NL** has $\otimes$, left and right division $\backslash, /$ forming a residuated triple. **LG** adds a dual residuated triple: coproduct $\oplus$, right and left difference $\oslash, \obslash$.

$$A \to C/B \quad \Leftrightarrow \quad A \otimes B \to C \quad \Leftrightarrow \quad B \to A \backslash C$$
$$B \obslash C \to A \quad \Leftrightarrow \quad C \to B \oplus A \quad \Leftrightarrow \quad C \oslash A \to B$$

Interpretation Algebraic (Ono, Buszkowski); Kripke-style relational (Dunn, Kurtonina). For the latter: frames $(W, R_\otimes, R_\oplus)$, with operations defined on subsets of W.

$x \Vdash$	$A \otimes B$	iff	$\exists yz.R_\otimes xyz$ and $y \Vdash A$ and $z \Vdash B$
$y \Vdash$	C/B	iff	$\forall xz.(R_\otimes xyz$ and $z \Vdash B)$ implies $x \Vdash C$
$z \Vdash$	$A \backslash C$	iff	$\forall xy.(R_\otimes xyz$ and $y \Vdash A)$ implies $x \Vdash C$

$x \Vdash$	$A \oplus B$	iff	$\forall yz.R_\oplus xyz$ implies $(y \Vdash A$ or $z \Vdash B)$
$y \Vdash$	$C \oslash B$	iff	$\exists xz.R_\oplus xyz$ and $z \nVdash B$ and $x \Vdash C$
$z \Vdash$	$A \obslash C$	iff	$\exists xy.R_\oplus xyz$ and $y \nVdash A$ and $x \Vdash C$

Note As yet no assumptions about relation between fusion $R_\otimes$, fission $R_\oplus$.

3. Through the Looking Glass

Two symmetries To the left-right symmetry $\cdot^{\bowtie}$ of **NL**, **LG** adds an arrow reversal symmetry $\cdot^{\infty}$. Together with identity and composition: Klein group.

$$A^{\bowtie} \xrightarrow{f^{\bowtie}} B^{\bowtie} \quad \Leftrightarrow \quad A \xrightarrow{f} B \quad \Leftrightarrow \quad B^{\infty} \xrightarrow{f^{\infty}} A^{\infty}$$

Translation tables

$$\bowtie \quad \frac{C/D \quad A \otimes B \quad B \oplus A \quad D \obslash C}{D \backslash C \quad B \otimes A \quad A \oplus B \quad C \oslash D} \qquad\qquad \infty \quad \frac{C/B \quad A \otimes B \quad A \backslash C}{B \obslash C \quad B \oplus A \quad C \oslash A}$$

3. Through the Looking Glass

Two symmetries To the left-right symmetry $\cdot^{\bowtie}$ of **NL**, **LG** adds an arrow reversal symmetry $\cdot^{\infty}$. Together with identity and composition: Klein group.

$$A^{\bowtie} \xrightarrow{f^{\bowtie}} B^{\bowtie} \quad \Leftrightarrow \quad A \xrightarrow{f} B \quad \Leftrightarrow \quad B^{\infty} \xrightarrow{f^{\infty}} A^{\infty}$$

Translation tables

$$\bowtie \quad \begin{array}{cccc} C/D & A \otimes B & B \oplus A & D \obslash C \\ \hline\hline D\backslash C & B \otimes A & A \oplus B & C \oslash D \end{array} \qquad\qquad \infty \quad \begin{array}{ccc} C/B & A \otimes B & A\backslash C \\ \hline\hline B \obslash C & B \oplus A & C \oslash A \end{array}$$

$\rightsquigarrow$ theorems form quartets — below the (co)unit laws:

$$B \obslash (B \oplus A) \to A \to B \oplus (B \obslash A) \;\overset{\infty}{\longleftrightarrow}\; (A \oplus B) \oslash B \to A \to (A \oslash B) \oplus B$$

$$\updownarrow\ \bowtie \qquad\qquad\qquad\qquad \updownarrow\ \infty$$

$$(A/B) \otimes B \to A \to (A \otimes B)/B \;\underset{\bowtie}{\longleftrightarrow}\; B \otimes (B\backslash A) \to A \to B\backslash(B \otimes A)$$

4. Distributivity

Interaction fusion, fission Two groups of structure-preserving, linear distributivities.

Option A Recipe: select a $\otimes/\oplus$ factor in the premise; simultaneously introduce the residual operations for the remaining two in the conclusion. Note: $\cdot^{\bowtie}$ symmetry.

$$\frac{A \otimes B \rightarrow C \oplus D}{C \obslash A \rightarrow D \,/\, B} \qquad \frac{A \otimes B \rightarrow C \oplus D}{B \oslash D \rightarrow A \,\backslash\, C}$$

$$\frac{A \otimes B \rightarrow C \oplus D}{C \obslash B \rightarrow A \,\backslash\, D} \qquad \frac{A \otimes B \rightarrow C \oplus D}{A \oslash D \rightarrow C \,/\, B}$$

4. Distributivity

Interaction fusion, fission Two groups of structure-preserving, linear distributivities.

Option A Recipe: select a $\otimes/\oplus$ factor in the premise; simultaneously introduce the residual operations for the remaining two in the conclusion. Note: $\cdot^{\bowtie}$ symmetry.

$$\frac{A \otimes B \rightarrow C \oplus D}{C \oslash A \rightarrow D \,/\, B} \qquad \frac{A \otimes B \rightarrow C \oplus D}{B \oslash D \rightarrow A \,\backslash\, C}$$

$$\frac{A \otimes B \rightarrow C \oplus D}{C \oslash B \rightarrow A \,\backslash\, D} \qquad \frac{A \otimes B \rightarrow C \oplus D}{A \oslash D \rightarrow C \,/\, B}$$

Option B Converses of A. Characteristic theorems: $(A \oplus B) \otimes C \rightarrow A \oplus (B \otimes C)$ etc

Entropy The distributivity rules are non-invertible entropy principles. For the combination of Option A and B, structure-preservation in fact is lost.

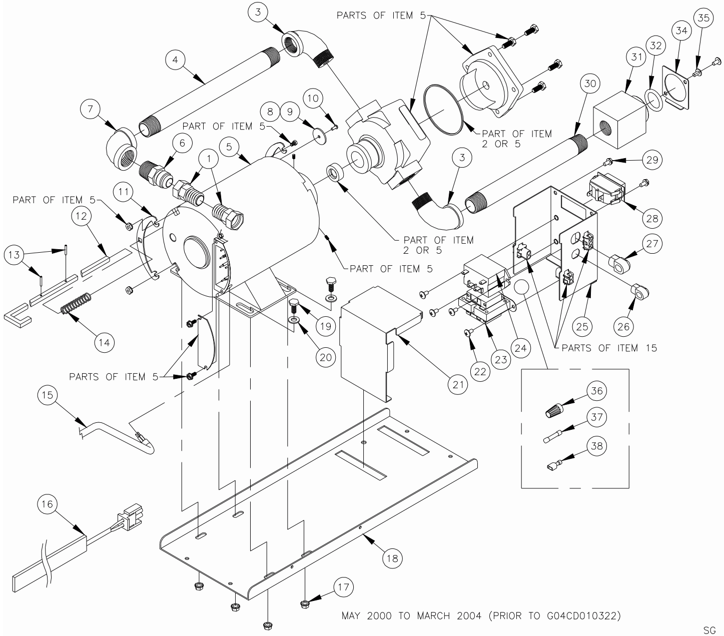
5. The dynamics of information flow

As a deductive system, the arrow calculus is quite unwieldy.

Within the proofs-as-computations tradition, we have two presentations that better capture the information flow in the composition of utterances.

- display sequent calculus
  - MM 2007; with focusing Bastenhof 2010
  - flow: continuation-passing-style
- graphical calculus: nets
  - Moot 2007, after Moot and Puite 2002
  - net assembly: 'exploded parts' diagram

Below, we'll use nets to illustrate how **LG** captures syntactic dependencies beyong CF, and display derivations for continuation-passing in meaning assembly.



