On combining grammar logics

Michael Moortgat

Columbus, 22 September 2014

Abstract

Current categorial grammar frameworks are generally obtained by mixing different logics: multimodal and hybrid typelogical grammar, displacement calculus, bilinear Lambek calculus (Lambek-Grishin) are representative examples.

In this talk I discuss two strategies for combining logics. One approach is to put together formulas with a distinct interpretation in one encompassing logic. A key example is Girard's "unity of logic" framework where reusable and non-reusable resources are put together, giving rise to two notions of implication (intuitionistic/linear) with the "!" operation establishing the communication between stable and non-reusable resources. Executing the same program for the grammar logics leads to a finegrained inventory of implications with Lambek's slashes and the non-directional implication of linear logic at the extremities.

The second strategy is to switch the focus from single logics to pairs (source, target) and structure-preserving mappings between them. Central issues here are: what is the information one wants to encode at the source end, and what properties of compositionality are delegated to the functorial passage from source to target.

Plan

- From IL to linear logic: finite resources
- From linear to grammar logic: precedence, dominance
- Atoms of variation: decomposing structural rules
- Lab session

Reading For the logical background: Wadler, A taste of linear logic.
Further useful background reading: the entries

substructural logics

typelogical grammar

in Stanford Encyclopedia of Philosopy (SEP, http://plato.stanford.edu).

Resource-sensitivity: motivation

Rules for valid reasoning come in two types:

- logical rules: how to use and derive complex propositions out of simpler ones
- structural rules: how to manipulate assumptions in constructing a proof. For example: by copying, deleting, permuting them

Substructural logics result from dropping some/all of the structural rules that traditional logic adopts.

- no copying/deletion: assumptions become resources, used up in reasoning; no permutation: the linear order of resources matters
- appropriate where logic is used to model cognitive processes, processes that depend on limited computational means.

$\rightsquigarrow$ logics of perception, action, natural language syntax and semantics (but also: economics, games, quantum mechanics ...)

Standard logic — the intuitionistic case $\mathbf{IL}_{\to,\times}$

Formulas Let's just look at conjunction and implication for a start.

$$A, B ::= p \mid A \times B \mid A \to B$$

- p : atomic propositions
- $A \times B$: conjunction, 'A and B'
- $A \to B$: implication, 'if A then B'

Judgements We restrict to intuitionistic sequents $\Gamma \vdash A$ with

Γ: a list of zero or more propositions

A: a single conclusion derived from Γ (compare: classical $\Gamma \vdash \Delta$)

Remark This is already a substructural logic — see why?

$\mathbf{IL}_{\to,\times}$: natural deduction rules

$$\frac{}{A \vdash A}\ \text{Ax}$$

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}\ \text{Exchange} \qquad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\ \text{Contraction} \qquad \frac{\Gamma \vdash B}{\Gamma, A \vdash B}\ \text{Weakening}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \to I \qquad \frac{\Gamma \vdash A \to B \quad \Delta \vdash A}{\Gamma, \Delta \vdash B} \to E$$

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \times B} \times I \qquad \frac{\Gamma \vdash A \times B \quad \Delta, A, B \vdash C}{\Delta, \Gamma \vdash C} \times E$$

- Exchange: order of assumptions doesn't matter
- Contraction: assumptions are re-usable
- Weakening: assumptions can be thrown away

Another conjunction?

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \times B} \times I' \qquad \frac{\Gamma \vdash A \times B}{\Gamma \vdash A} \times E'_1 \qquad \frac{\Gamma \vdash A \times B}{\Gamma \vdash B} \times E'_2$$

Not really . . . In the presence of Contraction, Weakening, (and Exchange) the different formulations are interderivable. For example, obtaining $(\times' I)$ from $(\times I)$

$$\dfrac{\dfrac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma, \Gamma \vdash A \times B} \times I}{\Gamma \vdash A \times B} \text{ Contraction}$$

This shows that the structural rules blur the picture of what logical constants one can distinguish.

A turning point: Girard, Linear Logic

*Because of its length and novelty this paper has not been subjected to the normal process of refereeing. The editor is prepared to share with the author any criticism that eventually will be expressed concerning this work.

Theoretical Computer Science 50(1987) 1–102

Actions vs situations

For reasoning about a dynamic world, we need notions of 'state' and non-monotonicity.

