Compositional Models of Vector-based Semantics: From Theory to Tractable Implementation

Day 3. Refining the type logic

Gijs Wijnholds & Michael Moortgat

ESSLLI 2022

Plan

- The Return of the Lambdas

  λ encodings of string semantics and vector-based interpretation

- Diamonds forever

  enhancing type logic with modalities $\{\Diamond_j, \Box_j\}_{j \in J}$

  - modalities for structural control
  - modalities encoding dependency info

- Discussion: rethinking constituency

Paving the way for implementation and empirical evaluation of $\leadsto$ Day 4, Day 5.

Preamble. The Return of the Lambdas

Recall the unpacking of lexical semantics for the e, t based interpretation:

We write AB for $A \multimap B$

WORD	SYN TYPE	$\lceil \cdot \rceil$	SEM TYPE
paper	n	PAPER	et
that	$(n\backslash n)/(s/np)$	$\lambda x \lambda y \lambda^{!} z.((y\ z) \wedge (x\ z))$	$(et)(et)!e\,t$
Bob	np	BOB	e
rejected	$(np\backslash s)/np$	REJECTED	eet

$$\begin{aligned} \lceil M \rceil &= ((\lceil \text{that} \rceil\ \lambda x.((\lceil \text{rejected} \rceil\ x)\ \lceil \text{Bob} \rceil))\ \lceil \text{paper} \rceil) \\ &= \lambda x.((\text{PAPER}\ x) \wedge ((\text{REJECTED}\ x)\ \text{BOB})) \end{aligned}$$

ACG, Lambda Grammars: λ calculus encoding of other interpretations:

- string semantics [de Groote and Pogodalla, 2004]
- vector semantics [Sadrzadeh and Muskens, 2018]

Lambdas for String Semantics

- Target signature: string as function type $* \multimap *$ (abbrev σ). Concatenation: composition; empty string: identity function.

$$+ := \lambda sri.s(r(i)) \qquad \epsilon := \lambda i.i$$

- type homomorphism: $\lceil A \rceil = \sigma$, for source atoms A.
- translating the constants:

WORD	SYN TYPE	$\lceil\cdot\rceil$	SEM TYPE
paper	n	$\texttt{paper}$	σ
that	$(n\backslash n)/(s/np)$	$\lambda Zs.s + \texttt{that} + (Z\ \epsilon)$	$(\sigma\sigma)\sigma\sigma$
Bob	np	$\texttt{Bob}$	σ
rejected	$(np\backslash s)/np$	$\lambda sr.r + \texttt{rejected} + s$	$\sigma\sigma\sigma$

$$\begin{aligned} \lceil M \rceil &= ((\lceil \text{that} \rceil\ \lambda x.((\lceil \text{rejected} \rceil\ x)\ \lceil \text{Bob} \rceil))\ \lceil \text{paper} \rceil) : \sigma \\ &= \texttt{paper} + \texttt{that} + \texttt{Bob} + \texttt{rejected} \end{aligned}$$

ACG refined Chomsky hierarchy via type homomorphisms of growing complexity.

Lambdas for Vector Semantics

- Target signature. I: index set; R: reals $\mathbb{R}$. Vectors: IR, rank n tensors $I^n R$ (curried). $M := I^2 R$, $C := I^3 R$, etc. Defined operations e.g.

$$
\begin{array}{lll}
\odot & := & \lambda vui.v_i \cdot u_i \qquad VVV \\
\times_1 & := & \lambda mvi.\sum_j m_{ij} \cdot v_j \qquad MVV \\
\times_2 & := & \lambda cvij.\sum_k c_{ijk} \cdot v_k \qquad CVM
\end{array}
$$

- type homomorphism: $\lceil A \rceil = V$ for source atoms A.
- translating the constants: $\lceil \text{paper} \rceil = \texttt{paper}$, $\lceil \text{Bob} \rceil = \texttt{Bob}$ (vectors),

WORD	SYN TYPE	$\lceil\cdot\rceil$	SEM TYPE
rejected	$(np\backslash s)/np$	$\lambda uv.(\texttt{rejected} \times_2 u) \times_1 v$	VVV
that	$(n\backslash n)/(s/np)$	$\lambda Zv. \ldots$	$(VV)VV$

$$
\begin{array}{lll}
\lceil M \rceil & = & ((\lceil \text{that} \rceil\ \lambda x.((\lceil \text{rejected} \rceil\ x)\ \lceil \text{Bob} \rceil))\ \lceil \text{paper} \rceil) : V \\
 & ? & \texttt{paper} \odot \tau_i \sum_j (\texttt{rejected}_{ijk} \cdot \texttt{Bob}_j)
\end{array}
$$

Proper recipe for 'that': exercise ...