MAY 2000 TO MARCH 2004 (PRIOR TO G04CD010322)

SG

6. Graphical calculus: LG proof nets

- Basic building blocks: links.
  - type: tensor, cotensor
  - premises $P_1, \ldots, P_n$, conclusions $C_1, \ldots, C_m$, $0 \leq n, m$
  - Main formula: empty or one of the P_i, C_j

6. Graphical calculus: LG proof nets

- Basic building blocks: links.
  - type: tensor, cotensor
  - premises $P_1, \ldots, P_n$, conclusions $C_1, \ldots, C_m$, $0 \leq n, m$
  - Main formula: empty or one of the P_i, C_j

- Proof structure. Set of links over finite set of frm's s.t. every frm is at most once premise and at most once conclusion of a link.
  - hypotheses: $\neg$ conclusion of any link
  - conclusions: $\neg$ premise of any link
  - axioms: $\neg$ main formula of any link

6. Graphical calculus: LG proof nets

- Basic building blocks: links.
  - type: tensor, cotensor
  - premises $P_1, \ldots, P_n$, conclusions $C_1, \ldots, C_m$, $0 \leq n, m$
  - Main formula: empty or one of the P_i, C_j

- Proof structure. Set of links over finite set of frm's s.t. every frm is at most once premise and at most once conclusion of a link.
  - hypotheses: $\neg$ conclusion of any link
  - conclusions: $\neg$ premise of any link
  - axioms: $\neg$ main formula of any link

- Abstract proof structure: PS with formulas at internal nodes erased.

6. Graphical calculus: LG proof nets

- Basic building blocks: links.
  - type: tensor, cotensor
  - premises $P_1, \ldots, P_n$, conclusions $C_1, \ldots, C_m$, $0 \leq n, m$
  - Main formula: empty or one of the P_i, C_j

- Proof structure. Set of links over finite set of frm's s.t. every frm is at most once premise and at most once conclusion of a link.
  - hypotheses: $\neg$ conclusion of any link
  - conclusions: $\neg$ premise of any link
  - axioms: $\neg$ main formula of any link

- Abstract proof structure: PS with formulas at internal nodes erased.

- Rewriting: logical and structural conversions $\leadsto$ next slides

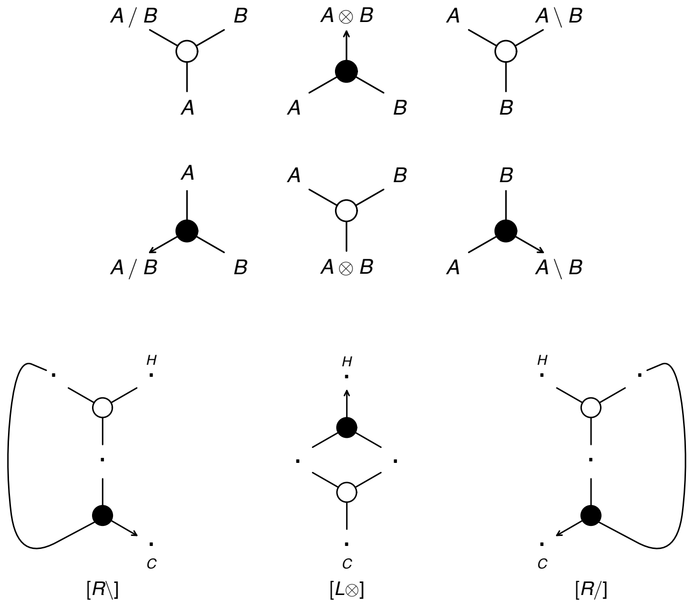
6. Graphical calculus: LG proof nets

- Basic building blocks: links.
  - type: tensor, cotensor
  - premises $P_1, \ldots, P_n$, conclusions $C_1, \ldots, C_m$, $0 \leq n, m$
  - Main formula: empty or one of the P_i, C_j

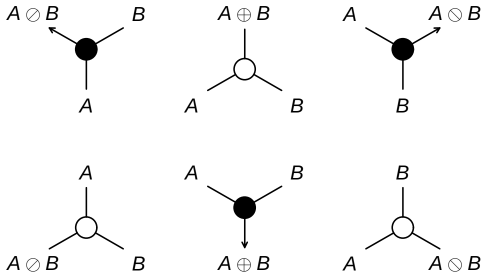
- Proof structure. Set of links over finite set of frm's s.t. every frm is at most once premise and at most once conclusion of a link.
  - hypotheses: $\neg$ conclusion of any link
  - conclusions: $\neg$ premise of any link
  - axioms: $\neg$ main formula of any link

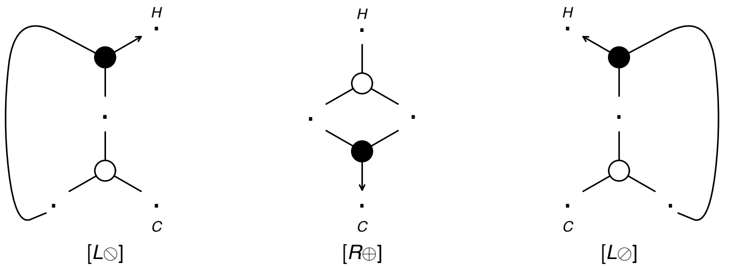
- Abstract proof structure: PS with formulas at internal nodes erased.
- Rewriting: logical and structural conversions $\rightsquigarrow$ next slides
- Proof net: APS converting to a tensor tree (possibly unrooted)

7. Binary links, contractions: tensor



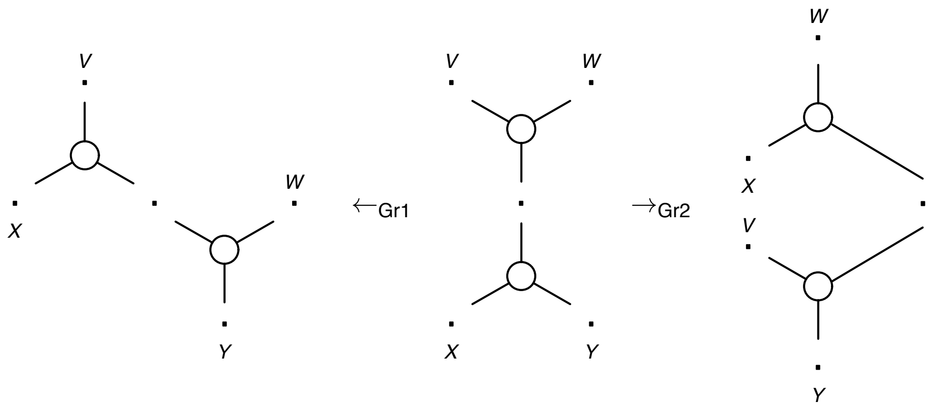
8. Binary links, contractions: tensor$^{\infty}$





9. Structural rewriting

Example Two of Grishin's distributivity laws.



$$X \cdot \obslash \cdot V \to Y \cdot / \cdot W \quad \stackrel{Gr1}{\Leftarrow} \quad V \cdot \otimes \cdot W \to X \cdot \oplus \cdot Y \quad \stackrel{Gr2}{\Rightarrow} \quad X \cdot \obslash \cdot W \to V \cdot \backslash \cdot Y$$

10. Beyond context-free

The original Lambek calculus **(N)L** is strictly context-free, whereas natural languages exhibit patterns beyond CF. Below some examples from formal language theory.

- squares: $\{w^2 \mid w \in \{a, b\}^+\}$
- counting dependencies: $\{a^n b^n c^n \mid n > 0\}$
- crossed dependencies: $\{a^n b^m c^n d^m \mid n, m > 0\}$

10. Beyond context-free

The original Lambek calculus **(N)L** is strictly context-free, whereas natural languages exhibit patterns beyond CF. Below some examples from formal language theory.

- squares: $\{w^2 \mid w \in \{a,b\}^+\}$
- counting dependencies: $\{a^n b^n c^n \mid n > 0\}$
- crossed dependencies: $\{a^n b^m c^n d^m \mid n, m > 0\}$

Mildly context-sensitive formalisms The above patterns are recognized by a family of grammar formalisms, the so-called 'mildly context-sensitive' family. MCS formalisms include the following. They recognize the same languages.