"Classical and intuitionistic logics deal with stable truths :

if A and $A \Rightarrow B$, then B, but A still holds.

This is perfect in mathematics, but wrong in real life, since real implication is causal. A causal implication cannot be iterated since the conditions are modified after its use ; this process of modification of the premises (conditions) is known in physics as reaction.

For instance, if A is to spend \$1 on a pack of cigarettes and B is to get them, you lose \$1 in this process, and you cannot do it a second time. The reaction here was that \$1 went out of your pocket."

Girard, Linear logic, its syntax and semantics

ILL: Intuitionistic Linear logic

- Contraction, Weakening are no longer freely available:

  assumptions become finite, material resources

  inference rules depending on shared or disjoint contexts

Formulas

$$A, B ::= p \mid A \otimes B \mid A \,\&\, B \mid A \multimap B \mid !\,A \mid \ldots$$

- $A \otimes B$: no sharing, 'both A and B', multiplicative conjunction
- $A \,\&\, B$: complete sharing, 'choose from A and B', additive conjunction
- $A \multimap B$: linear implication, 'consume resource A producing B
- $!\,A$: iterability of action A, absence of reaction

Linear logic: natural deduction rules

The fragment $\mathsf{IL}_{\multimap,\otimes,\&}$:

$$\frac{}{A \vdash A}\ \mathsf{Ax} \qquad \frac{\Gamma, B, A, \Delta \vdash C}{\Gamma, A, B, \Delta \vdash C}\ \mathsf{Exchange}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ \multimap I \qquad \frac{\Gamma \vdash A \multimap B \quad \Delta \vdash A}{\Gamma, \Delta \vdash B}\ \multimap E$$

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\ \otimes I \qquad \frac{\Gamma \vdash A \otimes B \quad \Delta, A, B \vdash C}{\Gamma, \Delta \vdash C}\ \otimes E$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \,\&\, B}\ \& I \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash A}\ \& E_1 \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash B}\ \& E_2$$

Reunification

"Now by the end of the century we are faced with an incredible number of logics [...].

Is logic still about pure reasoning? In other words, could there by a way to reunify logical systems — let us say those systems with a good sequent calculus — into a single sequent calculus ? Is it possible to handle the (legitimate) distinction classical/intuitionistic not through a change of system, but through a change of formulas? Is it possible to obtain classical effects by a restriction to classical formulas? Etc.

Of course, there surely are ways to achieve this by cheating, typically by considering a disjoint union of systems. However, all these jokes will be made impossible if we insist on the fact that the various systems represented should communicate freely [...]."

Girard 1993: On the unity of logic

Universal Logic

Linear and intuitionistic constants coexisting in one 'universal logic'.

- assumptions are sorted as linear: $\langle A \rangle$ or intuitionistic: $[A]$
- two kinds of axioms: intuitionistic versus linear
- Contraction/Weakening come back in a controlled form: $!\,A$ ('of course A')

$$\frac{}{\langle A \rangle \vdash A}\ \langle \mathsf{Ax} \rangle \qquad \frac{}{[A] \vdash A}\ [\mathsf{Ax}] \qquad \frac{\Gamma, \Delta \vdash A}{\Delta, \Gamma \vdash A}\ \text{Exchange}$$

$$\frac{\Gamma, [A], [A] \vdash B}{\Gamma, [A] \vdash B}\ \text{Contraction} \qquad \frac{\Gamma \vdash B}{\Gamma, [A] \vdash B}\ \text{Weakening}$$

$$\frac{[\Gamma] \vdash A}{[\Gamma] \vdash\, !\,A}\ !\,I \qquad \frac{\Delta \vdash\, !\,A \quad \Gamma, [A] \vdash B}{\Gamma, \Delta \vdash B}\ !\,E$$

Universal Logic, cont'd

The linear logical rules for $\multimap, \otimes, \&$, with sort annotation of the assumptions:

$$\frac{\Gamma, \langle A \rangle \vdash B}{\Gamma \vdash A \multimap B} \multimap I \qquad \frac{\Gamma \vdash A \multimap B \quad \Delta \vdash A}{\Gamma, \Delta \vdash B} \multimap E$$