Interpretation homomorphism: summary

Types source & target signatures have their own atomic types, operations

- each (source/target) type is associated with a denotation domain/semantic space
- $\lceil \cdot \rceil$ maps source atoms to target types, possibly non-atomic

Terms source & target signatures come with their own typed constants

- $\lceil \cdot \rceil$ maps source constants to target terms, possibly complex
- $\lceil \cdot \rceil$ respects types: source constant c^A is mapped to target term $M^{\lceil A \rceil}$

Opportunities? lifting the dimensionality curse, e.g. intersective adjectives:

WORD	SYN TYPE	$\lceil \cdot \rceil$	SEM TYPE
car	n	$\texttt{car}$	V
red	n/n	$\lambda v.\, \texttt{red}^V \odot v$	VV

Enhancing type logics with modalities

Why?

The Need for Control

- languages exhibit phenomena that seem to require some form of

  reordering, restructuring, copying

- global structural options are problematic

  too little (undergeneration), too much (overgeneration)

- extended type language with operations for structural control:
  - licensing structural reasoning that is lacking by default
  - blocking structural reasoning that would otherwise be available

Source reference [Kurtonina and Moortgat, 1997]

Associativity: too little

$$\frac{\dfrac{\text{what}}{np/(s/np)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\text{Maisie}}{np} \qquad \dfrac{\dfrac{\text{knew}}{(np\backslash s)/np} \qquad np \vdash np^1}{\text{knew} \cdot np \vdash np\backslash s}\,/E}{\text{Maisie} \cdot (\text{knew} \cdot np) \vdash s}\,\backslash E}{(\text{Maisie} \cdot \text{knew}) \cdot np \vdash s}\,\mathsf{A}^r}{\text{Maisie} \cdot \text{knew} \vdash s/np}\,/I^1}{\text{what} \cdot (\text{Maisie} \cdot \text{knew}) \vdash np}\,/E$$

Compare

- what Maisie knew ␣

  position of the hypothesis reachable thanks to A^r (right rotation) ☺

- what Maisie knew ␣ about her parents

  too little: A^r doesn't give access to an internal gap ☹

Associativity: too much

Locality constraint himself :: $((np\backslash s)/np)\backslash(np^{\text{♂}}\backslash s)$, herself :: $((np\backslash s)/np)\backslash(np^{\text{♀}}\backslash s)$

Alice hurt herself / *himself
Alice thinks Bob hurt *herself / himself

No way of distinguishing simple transitive verb $(np\backslash s)/np \neq$ string reducing to $(np\backslash s)/np$

$$\dfrac{\dfrac{\text{Alice}}{np^{\text{♀}}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{thinks}}{(np\backslash s)/s} \qquad \dfrac{\dfrac{\text{Bob}}{np} \qquad \dfrac{\dfrac{\text{hurt}}{(np\backslash s)/np} \qquad np \vdash np^1}{\text{hurt} \cdot np \vdash np\backslash s}\,/E}{\text{Bob} \cdot (\text{hurt} \cdot np) \vdash s}\,\backslash E}{\text{thinks} \cdot (\text{Bob} \cdot (\text{hurt} \cdot np)) \vdash np\backslash s}\,/E}{\text{thinks} \cdot ((\text{Bob} \cdot \text{hurt}) \cdot np) \vdash np\backslash s}\,\mathsf{A}^r}{(\text{thinks} \cdot (\text{Bob} \cdot \text{hurt})) \cdot np \vdash np\backslash s}\,\mathsf{A}^r}{\text{thinks} \cdot (\text{Bob} \cdot \text{hurt}) \vdash (np\backslash s)/np}\,/I^1 \qquad \dfrac{\text{herself}}{((np\backslash s)/np)\backslash(np^{\text{♀}}\backslash s)}}{(\text{thinks} \cdot (\text{Bob} \cdot \text{hurt})) \cdot \text{herself} \vdash np^{\text{♀}}\backslash s}\,\backslash E}{\text{Alice} \cdot ((\text{thinks} \cdot (\text{Bob} \cdot \text{hurt})) \cdot \text{herself}) \vdash s}\,\backslash E$$