- (L)TAG: (Lexicalized) Tree Adjoining Grammars (Joshi)
- LIG: Linear Indexed Grammars (Gazdar)
- CCG: Combinatory Categorial Grammars (Steedman)

10. Beyond context-free

The original Lambek calculus **(N)L** is strictly context-free, whereas natural languages exhibit patterns beyond CF. Below some examples from formal language theory.

- squares: $\{w^2 \mid w \in \{a,b\}^+\}$
- counting dependencies: $\{a^n b^n c^n \mid n > 0\}$
- crossed dependencies: $\{a^n b^m c^n d^m \mid n, m > 0\}$

Mildly context-sensitive formalisms The above patterns are recognized by a family of grammar formalisms, the so-called 'mildly context-sensitive' family. MCS formalisms include the following. They recognize the same languages.

- (L)TAG: (Lexicalized) Tree Adjoining Grammars (Joshi)
- LIG: Linear Indexed Grammars (Gazdar)
- CCG: Combinatory Categorial Grammars (Steedman)

Moot 2007 shows that LTAG can be straightforwardly translated in **LG**.

11. (L)TAG

(L)TAG is a rewrite system for trees (rather than strings). Σ (vocabulary) and N (non-terminals) as in CFG.

11. (L)TAG

(L)TAG is a rewrite system for trees (rather than strings). Σ (vocabulary) and N (non-terminals) as in CFG.

Elementary trees These are either:

- initial trees: internal nodes $\in N$, leafs from $(\Sigma \cup N)$;
- auxiliary trees: internal nodes $\in N$, leafs from $(\Sigma \cup N)$ one of which (the foot node, marked $*$) labeled with the same non-terminal as the root of the aux tree

11. (L)TAG

(L)TAG is a rewrite system for trees (rather than strings). Σ (vocabulary) and N (non-terminals) as in CFG.

Elementary trees These are either:

- initial trees: internal nodes $\in N$, leafs from $(\Sigma \cup N)$;
- auxiliary trees: internal nodes $\in N$, leafs from $(\Sigma \cup N)$ one of which (the foot node, marked $*$) labeled with the same non-terminal as the root of the aux tree

In an LTAG (Lexicalized TAG), every elementary tree has at least one element from Σ in its yield: the lexical anchor.

11. (L)TAG

(L)TAG is a rewrite system for trees (rather than strings). Σ (vocabulary) and N (non-terminals) as in CFG.

Elementary trees These are either:

- initial trees: internal nodes $\in N$, leafs from $(\Sigma \cup N)$;
- auxiliary trees: internal nodes $\in N$, leafs from $(\Sigma \cup N)$ one of which (the foot node, marked $*$) labeled with the same non-terminal as the root of the aux tree

In an LTAG (Lexicalized TAG), every elementary tree has at least one element from Σ in its yield: the lexical anchor.

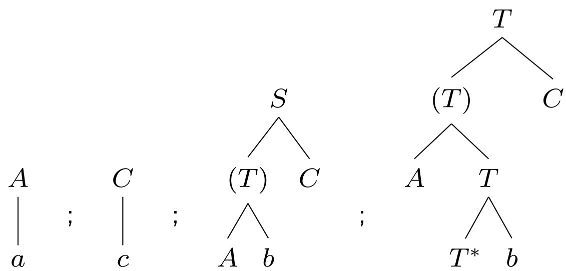
Operations Elementary trees are combined by two operations:

- substitution: replace a leaf ($\neq \alpha^*$) by an initial tree with the same label
- adjunction: expand an internal node α with an auxiliary tree with root/foot labeled α

12. Counting dependencies: LTAG and LG

$$\{a^n b^n c^n \mid n > 0\}$$

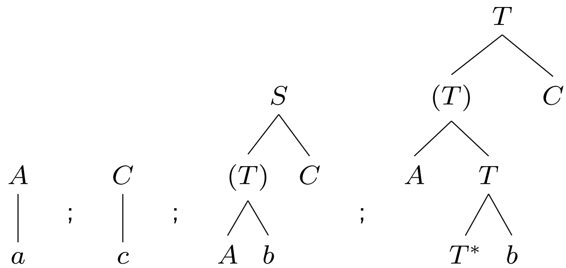
LTAG Auxiliary tree on the right; adjunction node (T).



12. Counting dependencies: LTAG and LG

$$\{a^n b^n c^n \mid n > 0\}$$

LTAG Auxiliary tree on the right; adjunction node (T).

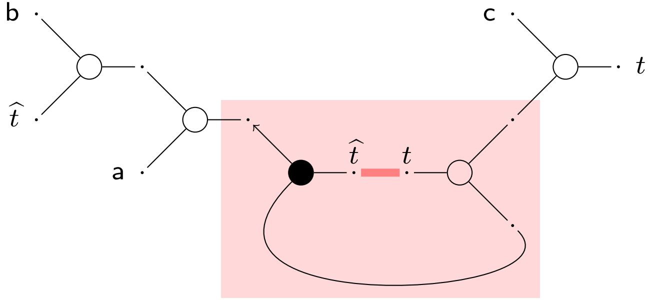


LG Type assignments with $\widehat{T}$ such that $\widehat{T} \rightarrow T$ but not v.v. $a :: A$, $c :: C$ and

$$b :: A \backslash ((T \oslash (S/C)) \obslash \widehat{T}) \quad ; \quad b :: \widehat{T} \backslash (A \backslash ((T \oslash (T/C)) \obslash \widehat{T}))$$

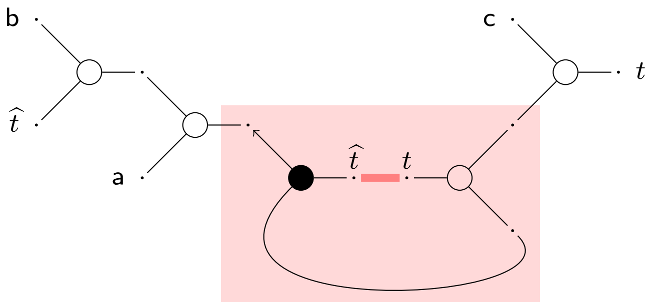
13. Deriving $aabbcc$: the auxiliary formula

Step 1 For $n > 1$, we use $n-1$ times the auxiliary formula b :: $\widehat{t}\backslash(a\backslash((t \oslash (t/c)) \obslash \widehat{t}))$.
The nth use (no further adjunction) is internally connected, and contracts.

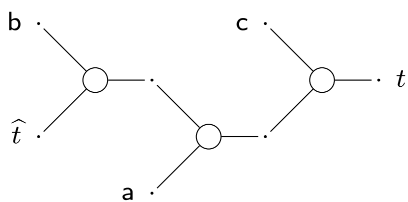


13. Deriving $aabbcc$: the auxiliary formula

Step 1 For $n > 1$, we use $n-1$ times the auxiliary formula b :: $\widehat{t}\backslash(a\backslash((t \oslash (t/c)) \otimes \widehat{t}))$.
The nth use (no further adjunction) is internally connected, and contracts.

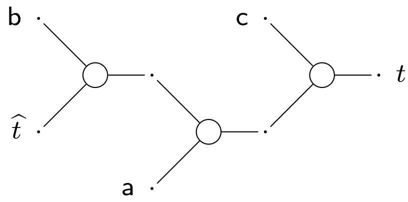


After contraction:



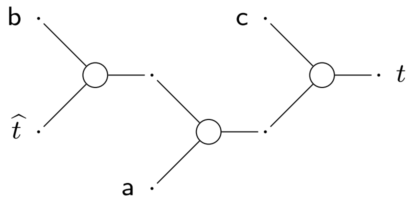
14. Deriving $aabbcc$: adjunction

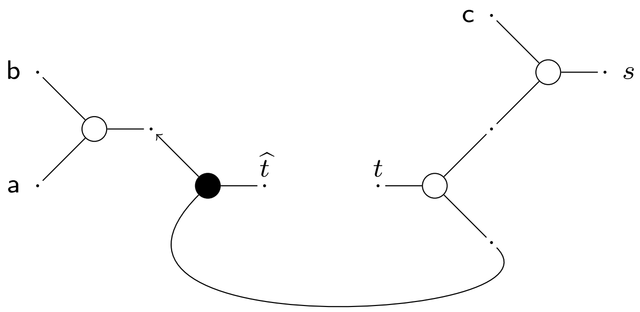
Step 2 To obtain $aabbcc$, take the contracted auxiliary graph of the previous slide