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} \otimes I \qquad \frac{\Gamma \vdash A \otimes B \quad \Delta, \langle A \rangle, \langle B \rangle \vdash C}{\Gamma, \Delta \vdash C} \otimes E$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \,\&\, B} \& I \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash A} \& E_1 \qquad \frac{\Gamma \vdash A \,\&\, B}{\Gamma \vdash B} \& E_2$$

For good measure, the rules for $\oplus$: additive disjunction, dual of $\&$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B} \oplus I_1 \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B} \oplus I_2 \qquad \frac{\Gamma \vdash A \oplus B \quad \Delta, \langle A \rangle \vdash C \quad \Delta, \langle B \rangle \vdash C}{\Gamma, \Delta \vdash C} \oplus E$$

Wadler, A taste of linear logic

Communication

- removing free copying/deletion splits $\times$ into multiplicative $\otimes$ versus additive $\&$
- the fission of $\times$ creates a new logical constant: !
- ! re-establishes communication between $\&$ (sharing) vs $\otimes$ (no sharing)

$$\langle\, !(A \,\&\, B)\, \rangle \vdash\, !\,A \otimes\, !\,B \qquad \langle\, !\,A \otimes\, !\,B\, \rangle \vdash\, !(A \,\&\, B)$$

Exercise prove these equivalences.

Intuition: if you are repeatedly given the choice between A and B, this is the same as repeatedly being presented with A and repeatedly being offered B.

Recovering lost expressivity

ILL has a more finegrained view on the logical constants than IL.

But thanks to !, no expressivity is lost:

$$\Gamma \vdash A \text{ is provable intuitionistically iff } (\Gamma \vdash A)^* \text{ is provable in linear logic}$$

Embedding translation For atomic formulas, $p^* = p$. For complex formulas

$$\begin{array}{rcl} (A \to B)^* & = & !\,A^* \multimap B^* \\ (A \times B)^* & = & A^* \,\&\, B^* \\ (A + B)^* & = & !\,A^* \oplus !\,B^* \end{array}$$

The translation is extended to sequents: $(\Gamma \vdash A)^* = [\Gamma^*] \vdash A^*$.

Exercise Show that the intuitionistic rules for $\times$, $+$ (renewable resources) can be derived from the corresponding rules of linear logic, together with the ! intro/elim rules.

Proofs as programs

All that was said about proofs can also be expressed symbolically via terms labeling these proofs.

Curry-Howard isomorphism between ILL and linear lambda calculus.

$$\frac{}{\langle x : A \rangle \vdash x : A}\ \langle Id \rangle \qquad \frac{}{[x : A] \vdash x : A}\ [Id]$$

$$\frac{\Gamma, [y : A], [z : A] \vdash u : B}{\Gamma, [x : A] \vdash u[x/y, x/z] : B}\ C \qquad \frac{\Gamma \vdash u : B}{\Gamma, [x : A] \vdash u : B}\ W$$

$$\frac{[\Gamma] \vdash t : A}{[\Gamma] \vdash {!}\,t : {!}\,A}\ {!}\,I \qquad \frac{\Gamma \vdash s : {!}\,A \quad \Delta, [x : A] \vdash u : B}{\Gamma, \Delta \vdash \textbf{case } s \textbf{ of } {!}\,x \textbf{ in } u : B}\ {!}\,E$$

$$\frac{\Gamma, \langle x : A \rangle \vdash u : B}{\Gamma \vdash \lambda\langle x \rangle.u : A \multimap B}\ \multimap I \qquad \frac{\Gamma \vdash s : A \multimap B \quad \Delta \vdash t : A}{\Gamma, \Delta \vdash s\langle t \rangle : B}\ \multimap E$$

Example: eco-programming

The IL program below uses the x parameter in a non-linear way, duplicating it.

$$\vdash \lambda f.\lambda x.f(x)(x) : (A \to (A \to B)) \to (A \to B)$$

Resource-conscious version narrowing down reuseability to the $!\,A$ formula.