Modalities for structural control

- The type language is extended with a pair of unary connectives: $\Diamond, \Box$ satisfying

$$\frac{\Diamond A \longrightarrow B}{A \longrightarrow \Box B}$$

- Logic: $\Diamond, \Box$ form a residuated pair. One easily shows

compositions: $\Diamond\Box A \longrightarrow A$ (interior) $\quad A \longrightarrow \Box\Diamond A$ (closure)

monotonicity: from $A \longrightarrow B$ infer $\Diamond A \longrightarrow \Diamond B$, $\Box A \longrightarrow \Box B$

- Structure: global rules $\rightsquigarrow$ $\Diamond$ controlled restricted versions, e.g.

$$A^{\diamond}: \quad (A \bullet B) \bullet \Diamond C \longrightarrow A \bullet (B \bullet \Diamond C)$$
$$C^{\diamond}: \quad (A \bullet B) \bullet \Diamond C \longrightarrow (A \bullet \Diamond C) \bullet B$$

Multimodal generalization families $\{\Diamond_i, \Box_i\}_{i \in I}$ for particular structural choices

Control operators: N.D. rules

Structures $\Gamma, \Delta ::= A \mid \langle\Gamma\rangle \mid \Gamma \cdot \Delta$

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

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

Shorthand $(\Diamond E')$ if left premise of $(\Diamond E)$ is an axiom:

$$\frac{\Gamma[\langle A\rangle] \vdash B}{\Gamma[\Diamond A] \vdash B}\ \Diamond E'$$

Control operators: terms

Terms: $M, N ::= x \mid \lambda x.M \mid M N \mid \triangledown M \mid \vartriangle M \mid \blacktriangledown M \mid \blacktriangle M$

$$\frac{\langle \Gamma \rangle \vdash M : A}{\Gamma \vdash \blacktriangle M : \Box A} \; \Box I \qquad \frac{\Gamma \vdash M : \Box A}{\langle \Gamma \rangle \vdash \blacktriangledown M : A} \; \Box E$$

$$\frac{\Gamma \vdash M : A}{\langle \Gamma \rangle \vdash \vartriangle M : \Diamond A} \; \Diamond I \qquad \frac{\Delta \vdash M : \Diamond A \quad \Gamma[\langle x : A \rangle] \vdash N : B}{\Gamma[\Delta] \vdash N[\triangledown M / x] : B} \; \Diamond E$$

Proof normalization: $\blacktriangledown \blacktriangle M = M$, $\blacktriangle \blacktriangledown M = M$; $\vartriangle \triangledown M = M$, $\triangledown \vartriangle M = M$

Interpretation We assume the role of the control modalities is restricted to the syntax:

$$\lceil \Diamond A \rceil = \lceil \Box A \rceil = \lceil A \rceil$$

But see [Correia et al., 2020] for a quantum interpretation of modalities for linguistic applications.

Controlled extraction: too little $\rightsquigarrow$ just fine

$\Diamond\Box np$: 'moveable' np; key-and-lock: contract $\Diamond\Box np$ to np, once in place.

$$\dfrac{\begin{array}{c}\text{found}\\ (np\backslash s)/np\end{array} \qquad \dfrac{\Box np \vdash \Box np}{\langle \Box np \rangle \vdash np}\ \Box E}{\text{found} \cdot \langle \Box np \rangle \vdash np\backslash s}\ /E \qquad \begin{array}{c}\text{there}\\ (np\backslash s)\backslash(np\backslash s)\end{array}$$

$$\dfrac{\text{found} \cdot \langle \Box np \rangle \vdash np\backslash s \qquad \begin{array}{c}\text{there}\\ (np\backslash s)\backslash(np\backslash s)\end{array}}{(\text{found} \cdot \langle \Box np \rangle) \cdot \text{there} \vdash np\backslash s}\ \backslash E$$