14. Deriving $aabbcc$: adjunction

Step 2 To obtain $aabbcc$, take the contracted auxiliary graph of the previous slide

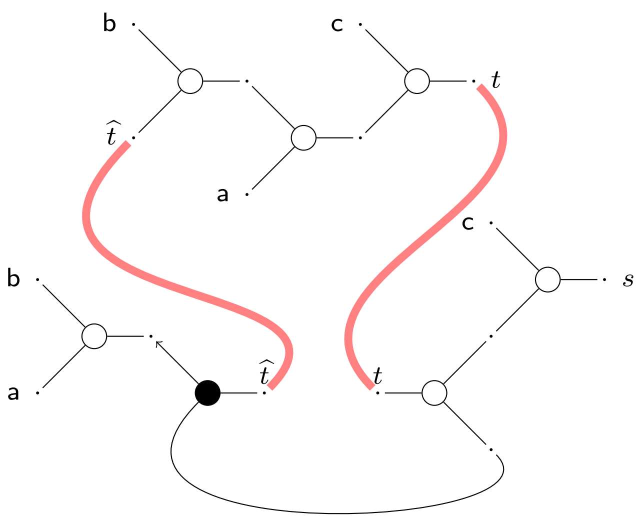




and adjoin into the initial graph for b :: $a\backslash((t \oslash (s/c)) \obslash \widehat{t})$

15. Deriving $aabbcc$: adjunction

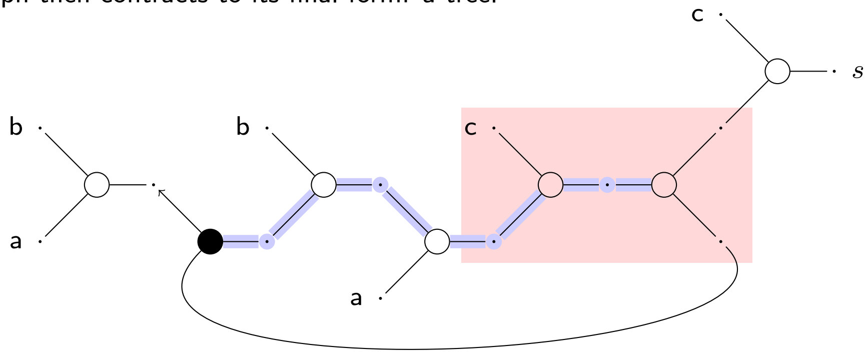
Step 2 To obtain $aabbcc$, take the contracted auxiliary graph of the previous slide



and adjoin into the initial graph for b :: $a\backslash((t \oslash (s/c)) \otimes \widehat{t})$

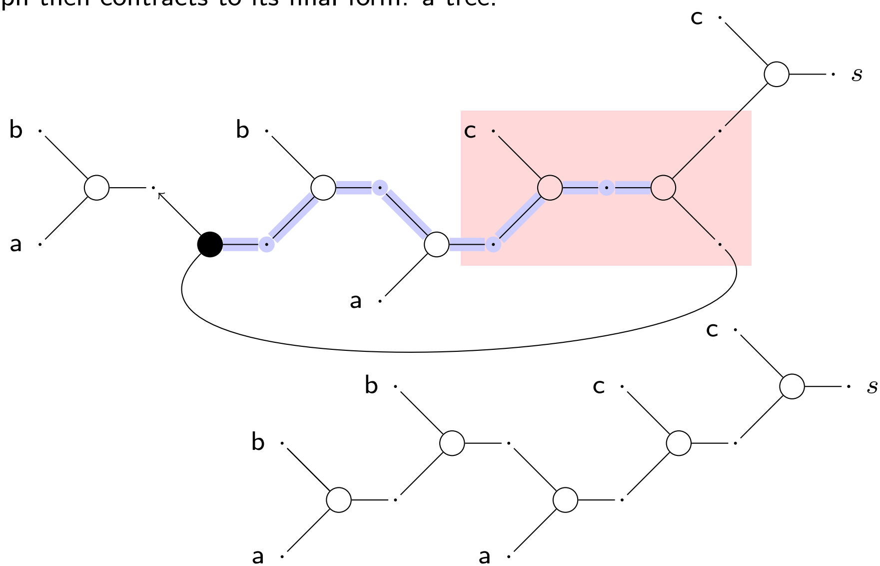
16. Deriving $aabbcc$: distribution

Step 3 In the rectangle is the input configuration for distribution. You can slide the rightmost tensor link to the matching cotensor link across the highlighted path. The graph then contracts to its final form: a tree.



16. Deriving $aabbcc$: distribution

Step 3 In the rectangle is the input configuration for distribution. You can slide the rightmost tensor link to the matching cotensor link across the highlighted path. The graph then contracts to its final form: a tree.



□

17. Beyond TAG cs

MIX has an equal number of a, b, c, in any order. Its recognition is beyond TAG.

$$\{w \in \{a,b,c\}^+ \mid |w|_a = |w|_b = |w|_c\}$$

LG Below an **LG** lexicon. Each entry abbreviates two type assignments: $\phi = s$ for an occurrence of the letter as the final item of the word, $\phi = s/s$ otherwise.

$$\begin{array}{lll} \mathsf{a} & :: & a \oslash \phi \\ \mathsf{b} & :: & \phi \oslash (s \obslash (a \obslash (s \oslash c))) \\ \mathsf{c} & :: & \phi \oslash c \end{array}$$

Idea: after distribution, antecedent $s/s, \ldots, s/s, s$ reducing to s which expands to context-free $a^n \; s \; (\psi \; c)^n$, where $\psi = s \obslash (a \obslash (s \oslash c))$.

17. Beyond TAG cs

MIX has an equal number of a, b, c, in any order. Its recognition is beyond TAG.

$$\{w \in \{a,b,c\}^+ \mid |w|_a = |w|_b = |w|_c\}$$

LG Below an **LG** lexicon. Each entry abbreviates two type assignments: $\phi = s$ for an occurrence of the letter as the final item of the word, $\phi = s/s$ otherwise.

$$\begin{array}{lcl} \mathsf{a} & :: & a \obslash \phi \\ \mathsf{b} & :: & \phi \oslash (s \obslash (a \obslash (s \oslash c))) \\ \mathsf{c} & :: & \phi \oslash c \end{array}$$

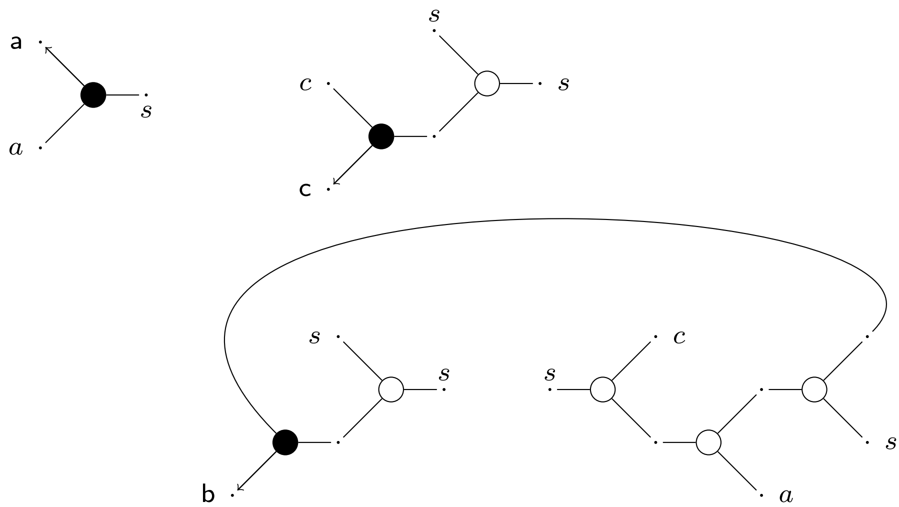
Idea: after distribution, antecedent $s/s, \ldots, s/s, s$ reducing to s which expands to context-free $a^n\ s\ (\psi\ c)^n$, where $\psi = s \obslash (a \obslash (s \oslash c))$.