$$\lambda\langle f\rangle\lambda\langle y\rangle.\textbf{case } y \textbf{ of } !\,x \textbf{ in } f\langle x\rangle\langle x\rangle$$

$$\dfrac{\dfrac{}{\langle !\,A\rangle \vdash !\,A}\,\langle\mathsf{Id}\rangle \quad \dfrac{\dfrac{\dfrac{}{[A] \vdash A}\,\langle\mathsf{Id}\rangle \quad \dfrac{}{[A] \vdash A}\,\langle\mathsf{Id}\rangle \quad \dfrac{}{\langle A \multimap (A \multimap B)\rangle \vdash A \multimap (A \multimap B)}\,\langle\mathsf{Id}\rangle}{[A],[A],\langle A \multimap (A \multimap B)\rangle \vdash B}\,\multimap E}{[A],\langle A \multimap (A \multimap B)\rangle \vdash B}\,\mathsf{Contr}}{\dfrac{\dfrac{\langle !\,A\rangle,\langle A \multimap (A \multimap B)\rangle \vdash B}{\langle A \multimap (A \multimap B)\rangle \vdash !\,A \multimap B}\,\multimap I}{\vdash (A \multimap (A \multimap B)) \multimap (!\,A \multimap B)}\,\multimap I}\,!\,E$$

From logic to language

The Story So Far

- Assumptions as stable truths vs finite resources
- Dropping copying/deletion splits logical constants
  - additive $\&, \oplus$: shared resources
  - multiplicative $\otimes$: no sharing
- No expressivity is lost: copying/deletion under ! control
- Universal logic: putting together the two styles of reasoning in one logic

The next step Going substructural: pushing it to the limit.

Grammar logics

(Linear logic) Why stop here?

Removing the remaining structural rules leads to logics of ordered, structured resources.
$\rightsquigarrow$ modelling grammatical composition.

- dropping Exchange: logic of strings, word order sensitivity
- dropping rebracketing: logic of phrases, constituent structure

J. Lambek 1958, 1961 (compare linear logic: Girard 1987)

- 1958: The mathematics of sentence structure (@ *JSTOR*)
- 1961: On the calculus of syntactic types

Joachim Lambek, December 5, 1922 – June 23, 2014

Festschrift *Categories and Types in Logic, Language, and Physics. Essays dedicated to Jim Lambek on the occasion of his 90th birthday*. Casadio, Coecke, Moortgat, and Scott (eds.) LNCS 8222, Springer, 2014.

Lambek's Syntactic Calculus

Formulas

$$A, B ::= p \mid A \otimes B \mid A\backslash B \mid B/A \mid \ldots$$

- $A \otimes B$: sequential composition, 'A and then B'
- $A\backslash B$: left imcompleteness, 'consume A to the left producing B'
- B/A: right incompleteness, 'consume A to the right producing B'

Two versions:

- **L**: assumptions as a list/string (non-commutative)
- **NL**: assumptions as a tree (non-associative)

L: strings of assumptions

$$\frac{}{A \vdash A}\ \mathsf{Ax}$$

$$\frac{A, \Gamma \vdash B}{\Gamma \vdash A\backslash B}\ \backslash I \qquad \frac{\Gamma \vdash A \quad \Delta \vdash A\backslash B}{\Gamma, \Delta \vdash B}\ \backslash E$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash B/A}\ /I \qquad \frac{\Gamma \vdash B/A \quad \Delta \vdash A}{\Gamma, \Delta \vdash B}\ /E$$

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\ \otimes I \qquad \frac{\Gamma \vdash A \otimes B \quad \Delta, A, B, \Delta' \vdash C}{\Delta, \Gamma, \Delta' \vdash C}\ \otimes E$$

Associativity The comma still hides a structural rule: Associativity.

NL: bracketed strings of assumptions

Comma as a 2-place structure-building operation $(-, -)$.

Structures $S := A \mid (S, S)$

Notation: $\Gamma[\Delta]$ structure Γ with substructure Δ (see $\otimes$ Elimination)

$$\frac{}{A \vdash A}\ \text{Ax}$$

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

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

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

L can now be obtained via an explicit Associativity structural rule:

$$\frac{\Gamma[(\Delta, (\Delta', \Delta''))] \vdash A}{\Gamma[((\Delta, \Delta'), \Delta'')] \vdash A}\ \text{(A)}$$

Proofs as programs: grammar logic

In the term language for Lambek's grammar logic, we have to make a distinction between left and right application, and between left and right abstraction.