$$\dfrac{\begin{array}{c}\text{Alice}\\ np\end{array} \qquad (\text{found} \cdot \langle \Box np \rangle) \cdot \text{there} \vdash np\backslash s}{\dfrac{\dfrac{\dfrac{\dfrac{\text{Alice} \cdot ((\text{found} \cdot \color{red}{\langle \Box np \rangle}) \cdot \text{there}) \vdash s}{\text{Alice} \cdot ((\text{found} \cdot \text{there}) \cdot \color{red}{\langle \Box np \rangle}) \vdash s}\ \mathsf{C}^r_\Diamond}{(\text{Alice} \cdot (\text{found} \cdot \text{there})) \cdot \color{red}{\langle \Box np \rangle} \vdash s}\ \mathsf{A}^r_\Diamond}{(\text{Alice} \cdot (\text{found} \cdot \text{there})) \cdot \Diamond\Box np \vdash s}\ \Diamond E'}{\text{Alice} \cdot (\text{found} \cdot \text{there}) \vdash s/\Diamond\Box np}\ /I}\ \backslash E$$

$$\dfrac{\begin{array}{c}\text{what}\\ np/(s/\Diamond\Box np)\end{array} \qquad \text{Alice} \cdot (\text{found} \cdot \text{there}) \vdash s/\Diamond\Box np}{\text{what} \cdot (\text{Alice} \cdot (\text{found} \cdot \text{there})) \vdash np}\ /E$$

$\mathsf{A}^r_\Diamond$: controlled Associativity, $(A \bullet B) \bullet \Diamond C \longrightarrow A \bullet (B \bullet \Diamond C)$

$\mathsf{C}^r_\Diamond$: controlled Commutativity, $(A \bullet B) \bullet \Diamond C \longrightarrow (A \bullet \Diamond C) \bullet B$

Your turn

Show that ($xright$) (with Δ non-empty) is a derived inference rule.

writing $!_x\, A$ for $\Diamond\Box A$

$$\frac{\Gamma[\Delta \cdot A] \vdash B}{\Gamma[\Delta] \vdash B/\,!_x\, A}\; xright$$

($xright$) telescopes a sequence of $\mathsf{A}^r_\Diamond$, $\mathsf{C}^r_\Diamond$ structural steps flanked by / Intro and $\Diamond, \Box$ Elim into a one-step inference rule:

$$\dfrac{\dfrac{\text{Alice}}{np} \quad \dfrac{\dfrac{\dfrac{\dfrac{\text{found}}{(np\backslash s)/np} \quad \dfrac{}{np \vdash np}}{\text{found} \cdot np \vdash np\backslash s}\,/E \quad \dfrac{\text{there}}{(np\backslash s)\backslash(np\backslash s)}}{(\text{found} \cdot np) \cdot \text{there} \vdash np\backslash s}\,\backslash E}{}}{\dfrac{\text{Alice} \cdot ((\text{found} \cdot np) \cdot \text{there}) \vdash s}{\text{Alice} \cdot (\text{found} \cdot \text{there}) \vdash s/\,!_x\, np}\, xright}\,\backslash E$$

Mind the gaps (Expert level)

Your turn compute the vector-based interpretation for some more challenging examples of parasitic gaps in [Moortgat et al., 2020]

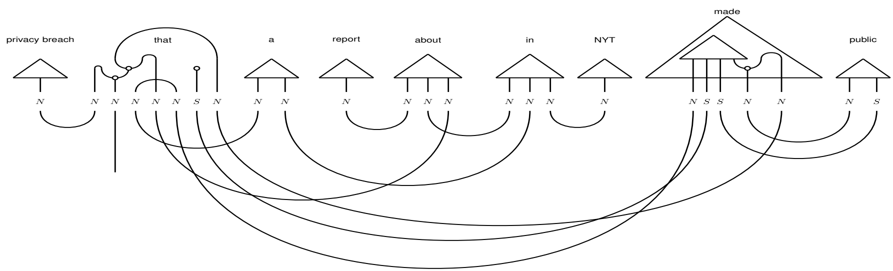
- papers that Bob rejected ␣ (immediately) without reading ␣$_p$ (carefully)

  non-peripheral gaps

- security breach that a report about ␣$_p$ in the NYT made ␣ public

  co-argument gaps

Diagrammatically



Blocking structural rules

Compare the following $iv := np\backslash s$

Napoleon slept in this bed	in :: $(iv\backslash iv)/np$
N slept during the speech	during :: $(iv\backslash iv)/np$
the bed which N slept in	which :: $(n\backslash n)/(s/\Diamond\Box np)$
*the speech which N slept during	

In general, English allows preposition stranding, which is derivable with the controlled asso/commu rules. But some modifiers behave as islands, inaccessible for extraction.