Generalization (Melissen 2009) All languages which are the intersection of a context-free language and the permutation closure of a context-free language are recognizable in **LG**. (E.g. for $k = |\Sigma|$, k-MIX, counting dependencies $a_1^n \ldots a_k^n$).

Open question Upper bound **LG** recognition?

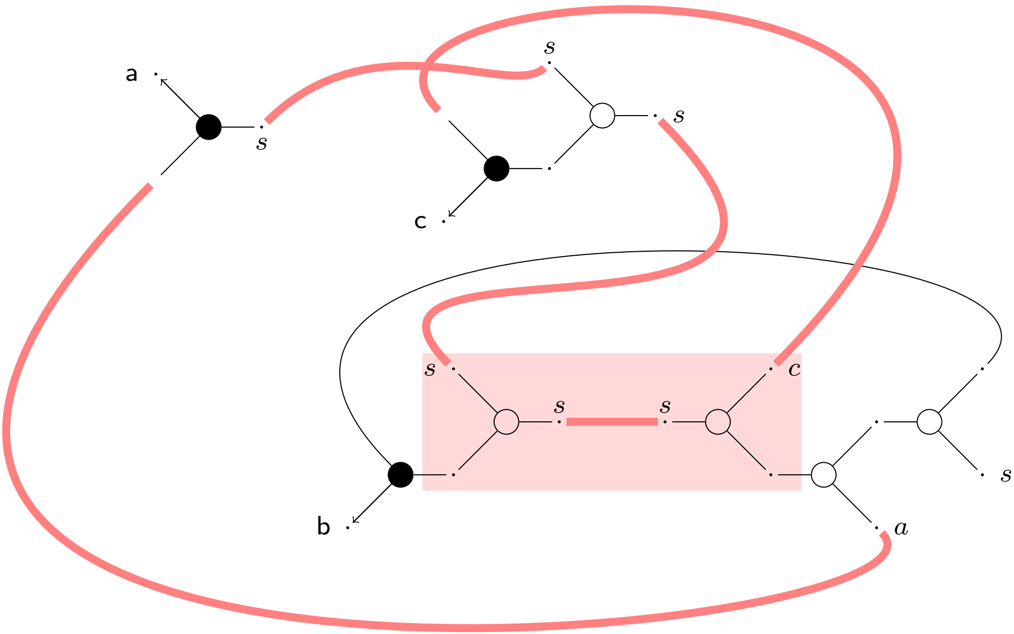
18. Connections for MIX

Below the partial nets for a :: $a \obslash s$, c :: $(s/s) \oslash c$, and b :: $(s/s) \oslash (s \obslash (a \obslash (s \oslash c)))$



18. Connections for MIX

Below the partial nets for a :: $a \obslash s$, c :: $(s/s) \oslash c$, and b :: $(s/s) \oslash (s \obslash (a \obslash (s \oslash c)))$
Connections producing the string bca. The input for distribution is highlighted.



19. Continuation semantics for LG

Bernardi & MM 2007, 2010, after Curien/Herbelin; Bastenhof 2010, after Andreoli. The program schematically:

$$\mathbf{LG}^{\mathcal{A}} \xrightarrow{\lceil\cdot\rceil} \mathbf{LP}^{\mathcal{A}\cup\{\bot\}}_{\times,\cdot^{\bot}} \xrightarrow{[\![\cdot]\!]} \mathbf{IL}^{\{e,t\}}$$

Two-step interpretation

- $\lceil\cdot\rceil$: double-negation/continuation-passing-style translation
  - maps multiple conclusion source logic to single conclusion linear logic/**LP**
  - response type $\bot$, linear products, negation $A^{\bot} \triangleq A \rightarrow \bot$

19. Continuation semantics for LG

Bernardi & MM 2007, 2010, after Curien/Herbelin; Bastenhof 2010, after Andreoli. The program schematically:

$$\mathbf{LG}^{\mathcal{A}} \xrightarrow{\lceil\cdot\rceil} \mathbf{LP}^{\mathcal{A}\cup\{\perp\}}_{\times,\cdot^{\perp}} \xrightarrow{[\![\cdot]\!]} \mathbf{IL}^{\{e,t\}}$$

Two-step interpretation

- $\lceil\cdot\rceil$: double-negation/continuation-passing-style translation
  - maps multiple conclusion source logic to single conclusion linear logic/**LP**
  - response type $\perp$, linear products, negation $A^{\perp} \triangleq A \rightarrow \perp$
- $[\![\cdot]\!]$: combining lexical with derivational semantics
  - atomic types: $[\![np]\!] = e$, $[\![s]\!] = [\![\perp]\!] = t$
  - terms: possible nonlinearity restricted to constants;

$$[\![(M\ N)]\!] = ([\![M]\!]\ [\![N]\!]) \quad ; \quad [\![\lambda x.M]\!] = \lambda\widetilde{x}.[\![M]\!]$$

20. LG display sequent calculus

Unfocused sequents statements $X \vdash Y$, with X (Y) input (output) structures.

$$\begin{array}{lcl} \mathcal{I} & ::= & x : A \mid \mathcal{I} \cdot \otimes \cdot \mathcal{I} \mid \mathcal{I} \cdot \oslash \cdot \mathcal{O} \mid \mathcal{O} \cdot \obslash \cdot \mathcal{I} \\ \mathcal{O} & ::= & \alpha : A \mid \mathcal{O} \cdot \oplus \cdot \mathcal{O} \mid \mathcal{I} \cdot \backslash \cdot \mathcal{O} \mid \mathcal{O} \cdot / \cdot \mathcal{I} \end{array}$$

20. LG display sequent calculus

Unfocused sequents statements $X \vdash Y$, with X (Y) input (output) structures.

$$\begin{array}{lcl} \mathcal{I} & ::= & x : A \mid \mathcal{I} \cdot \otimes \cdot \mathcal{I} \mid \mathcal{I} \cdot \oslash \cdot \mathcal{O} \mid \mathcal{O} \cdot \obslash \cdot \mathcal{I} \\ \mathcal{O} & ::= & \alpha : A \mid \mathcal{O} \cdot \oplus \cdot \mathcal{O} \mid \mathcal{I} \cdot \backslash \cdot \mathcal{O} \mid \mathcal{O} \cdot / \cdot \mathcal{I} \end{array}$$

Focus For the mapping to **LP**, we now allow at most one formula to be unlabeled; this formula is said to be in focus.

- the focus formula determines the type of the **LP** target term
- three types of sequents:
  - $X \vdash Y$, no formula in focus: domain of application of structural rules
  - $A \vdash Y$, focus left
  - $X \vdash B$, focus right

20. LG display sequent calculus

Unfocused sequents statements $X \vdash Y$, with X (Y) input (output) structures.

$$\begin{array}{lll} \mathcal{I} & ::= & x : A \mid \mathcal{I} \cdot\otimes\cdot \mathcal{I} \mid \mathcal{I} \cdot\oslash\cdot \mathcal{O} \mid \mathcal{O} \cdot\obslash\cdot \mathcal{I} \\ \mathcal{O} & ::= & \alpha : A \mid \mathcal{O} \cdot\oplus\cdot \mathcal{O} \mid \mathcal{I} \cdot\backslash\cdot \mathcal{O} \mid \mathcal{O} \cdot/\cdot \mathcal{I} \end{array}$$

Focus For the mapping to **LP**, we now allow at most one formula to be unlabeled; this formula is said to be in focus.

- the focus formula determines the type of the **LP** target term
- three types of sequents:
  - $X \vdash Y$, no formula in focus: domain of application of structural rules
  - $A \vdash Y$, focus left
  - $X \vdash B$, focus right

We first adjust the **LG** inference rules for the focus information. Then we impose the restrictions on the choice of the focus formula that lead to normal proofs.