Curry-Howard grammar logic $\simeq$ ordered linear lambda calculus.

$$\frac{}{x : A \vdash x : A}\ \mathsf{Ax}$$

$$\frac{x : A, \Gamma \vdash t : B}{\Gamma \vdash \lambda^{\backslash} x.t : A \backslash B}\ \backslash I \qquad \frac{\Gamma \vdash u : A \quad \Delta \vdash t : A \backslash B}{\Gamma, \Delta \vdash (u \triangleright t) : B}\ \backslash E$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda^{/} x.t : B / A}\ /I \qquad \frac{\Gamma \vdash t : B/A \quad \Delta \vdash u : A}{\Gamma, \Delta \vdash (t \triangleleft u) : B}\ /E$$

Ref Wansing 1992, Formulas-as-types for a hierarchy of sublogics of intuitionistic propositional logic

Shorthand format

In sequents

$$\Big(\Big)(x_1 : A_1, \ldots, x_n : A_n) \vdash t : B$$

(where the big parentheses stand for some bracketing of the terminal assumptions $x_i : A_i$) we can omit the types on the A_i and the term on B.

Example substituting lexical constants for the free assumption variables x_i.

$$\dfrac{\dfrac{\text{the}}{np/n} \qquad \dfrac{\dfrac{\text{book}}{n} \qquad \dfrac{\dfrac{\text{that}}{(n\backslash n)/(s/np)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\text{carroll}}{np} \qquad \dfrac{\dfrac{\text{wrote}}{(np\backslash s)/np} \qquad [\text{␣} \vdash np]^1}{\text{wrote} \cdot \text{␣} \vdash np\backslash s}\,[/E]}{\text{carroll} \cdot (\text{wrote} \cdot \text{␣}) \vdash s}\,[\backslash E]}{(\text{carroll} \cdot \text{wrote}) \cdot \text{␣} \vdash s}\,A}{\text{carroll} \cdot \text{wrote} \vdash s/np}\,[/I]^1}{\text{that} \cdot (\text{carroll} \cdot \text{wrote}) \vdash n\backslash n}\,[/E]}{\text{book} \cdot (\text{that} \cdot (\text{carroll} \cdot \text{wrote})) \vdash n}\,[\backslash E]}{\text{the} \cdot (\text{book} \cdot (\text{that} \cdot (\text{carroll} \cdot \text{wrote}))) \vdash np}\,[/E]$$

To compress derivations horizontally, axioms $x : A \vdash x : A$ are written $\dfrac{x}{A}$

Proof term, yield

The term t in Curry-Howard correspondence with the above derivation is

$$\text{the} \cdot (\text{book} \cdot (\text{that} \cdot (\text{carroll} \cdot \text{wrote}))) \vdash t : np$$

$$t \quad = \quad \text{the} \lhd (\text{book} \rhd (\text{that} \lhd (\lambda^{/} x.(\text{carrol} \rhd (\text{wrote} \lhd x)))))$$

The yield function $[\![t]\!]$ computes the underlying string (bound variables are silent).

$$\begin{array}{rcll}
[\![a]\!] & = & a & \text{where } a \text{ is a constant or free variable} \\
[\![x]\!] & = & \epsilon & \text{for bound variables } x \\
[\![t \lhd u]\!] & = & [\![t]\!] \cdot [\![u]\!] & \\
[\![u \rhd t]\!] & = & [\![u]\!] \cdot [\![t]\!] & \\
[\![\lambda^{\backslash} x.t]\!] & = & [\![t]\!] & \\
[\![\lambda^{/} x.t]\!] & = & [\![t]\!] &
\end{array}$$

Structural control

We don't want to completely drop Exchange, Rebracketing, but bring them back in a controlled form. Compare: Contraction/Weakening in the intuitionistic/linear ! case.

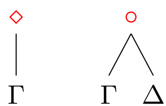
Control operators A pair of unary type-forming operations:

$$A, B \quad ::= \quad p \mid \Diamond A \mid \Box A \mid A \otimes B \mid A/B \mid B\backslash A$$

Structures Next to the 2-place structure-building operation $(\cdot,\cdot)$ now also 1-place $\langle\cdot\rangle$ as structural counterpart of $\Diamond$.

$$\Gamma, \Delta \quad ::= \quad A \mid \langle\Gamma\rangle \mid (\Gamma, \Delta)$$

Trees the above is a linear notation for the trees below:



Logical rules

$\diamondsuit, \Box$ form a residuated pair: $\diamondsuit\Box A \vdash A \vdash \Box\diamondsuit A$.