$\Diamond$ as an obstacle a modified type assignment imposes the desired island constraint:

during :: $(\Box(iv\backslash iv))/np$ Morrill 1994, url

- during first has to compose with its np object
- the result type $\Box(iv\backslash iv)$ is *locked* by $\Box$
- $\Diamond$ *unlocks* $\Box(iv\backslash iv)$, thus sealing off during np as an island

In what follows, we refine the idea of modalities projecting locality domains to take into account dependency info.

Encoding dependency structure

Heads vs dependents

Dependency roles articulate the linguistic material on the basis of two oppositions:

- head - complement relations
  - verbal domain: subj, (in)direct object, ...
  - nominal domain: prepositional object, ...
- adjunct - head relations
  - verbal domain: (time, manner, ...) adverbial
  - nominal domain: adjectival, numeral, determiner, ...

Compare: fa-structure: function vs argument

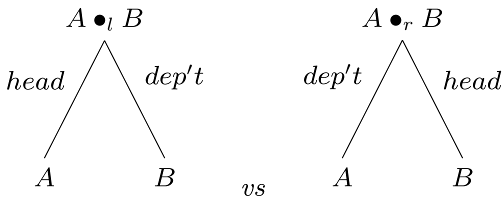
Orthogonality The fa and the dependency articulation are in general not aligned. This asks for a multidimensional type logic.

E.g. Determiner. Semantically, characteristic function of $([\![N]\!], [\![VP]\!])$ relation; morphologically, dependent on head noun.

A first step: bimodal syntactic calculus

cf MM and Morrill 1991, Heads and phrases. Type calculus for dependency and constituent structure. Ms.

Product $\bullet$ is split in a left-headed $\bullet_l$ and a right-headed $\bullet_r$ version:



$$\text{RES} \qquad \begin{array}{ccccc} A \longrightarrow C/_l B & \text{iff} & A \bullet_l B \longrightarrow C & \text{iff} & B \longrightarrow A\backslash_l C \\ A \longrightarrow C/_r B & \text{iff} & A \bullet_r B \longrightarrow C & \text{iff} & B \longrightarrow A\backslash_r C \end{array}$$

head functor: $C/_l B, \quad A\backslash_r C$

dependent functor: $C/_r B, \quad A\backslash_l C$

Concrete interpretation modelling intonation, prosodic prominence, [Hendriks, 1997]

Deconstructing the headed product

Define $\bullet_l$, $\bullet_r$ as compositions of regular $\bullet$ and modal marking of the dependent:

$$A \bullet_l B := A \bullet \Diamond B \qquad A \bullet_r B := \Diamond A \bullet B$$

Residuation: translation of the slashes

recall: $\Diamond A \longrightarrow B$ iff $A \longrightarrow \Box B$

$$\dfrac{A \longrightarrow C/_l B}{\dfrac{A \bullet_l B \longrightarrow C}{B \longrightarrow A\backslash_l C}} \quad \rightsquigarrow \quad \dfrac{A \longrightarrow C/\Diamond B}{\dfrac{A \bullet \Diamond B \longrightarrow C}{\dfrac{\Diamond B \longrightarrow A\backslash C}{B \longrightarrow \Box(A\backslash C)}}}$$

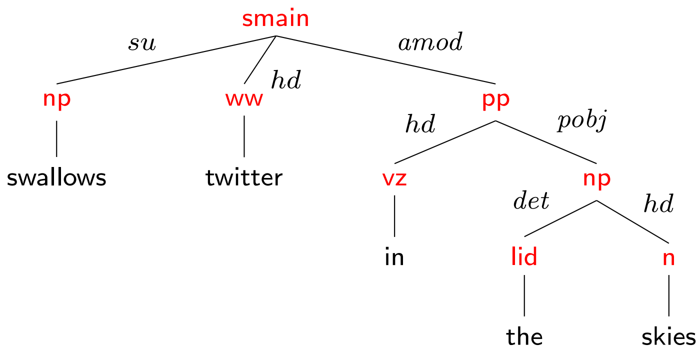
$$\dfrac{A \longrightarrow C/_r B}{\dfrac{A \bullet_r B \longrightarrow C}{B \longrightarrow A\backslash_r C}} \quad \rightsquigarrow \quad \dfrac{A \longrightarrow \Box(C/B)}{\dfrac{\Diamond A \longrightarrow C/B}{\dfrac{\Diamond A \bullet B \longrightarrow C}{B \longrightarrow \Diamond A\backslash C}}}$$