21. Focus-sensitive rules

Axioms, cut

$$\frac{}{x : p \vdash p} \qquad \frac{X \vdash A \quad A \vdash Y}{X \vdash Y}\ \text{Cut} \qquad \frac{}{p \vdash \alpha : p}$$

21. Focus-sensitive rules

Axioms, cut

$$\frac{}{x : p \vdash p} \qquad \frac{X \vdash A \quad A \vdash Y}{X \vdash Y}\ \text{Cut} \qquad \frac{}{p \vdash \alpha : p}$$

Rewrite rules Composing a passive formula from passive subformulas. Examples:

$$\frac{X \vdash x : A \cdot \backslash \cdot \beta : B}{X \vdash \gamma : A \backslash B}\ \backslash R \qquad \frac{x : A \cdot \oslash \cdot \beta : B \vdash Y}{z : A \oslash B \vdash Y}\ \oslash L$$

21. Focus-sensitive rules

Axioms, cut

$$\frac{}{x : p \vdash p} \qquad \frac{X \vdash A \quad A \vdash Y}{X \vdash Y}\text{ Cut} \qquad \frac{}{p \vdash \alpha : p}$$

Rewrite rules Composing a passive formula from passive subformulas. Examples:

$$\frac{X \vdash x : A \cdot \backslash \cdot \beta : B}{X \vdash \gamma : A\backslash B}\ \backslash R \qquad \frac{x : A \cdot \oslash \cdot \beta : B \vdash Y}{z : A \oslash B \vdash Y}\ \oslash L$$

Monotonicity rules Focus propagates from conclusion to premises. Examples:

$$\frac{X \vdash A \qquad B \vdash Y}{A\backslash B \vdash X \cdot \backslash \cdot Y}\ \backslash L \qquad \frac{X \vdash A \qquad B \vdash Y}{X \cdot \oslash \cdot Y \vdash A \oslash B}\ \oslash R$$

22. (De)focusing

To connect the different stages of a proof, we need rules for (de)focusing a formula.

$$\frac{A \vdash Y}{x : A \vdash Y}\ \widetilde{\mu}^* \qquad\qquad \frac{X \vdash A}{X \vdash \alpha : A}\ \mu^*$$

$$\frac{x : A \vdash Y}{A \vdash Y}\ \widetilde{\mu} \qquad\qquad \frac{X \vdash \alpha : A}{X \vdash A}\ \mu$$

22. (De)focusing

To connect the different stages of a proof, we need rules for (de)focusing a formula.

$$\frac{A \vdash Y}{x : A \vdash Y}\ \widetilde{\mu}^* \qquad \frac{X \vdash A}{X \vdash \alpha : A}\ \mu^*$$

$$\frac{x : A \vdash Y}{A \vdash Y}\ \widetilde{\mu} \qquad \frac{X \vdash \alpha : A}{X \vdash A}\ \mu$$

In the presence of $\mu^{(*)}/\widetilde{\mu}^{(*)}$ one can do with one axiom schema. For example:

$$\frac{}{p \vdash \alpha : p}\ ? \qquad \rightsquigarrow \qquad \cfrac{\cfrac{\overline{x : p \vdash p}}{x : p \vdash \alpha : p}\ \mu^*}{p \vdash \alpha : p}\ \widetilde{\mu}$$

23. Sample derivation

In the following derivation, the focus formula is highlighted.

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{\overline{n^{y} \vdash n} \qquad \overline{np \vdash np^{\beta}}}{np/n \vdash np^{\beta} \cdot / \cdot n^{y}}\ /L
        }{(np/n)^{x} \vdash np^{\beta} \cdot / \cdot n^{y}}\ \widetilde{\mu}^{*}
      }{(np/n)^{x} \cdot \otimes \cdot n^{y} \vdash np^{\beta}}\ r
    }{(np/n)^{x} \cdot \otimes \cdot n^{y} \vdash np}\ \mu
    \qquad \overline{s \vdash s^{\alpha}}
  }{np\backslash s \vdash ((np/n)^{x} \cdot \otimes \cdot n^{y}) \cdot \backslash \cdot s^{\alpha}}\ \backslash L
}{
  \dfrac{
    \dfrac{(np\backslash s)^{z} \vdash ((np/n)^{x} \cdot \otimes \cdot n^{y}) \cdot \backslash \cdot s^{\alpha}}{((np/n)^{x} \cdot \otimes \cdot n^{y}) \cdot \otimes \cdot (np\backslash s)^{z} \vdash s^{\alpha}}\ r
  }{((np/n)^{x} \cdot \otimes \cdot n^{y}) \cdot \otimes \cdot (np\backslash s)^{z} \vdash s}\ \mu
}\ \widetilde{\mu}^{*}
$$

As long as the choice of the focus formula is free, there is another derivation, that first focuses on np/n . . . the spurious ambiguity problem.

24. Restricting (de)focusing

Complementary to the distinction between input/output structures, we distinguish input (negative) and output (positive) formulas:

$$\begin{array}{lcl} \text{(negative)} \quad \mathcal{I}^f & ::= & A \oplus B \mid A \backslash B \mid B / A \\ \text{(positive)} \quad \mathcal{O}^f & ::= & A \otimes B \mid A \oslash B \mid B \obslash A \end{array}$$

- $\mathcal{I}^f$ (negative): monotonicity rule is sequent (L) rule
- $\mathcal{O}^f$ (positive): monotonicity rule is sequent (R) rule

24. Restricting (de)focusing

Complementary to the distinction between input/output structures, we distinguish input (negative) and output (positive) formulas:

$$\begin{array}{llll}\text{(negative)} & \mathcal{I}^f & ::= & A \oplus B \mid A \backslash B \mid B / A \\ \text{(positive)} & \mathcal{O}^f & ::= & A \otimes B \mid A \oslash B \mid B \obslash A\end{array}$$

- $\mathcal{I}^f$ (negative): monotonicity rule is sequent (L) rule
- $\mathcal{O}^f$ (positive): monotonicity rule is sequent (R) rule

Conditions on (de)focusing $\mu, \widetilde{\mu}^*$: provided $A \in \mathcal{I}^f$; $\widetilde{\mu}, \mu^*$: provided $A \in \mathcal{O}^f \cup \mathcal{A}$.

$$\frac{A \vdash Y}{x : A \vdash Y}\ \widetilde{\mu}^* \qquad \frac{X \vdash A}{X \vdash \alpha : A}\ \mu^*$$

$$\frac{x : A \vdash Y}{A \vdash Y}\ \widetilde{\mu} \qquad \frac{X \vdash \alpha : A}{X \vdash A}\ \mu$$

25. Pruning effect

The derivation on the right violates the formula restriction on the (μ) rule: $np \notin \mathcal{I}^f$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot n \cdot \vdash n} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot np \cdot \vdash np} \qquad \dfrac{\dfrac{\dfrac{}{\cdot s \cdot \vdash s}}{\cdot s \cdot \vdash \cdot s \cdot}\,\mu^*}{s \vdash \cdot s \cdot}\,\widetilde{\mu}}{np \backslash s \vdash np \cdot \backslash \cdot s}\,\backslash L}{\cdot (np \backslash s) \cdot \vdash np \cdot \backslash \cdot s}\,\widetilde{\mu}^*}{np \cdot \otimes \cdot (np \backslash s) \vdash \cdot s \cdot}\,r}{\cdot np \cdot \vdash s \cdot / \cdot (np \backslash s)}\,r}{np \vdash s \cdot / \cdot (np \backslash s)}\,\widetilde{\mu}}{np/n \vdash (s \cdot / \cdot (np \backslash s)) \cdot / \cdot n}\,/L}{\cdot (np/n) \cdot \vdash (s \cdot / \cdot (np \backslash s)) \cdot / \cdot n}\,\widetilde{\mu}^*}{(np/n) \cdot \otimes \cdot n \vdash s \cdot / \cdot (np \backslash s)}\,r}{((np/n) \cdot \otimes \cdot n) \cdot \otimes \cdot (np \backslash s) \vdash \cdot s \cdot}\,r
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\lightning}{(np/n) \cdot \otimes \cdot n \vdash np}\,\mu \qquad \dfrac{\dfrac{\dfrac{}{\cdot s \cdot \vdash s}}{\cdot s \cdot \vdash \cdot s \cdot}\,\mu^*}{s \vdash \cdot s \cdot}\,\widetilde{\mu}}{np \backslash s \vdash ((np/n) \cdot \otimes \cdot n) \cdot \backslash \cdot s}\,\backslash L}{\cdot (np \backslash s) \cdot \vdash ((np/n) \cdot \otimes \cdot n) \cdot \backslash \cdot s}\,\widetilde{\mu}^*}{((np/n) \cdot \otimes \cdot n) \cdot \otimes \cdot (np \backslash s) \vdash \cdot s \cdot}\,r
$$

Remark $\mathbf{L}^*$ allows the derivation on the right, and breaks off the one on the left.