$$\frac{\Gamma \vdash \Box A}{\langle\Gamma\rangle \vdash A}\ \Box E \qquad \frac{\langle\Gamma\rangle \vdash A}{\Gamma \vdash \Box A}\ \Box I$$

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

Proofs as programs The term language is extended with (de)constructors for the $\diamondsuit, \Box$ Intro/Elim steps.

$$\frac{\Gamma \vdash t : \Box A}{\langle\Gamma\rangle \vdash {}^{\vee}t : A}\ \Box E \qquad \frac{\langle\Gamma\rangle \vdash t : A}{\Gamma \vdash {}^{\wedge}t : \Box A}\ \Box I$$

$$\frac{\Gamma \vdash t : A}{\langle\Gamma\rangle \vdash {}^{\cap}t : \diamondsuit A}\ \diamondsuit I \qquad \frac{\Delta \vdash u : \diamondsuit A \quad \Gamma[\langle x : A\rangle] \vdash t : B}{\Gamma[\Delta] \vdash t[{}^{\cup}u/x] : B}\ \diamondsuit E$$

Embeddings

Structural control can be realized in two ways:

- $\diamondsuit$ as licence: allow a structural rule that would not be available without $\diamondsuit$
- $\diamondsuit$ as obstacle: block a structural option that would otherwise be possible

Structural control $\mathcal{L}$, $\mathcal{L}'$ two logics that differ w.r.t. structural option P: $\mathcal{L}' = \mathcal{L} + P$. We express $\mathcal{L}$ in $\mathcal{L}'$ or vv, via translations $\cdot^\flat$, $\cdot^\sharp$:

(obstacle) $A \vdash B$ is provable in $\mathcal{L}_{/,\otimes,\backslash}$ iff $A^\flat \vdash B^\flat$ is provable in $\mathcal{L}'_{\diamondsuit,\Box,/,\otimes,\backslash}$

the translation blocks applications of P

(licence) $A \vdash B$ is provable in $\mathcal{L}'_{/,\otimes,\backslash}$ iff $A^\sharp \vdash B^\sharp$ is provable in $\mathcal{L}_{\diamondsuit,\Box,/,\otimes,\backslash} + P_\diamond$

$P_\diamond$: image of P under $\cdot^\sharp$; allowing a 'modal' version of P

(Kurtonina & MM 1996, Structural control)

Illustration: trees versus strings

Let $\mathcal{L}$ be the base logic **NL** (assumptions form a tree, no structural rules at all) and $\mathcal{L}'$ the string logic **L**, with associative tensor.

Translations One schema fits both $\cdot^\flat$ and $\cdot^\sharp$

$$\begin{array}{rcl} p^\natural & = & p \\ (A \otimes B)^\natural & = & \Diamond(A^\natural \otimes B^\natural) \\ (A/B)^\natural & = & \Box A^\natural / B^\natural \\ (B\backslash A)^\natural & = & B^\natural \backslash \Box A^\natural \end{array}$$

$\Diamond$ as obstacle $\cdot^\flat$ expresses **NL** in **L**: $\Diamond$ blocks all possible applications of (A). Example: the $\cdot^\flat$ translation of (†) fails.

$$\dagger \quad (a\backslash b) \otimes (b\backslash c) \vdash a\backslash c$$

$\Diamond$ as licence With $\cdot^\sharp$, **L** can be expressed in **NL** + $\mathsf{A}_\diamond$. $\Diamond$ provides access to a modal version of (A). The $\cdot^\sharp$ translation of (†) is derivable.

$$\Diamond(\Diamond(A \otimes B) \otimes C) \dashv\vdash \Diamond(A \otimes \Diamond(B \otimes C)) \quad (\mathsf{A}_\diamond) = (\mathsf{A})^\sharp$$

A substructural landscape

logic	structure	associative	commutative
NL	tree	-	-
L	string	✓	-
NLP	mobile	-	✓
LP (=MILL)	multiset	✓	✓

MILL: the multiplicative fragment of ILL

Controlling resource management No expressivity is lost:

- controlled copying / deletion: !
- controlled reordering / restructuring: $\diamond$

Deconstructing the structural rules

The global versions of the structural options are too coarse $\rightsquigarrow$ decompose them in their elementary parts.