Multimodal generalization families $\{\Diamond_d, \Box_d\}_{d \in DepLabel}$

- $\Diamond_d A\backslash C$, $C/\Diamond_d B$ head functor assigning dependency role d to its complement
- $\Box_d(A\backslash C)$, $\Box_d(C/B)$ dependent functor projecting adjunct role d

Extracting types from structured data

Dutch treebank LASSY Annotation DAGs, nodes: synt categories, edges: dependency relations. Re-entrancy: higher-order types.



Extracted types:

swallows : np twitter : $\Diamond_{su} np \backslash s$ in : $\Box_{amod}(s \backslash s) / \Diamond_{pobj} np$ the : $\Box_{det}(np/n)$ skies : n

Dependency structure

Derivation, N.D. style:

$$\dfrac{\text{the}}{\Box_{det}(np/n)} \qquad \dfrac{\Box_{det}(np/n)}{\langle\text{the}\rangle^{det} \vdash np/n}\ \Box E \qquad \dfrac{\text{skies}}{n}$$

$$\dfrac{\langle\text{the}\rangle^{det} \vdash np/n \qquad n}{\langle\text{the}\rangle^{det} \cdot \text{skies} \vdash np}\ /E \qquad \dfrac{\langle\text{the}\rangle^{det} \cdot \text{skies} \vdash np}{\langle\langle\text{the}\rangle^{det} \cdot \text{skies}\rangle^{pobj} \vdash \Diamond_{pobj} np}\ \Diamond I$$

$$\dfrac{\text{in}}{\Box_{amod}(s\backslash s)/\Diamond_{pobj} np} \qquad \dfrac{\Box_{amod}(s\backslash s)/\Diamond_{pobj} np \qquad \langle\langle\text{the}\rangle^{det} \cdot \text{skies}\rangle^{pobj} \vdash \Diamond_{pobj} np}{\text{in} \cdot \langle\langle\text{the}\rangle^{det} \cdot \text{skies}\rangle^{pobj} \vdash \Box_{amod}(s\backslash s)}\ /E$$

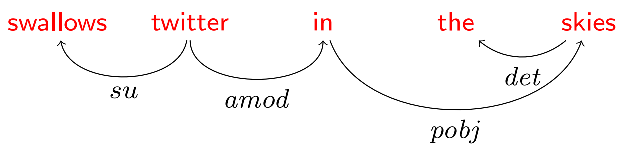
$$\dfrac{\text{in} \cdot \langle\langle\text{the}\rangle^{det} \cdot \text{skies}\rangle^{pobj} \vdash \Box_{amod}(s\backslash s)}{\langle\text{in} \cdot \langle\langle\text{the}\rangle^{det} \cdot \text{skies}\rangle^{pobj}\rangle^{amod} \vdash s\backslash s}\ \Box E$$

$$\dfrac{\dfrac{\text{swallows}}{np}}{\langle\text{swallows}\rangle^{su} \vdash \Diamond_{su} np}\ \Diamond I \qquad \dfrac{\text{twitter}}{\Diamond_{su} np\backslash s}$$

$$\dfrac{\langle\text{swallows}\rangle^{su} \vdash \Diamond_{su} np \qquad \Diamond_{su} np\backslash s}{\langle\text{swallows}\rangle^{su} \cdot \text{twitter} \vdash s}\ \backslash E$$

$$\dfrac{\langle\text{swallows}\rangle^{su} \cdot \text{twitter} \vdash s \qquad \langle\text{in} \cdot \langle\langle\text{the}\rangle^{det} \cdot \text{skies}\rangle^{pobj}\rangle^{amod} \vdash s\backslash s}{(\langle\text{swallows}\rangle^{su} \cdot \text{twitter}) \cdot \langle\text{in} \cdot \langle\langle\text{the}\rangle^{det} \cdot \text{skies}\rangle^{pobj}\rangle^{amod} \vdash s}\ \backslash E$$

Induced dependency structure:



$\rightsquigarrow$ within dependency domain, outgoing arcs from head to (head of) dependents