26. Focus shifting

We compile a branch from $(\widetilde{\mu}^*)$ via a sequence (possibly empty) of structural rules and rewrite rules to (μ) in a derived inference rule with the $\widetilde{\mu}^*$ restrictions on A and the μ restrictions on B.

$$
\begin{array}{c}
\dfrac{A \vdash Y}{x : A \vdash Y}\ \widetilde{\mu}^* \\
\vdots \\
(res, distr, rewrite) \\
\vdots \\
\dfrac{X \vdash \beta : B}{X \vdash B}\ \mu
\end{array}
\qquad \rightsquigarrow \qquad
\dfrac{A \vdash Y}{X \vdash B} \leftrightharpoons
$$

26. Focus shifting

We compile a branch from $(\widetilde{\mu}^*)$ via a sequence (possibly empty) of structural rules and rewrite rules to (μ) in a derived inference rule with the $\widetilde{\mu}^*$ restrictions on A and the μ restrictions on B.

$$\begin{array}{c} \dfrac{A \vdash Y}{x : A \vdash Y}\ \widetilde{\mu}^* \\ \vdots \\ (res, distr, rewrite) \\ \vdots \\ \dfrac{X \vdash \beta : B}{X \vdash B}\ \mu \end{array} \quad \rightsquigarrow \quad \dfrac{A \vdash Y}{X \vdash B} \leftrightharpoons$$

For the four combinations of $\mu^*, \widetilde{\mu}^*$ and $\mu, \widetilde{\mu}$, this results in the following rules.

$$\dfrac{A \vdash Y}{X \vdash B} \leftrightharpoons \qquad \dfrac{X' \vdash A}{X \vdash B} \Rightarrow \qquad \dfrac{X \vdash A}{B \vdash Y} \rightleftharpoons \qquad \dfrac{A \vdash Y'}{B \vdash Y} \Leftarrow$$

Remark For the endsequent, we can relax the formula restriction on B.

27. Sample derivation: focus shifting

Compare the verbose derivation of the left with the result of compiling away the display equivalences.

$$
\dfrac{
  \dfrac{}{\cdot n \cdot \vdash n}
  \qquad
  \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{
    \dfrac{}{\cdot np \cdot \vdash np}
    \qquad
    \dfrac{\dfrac{\dfrac{}{\cdot s \cdot \vdash s}}{\cdot s \cdot \vdash \cdot s \cdot}\,\mu^*}{s \vdash \cdot s \cdot}\,\widetilde{\mu}
  }{np\backslash s \vdash np \cdot \backslash \cdot s}\,\backslash L
  }{\cdot (np\backslash s) \cdot \vdash np \cdot \backslash \cdot s}\,\widetilde{\mu}^*
  }{np \cdot \otimes \cdot (np\backslash s) \vdash \cdot s \cdot}\,r
  }{\cdot np \cdot \vdash s \cdot / \cdot (np\backslash s)}\,r
  }{np \vdash s \cdot / \cdot (np\backslash s)}\,\widetilde{\mu}
}{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{np/n \vdash (s \cdot / \cdot (np\backslash s)) \cdot / \cdot n}{\cdot (np/n) \cdot \vdash (s \cdot / \cdot (np\backslash s)) \cdot / \cdot n}\,\widetilde{\mu}^*
      }{(np/n) \cdot \otimes \cdot n \vdash s \cdot / \cdot (np\backslash s)}\,r
    }{((np/n) \cdot \otimes \cdot n) \cdot \otimes \cdot (np\backslash s) \vdash \cdot s \cdot}\,r
  }{((np/n) \cdot \otimes \cdot n) \cdot \otimes \cdot (np\backslash s) \vdash s}\,\mu
}\,/L
$$

$$
\dfrac{
  \dfrac{
    \dfrac{}{\cdot n \cdot \vdash n}
    \qquad
    \dfrac{\dfrac{
      \dfrac{}{\cdot np \cdot \vdash np}
      \qquad
      \dfrac{\dfrac{}{\cdot s \cdot \vdash s}}{s \vdash \cdot s \cdot}\,\rightleftharpoons
    }{np\backslash s \vdash np \cdot \backslash \cdot s}\,\backslash L
    }{np \vdash s \cdot / \cdot np\backslash s}\,\Leftarrow
  }{np/n \vdash (s \cdot / \cdot np\backslash s) \cdot / \cdot n}\,/L
}{(np/n \cdot \otimes \cdot n) \cdot \otimes \cdot np\backslash s \vdash s}\,\leftrightharpoons
$$

28. From normal LG proofs to LP terms

For normal **LG** derivations, we have the following term construction rules:

- monotonicity rules: linear pairs $\langle M, N \rangle$
- rewrite rules: **case** ξ **of** $\langle \phi, \psi \rangle$ **in** M
- $\widetilde{\mu}^*, \mu^*$: linear application $(x\ M)$, $(\alpha\ M)$
- $\widetilde{\mu}, \mu$: linear abstraction $\lambda x.M$, $\lambda \alpha.M$

28. From normal LG proofs to LP terms

For normal **LG** derivations, we have the following term construction rules:

- monotonicity rules: linear pairs $\langle M, N \rangle$
- rewrite rules: **case** ξ **of** $\langle \phi, \psi \rangle$ **in** M
- $\widetilde{\mu}^*, \mu^*$: linear application $(x\ M)$, $(\alpha\ M)$
- $\widetilde{\mu}, \mu$: linear abstraction $\lambda x.M$, $\lambda \alpha.M$

$$\frac{X \vdash A \qquad B \vdash Y}{A \backslash B \vdash X \cdot \backslash \cdot Y} \backslash L \qquad \frac{X \vdash x : A \cdot \backslash \cdot \beta : B}{X \vdash \gamma : A \backslash B} \backslash R$$

$$\lceil \backslash L \rceil = \langle M, N \rangle \qquad \lceil \backslash R \rceil = \textbf{case } \gamma \textbf{ of } \langle x, \beta \rangle \textbf{ in } M$$

$$\frac{A \vdash Y}{x : A \vdash Y} \widetilde{\mu}^* \quad \lceil \widetilde{\mu}^* \rceil = (x\ M) \qquad \frac{X \vdash A}{X \vdash \alpha : A} \mu^* \quad \lceil \mu^* \rceil = (\alpha\ M)$$

$$\frac{x : A \vdash Y}{A \vdash Y} \widetilde{\mu} \quad \lceil \widetilde{\mu} \rceil = \lambda x.M \qquad \frac{X \vdash \alpha : A}{X \vdash A} \mu \quad \lceil \mu \rceil = \lambda \alpha.M$$

29. Computing the proof term

We calculate the **LP** proof term for our example.

$$
\dfrac{\dfrac{}{\cdot\, n \,\cdot \vdash n} \qquad \dfrac{\dfrac{\dfrac{}{\cdot\, np \,\cdot \vdash np} \qquad \dfrac{\dfrac{}{\cdot\, s \,\cdot \vdash s}}{s \vdash \cdot\, s \,\cdot}\rightleftharpoons}{np\backslash s \vdash np \cdot\backslash\cdot s}\backslash L}{np \vdash s \cdot/\cdot np\backslash s}\Leftarrow}{\dfrac{np/n \vdash (s \cdot/\cdot np\backslash s) \cdot/\cdot n}{((np/n)^{x} \cdot\otimes\cdot n^{y}) \cdot\otimes\cdot (np\backslash s)^{z} \vdash s}\leftrightharpoons}/L
$$

$$
\dfrac{\dfrac{}{y} \qquad \dfrac{\dfrac{\dfrac{}{v} \qquad \dfrac{\dfrac{}{u}}{\lambda u.(\alpha\ u)}\rightleftharpoons}{\langle v, \lambda u.(\alpha\ u)\rangle}\backslash L}{\lambda v.(z\ \langle v, \lambda u.(\alpha\ u)\rangle)}\Leftarrow}{\dfrac{\langle y, \lambda v.(z\ \langle v, \lambda u.(\alpha\ u)\rangle)\rangle}{\lambda \alpha.(x\ \langle y, \lambda v.(z\ \langle v, \lambda u.(\alpha\ u)\rangle)\rangle)}\leftrightharpoons}/L
$$