Postulates controlled ass, comm. Symmetries: left / right; up / down.

$$\Diamond C \otimes (B \otimes A) \to (\Diamond C \otimes B) \otimes A \qquad (A \otimes B) \otimes \Diamond C \to A \otimes (B \otimes \Diamond C)$$

$$\Diamond C \otimes (B \otimes A) \to B \otimes (\Diamond C \otimes A) \qquad (A \otimes B) \otimes \Diamond C \to (A \otimes \Diamond C) \otimes B$$

$$(\Diamond C \otimes B) \otimes A \to \Diamond C \otimes (B \otimes A) \qquad A \otimes (B \otimes \Diamond C) \to (A \otimes B) \otimes \Diamond C$$

$$B \otimes (\Diamond C \otimes A) \to \Diamond C \otimes (B \otimes A) \qquad (A \otimes \Diamond C) \otimes B \to (A \otimes B) \otimes \Diamond C$$

N.D. sequent rules each postulate $A \to B$ corresponds to a sequent rule

$$\frac{\Gamma[B'] \vdash C}{\Gamma[A'] \vdash C}$$

(A': A with formula variables/connectives replaced by their structural counterpart)

Action at a distance: right branches

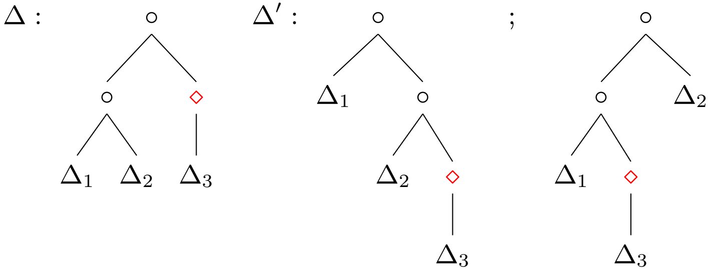
Postulates

$$(A \otimes B) \otimes \Diamond C \rightarrow A \otimes (B \otimes \Diamond C) \quad (P1)$$

$$(A \otimes B) \otimes \Diamond C \rightarrow (A \otimes \Diamond C) \otimes B \quad (P2)$$

N.D. rules

$$\frac{\Gamma[\Delta'] \vdash A}{\Gamma[\Delta] \vdash A}\ P1; P2$$



Action at a distance: left branches

Postulates

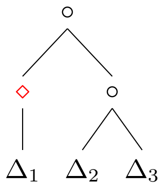
$$\Diamond C \otimes (B \otimes A) \to (\Diamond C \otimes B) \otimes A \quad (P1')$$

$$\Diamond C \otimes (B \otimes A) \to B \otimes (\Diamond C \otimes A) \quad (P2')$$

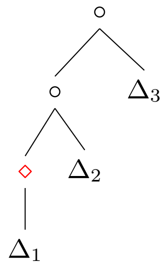
N.D. rules

$$\frac{\Gamma[\Delta'] \vdash A}{\Gamma[\Delta] \vdash A}\; P1'; P2'$$

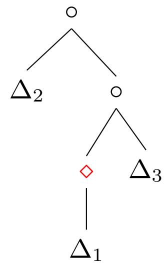
Δ :



Δ' :



;



Extraction: structurally controlled

Synthetic connectives, derived inference rules.

$A \upharpoonright B := A/\Diamond\Box B$

$$\frac{\Gamma[(\Delta, B)] \vdash A}{\Gamma[\Delta] \vdash A \upharpoonright B}$$

$$\frac{\Gamma[(\Delta, y : B)] \vdash t : A}{\Gamma[\Delta] \vdash \lambda^{/} x.t[{}^{\vee\cup}x/y] : A \upharpoonright B}$$

$B \upharpoonleft A := \Diamond\Box B\backslash A$

$$\frac{\Gamma[(B, \Delta)] \vdash A}{\Gamma[\Delta] \vdash B \upharpoonleft A}$$

$$\frac{\Gamma[(y : B, \Delta)] \vdash t : A}{\Gamma[\Delta] \vdash \lambda^{\backslash} x.t[{}^{\vee\cup}x/y] : B \upharpoonleft A}$$

Up / down symmetry: infixation

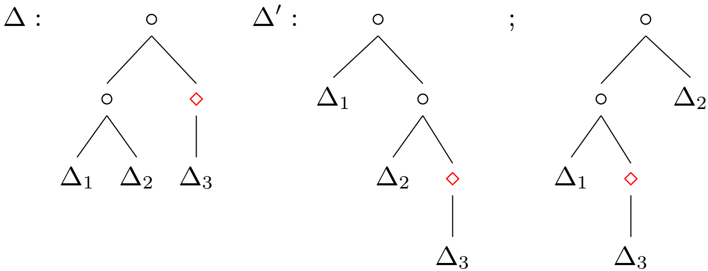
Postulates

$$A \otimes (B \otimes \Diamond C) \rightarrow (A \otimes B) \otimes \Diamond C \quad (P1)$$

$$(A \otimes \Diamond C) \otimes B \rightarrow (A \otimes B) \otimes \Diamond C \quad (P2)$$

N.D. rules

$$\frac{\Gamma[\Delta] \vdash A}{\Gamma[\Delta'] \vdash A} \ P1; P2$$



Exercise Add the left/right symmetric version.

Left, right infixation

Derived inference rules for defined infixation operations.

Left/right symmetry $B \downharpoonleft A := \Diamond\Box(B\backslash A)$; $A \downharpoonright B := \Diamond\Box(A/B)$

$$\frac{\Gamma \vdash B \downharpoonleft A \quad \Delta[\Delta'] \vdash B}{\Delta[(\Delta', \Gamma)] \vdash A} \qquad \frac{\Gamma \vdash A \downharpoonright B \quad \Delta[\Delta'] \vdash B}{\Delta[(\Gamma, \Delta')] \vdash A}$$

Proofs as programs The rules with their term annotation.

$$\frac{\Gamma \vdash t : A \downharpoonright B \quad \Delta[\Delta'] \vdash u : B}{\Delta[(\Gamma, \Delta')] \vdash ({}^{\vee\cup}t) \triangleleft u : A}$$

$$\frac{\Gamma \vdash t : B \downharpoonleft A \quad \Delta[\Delta'] \vdash u : B}{\Delta[(\Delta', \Gamma)] \vdash u \triangleright ({}^{\vee\cup}t) : A}$$

Lab session: extraction NL

We write s for the NL verb-final clause. Some lexical entries, with IL lexical meaning recipes.

footman, result	lakei, resultaat	::	n	
(dets)	de, het, dit	::	np/n	
teases	plaagt	::	$np\backslash(np\backslash s)$	
counts	rekent	::	$pp\backslash(np\backslash s)$	
on	op	::	pp/np	
not	niet	::	$(np\backslash s)/(np\backslash s)$	$\lambda x \lambda y.\neg(x\ y)$
(relpro)	die, dat	::	$(n\backslash n)/(np \upharpoonright s)$	$\lambda x \lambda y \lambda z.((x\ z) \wedge (y\ z))$

Add entries for er (pro neuter) and waar (relpro neuter). Give derivations of the test phrases below, and their interpretation.

ik weet dat	Alice de lakei plaagt $\vdash s$	
ik weet dat	Alice op dit resultaat rekent $\vdash s$	
ik weet dat	Alice er op rekent $\vdash s$	
	de lakei die Alice plaagt $\vdash np$	two derivations!
	het resultaat waar Alice op rekent $\vdash np$	
	het resultaat dat Alice op rekent $\vdash np$	underivable!
ik weet dat	Alice er niet op rekent $\vdash s$	

Solutions

First try for er, base-generated to the left of the preposition. The i_{1750} is a mock pronoun meaning.

$$\text{er} \quad :: \quad pp/(pp/np) \quad \lambda x.(x\ i_{1750})$$

Now waar is simply the wh form corresponding to er ($rpro$ abbrev $pp/(pp/np)$).

$$\text{waar} \quad :: \quad (n\backslash n)/(rpro \upharpoonright s) \quad \lambda x \lambda y \lambda z.((y\ z) \wedge (x\ \lambda w.(w\ z)))$$

The entry for er has to be refined: it is a verb phrase initial clitic that can get separated from its preposition as in er niet op rekent (iv abbrev $np\backslash s$)

$$\text{er} \quad :: \quad iv/(rpro\backslash iv) \quad \lambda x \lambda y.((x\ \lambda z.(z\ i_{1750}))\ y)$$

After substituting lexical meaning recipes (and term reduction), the last example is associated with the desired meaning recipe

$$\neg(\textsc{count}\ (\textsc{on}\ i_{1750})\ \textsc{alice})$$

□