29. Computing the proof term

We calculate the **LP** proof term for our example.

$$
\dfrac{\dfrac{}{\cdot\, n \cdot \vdash n} \qquad \dfrac{\dfrac{\dfrac{}{\cdot\, np \cdot \vdash np} \qquad \dfrac{\dfrac{}{\cdot\, s \cdot \vdash s}}{s \vdash \cdot\, s \,\cdot}\rightleftharpoons}{\dfrac{np\backslash s \vdash np \cdot \backslash \cdot s}{np \vdash s \cdot / \cdot np\backslash s}\Leftarrow}\backslash L}{\dfrac{np/n \vdash (s \cdot / \cdot np\backslash s) \cdot / \cdot n}{((np/n)^{x} \cdot \otimes \cdot n^{y}) \cdot \otimes \cdot (np\backslash s)^{z} \vdash s}\leftrightharpoons}/L
\qquad
\dfrac{\dfrac{}{y} \qquad \dfrac{\dfrac{\dfrac{}{v} \qquad \dfrac{\dfrac{}{u}}{\lambda u.(\alpha\ u)}\rightleftharpoons}{\dfrac{\langle v, \lambda u.(\alpha\ u)\rangle}{\lambda v.(z\ \langle v, \lambda u.(\alpha\ u)\rangle)}\Leftarrow}\backslash L}{\dfrac{\langle y, \lambda v.(z\ \langle v, \lambda u.(\alpha\ u)\rangle)\rangle}{\lambda \alpha.(x\ \langle y, \lambda v.(z\ \langle v, \lambda u.(\alpha\ u)\rangle)\rangle)}\leftrightharpoons}/L
$$

The final result can be simplified by η conversion, and by applying some canonical isomorphisms to get rid of the pairs:

$$A \times B \to C \quad \longrightarrow_{\mathsf{curry}} \quad A \to B \to C \quad \longrightarrow_{\mathsf{swap}} \quad B \to A \to C$$

$$\lambda\alpha.(x\ \langle y, \lambda v.(z\ \langle v, \lambda u.(\alpha\ u)\rangle)\rangle) \longrightarrow_{\eta,\mathsf{swap}\circ\mathsf{curry}} \lambda\alpha.(x\ (z\ \alpha)\ y)$$

30. Lexical insertion

Typing the proof term Here is what the **LP** typing rules for $\lambda\alpha.(x^{\mathsf{some}}\ (z^{\mathsf{left}}\ \alpha)\ y^{\mathsf{student}})$ tell us about $\lceil\cdot\rceil$.

$$\begin{array}{rcl} \text{some} : \lceil np/n \rceil & = & \lceil np \rceil^{\perp} \to \lceil n \rceil^{\perp} \\ \text{student} : \lceil n \rceil & = & \lceil n \rceil \\ \text{left} : \lceil np \backslash s \rceil & = & \lceil s \rceil^{\perp} \to \lceil np \rceil^{\perp} \end{array}$$

Lexical insertion The second stage of the interpretation is the substitution of lexical terms for the parameters (variables that remain unbound) of the **LP** proof term.

Here are translations respecting $\lceil\cdot\rceil$, assuming $[\![\perp]\!] = [\![s]\!] = t$, $[\![np]\!] = e$, and $[\![n]\!] = e \to t$ and nonlogical constants $\textsc{student}$,$\textsc{left}$ with the indicated type.

$$[\![\cdot]\!] : \quad \begin{array}{rcl} \text{some} & \mapsto & \lambda P \lambda Q.(\exists\ \lambda x.((Q\ x) \wedge (P\ x))) \\ \text{student} & \mapsto & \textsc{student}^{e \to t} \\ \text{left} & \mapsto & \lambda c \lambda x.(c\ (\textsc{left}^{e \to t}\ x)) \end{array}$$

$$[\![\lambda\alpha.(x^{\mathsf{some}}\ (z^{\mathsf{left}}\ \alpha)\ y^{\mathsf{student}})]\!] = \lambda c.(\exists \lambda x.((\textsc{student}\ x) \wedge (c\ (\textsc{left}\ x))))$$

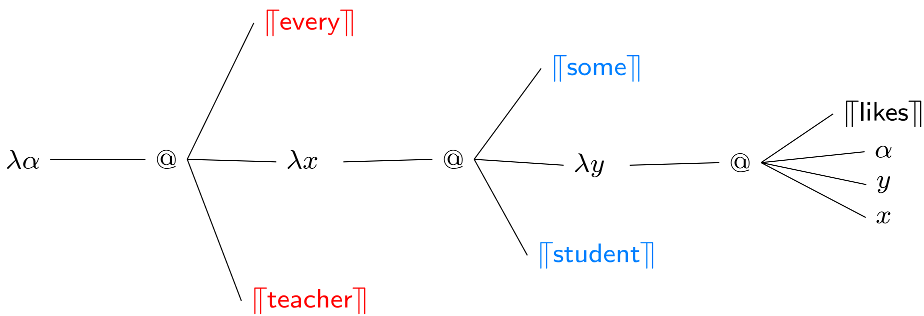
Remark c of type $[\![\lceil s \rceil^{\perp}]\!] = t \to t$, i.e. abstraction over a sentence continuation.

31. Illustration: quantifier scope

The 2-QP sentence below allows for two focused **LG** proofs.

$$((np/n)^{\text{every}} \cdot \otimes \cdot n^{\text{teacher}}) \cdot \otimes \cdot (((np\backslash s)/np)^{\text{likes}} \cdot \otimes \cdot ((np/n)^{\text{some}} \cdot \otimes \cdot n^{\text{student}})) \vdash s$$

M_1 :



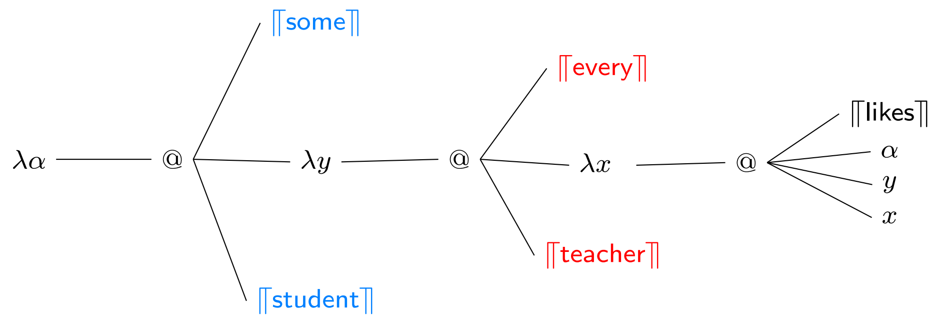
With $[\![\text{likes}]\!] = \lambda c \lambda y \lambda x.(c\ (\text{LIKES}^{e \to e \to t}\ y\ x))$, we obtain the familiar surface (M_1) and inverted (M_2) reading.

31. Illustration: quantifier scope

The 2-QP sentence below allows for two focused **LG** proofs.

$$((np/n)^{\mathsf{every}} \cdot \otimes \cdot n^{\mathsf{teacher}}) \cdot \otimes \cdot (((np\backslash s)/np)^{\mathsf{likes}} \cdot \otimes \cdot ((np/n)^{\mathsf{some}} \cdot \otimes \cdot n^{\mathsf{student}})) \vdash s$$

M_2 :



With $[\![\mathsf{likes}]\!] = \lambda c\lambda y\lambda x.(c\ (\textsc{likes}^{e\to e\to t}\ y\ x))$, we obtain the familiar surface (M_1) and inverted (M_2) reading.

32. Conclusions

The symmetric Lambek-Grishin calculus offers powerful tools to tackle the expressive limitations of the original Lambek calculi:

- Form
  - logical distributivity laws relating dual families
  - natural analysis for non-CF patterns
- Meaning
  - continuation semantics for multiple-conclusion source calculus
  - optimizes division of labour between syntax and semantics

More to explore Categorial type logics. Chapter update. Handbook of Logic and Language, 2nd edition. Elsevier, 2011.

□