Tomorrow's episode: demo

Kokos Kogkalidis introduces you to his work on resources and tools for computational study of Dutch:

- extracting a type lexicon
- supertagging $\cong$ almost-parsing
- neural parsing

If you want to try things out, see the readme on

https://github.com/konstantinosKokos/lassy-tlg-extraction

for the extracted proofbank

https://github.com/konstantinosKokos/dynamic-proof-nets

for the parser

Discussion: rethinking constituency

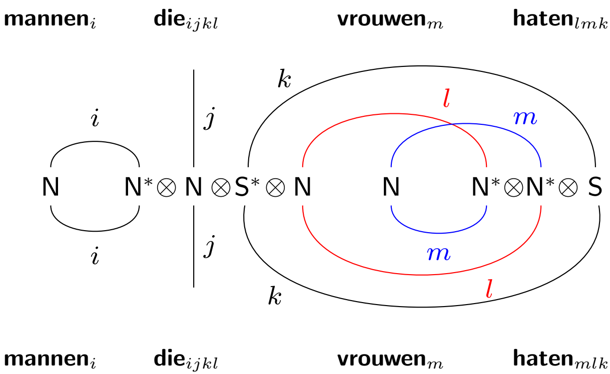
Case study: Extraction revisited

Recall Dutch left-branch extraction via controlled associativity, commutativity

$$\Diamond_x A \bullet (B \bullet C) \longrightarrow (\Diamond_x A \bullet B) \bullet C \qquad \Diamond_x A \bullet (B \bullet C) \longrightarrow B \bullet (\Diamond_x A \bullet C)$$

Relative pronoun die :: $(n\backslash n)/(!_x\, np\backslash s)$ $\qquad !_x\, A \triangleq \Diamond_x \Box_x A$

ambiguous between subj/obj relativization: $\qquad s$ subordinate clause, head-final



[Moortgat and Wijnholds, 2017]

Extraction revisited (cont'd)

Dependency refinement derivational ambiguity is traded in for lexical ambiguity, to be resolved in the supertagging phase.

$$\begin{array}{rcl} \text{haten} & :: & \diamondsuit_{obj}\, np\backslash(\diamondsuit_{subj}\, np\backslash s) \\ \text{die} & :: & \Box_{mod}(n\backslash n)/\diamondsuit_{body}(\diamondsuit_{subj}\, np\backslash s) \\ \text{die} & :: & \Box_{mod}(n\backslash n)/\diamondsuit_{body}(\diamondsuit_{obj}\, np\backslash s) \end{array}$$

Down the rabbit hole The above types restrict access to immediate dependents of the rel clause body. die :: $\Box_{mod}(n\backslash n)/\diamondsuit_{body}(!_x\diamondsuit_{subj|obj} np\backslash s)$ reaches more deeply embedded hypotheses.

The $xleft$ (derived) inference rule now has $\Gamma[\;\;]$ traversing unary+binary structure:

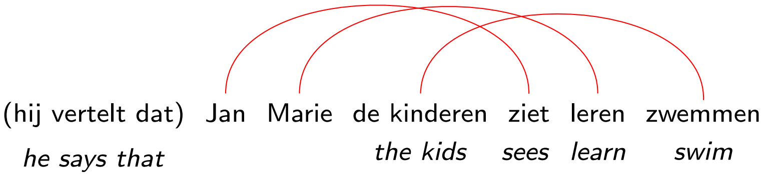
$$\frac{\Gamma[A \cdot \Delta] \vdash B}{\Gamma[\Delta] \vdash !_x\, A\backslash B}\; xleft$$

Categorically, this requires extra postulates allowing $\diamondsuit_x$ to commute with dependency modalities $\diamondsuit_d$ for (all|some) $d \in DepLabel$:

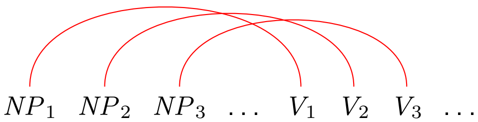
$$\diamondsuit_x\, A \bullet \diamondsuit_d\, B \longrightarrow \diamondsuit_d(\diamondsuit_x\, A \bullet B)$$

Case study: Dutch verb clusters

Crossing dependencies Huijbregts 1984, Shieber 1985



Pattern copy language w^2, i.e. mildly context-sensitive



Handling verb clusters: two type-logical options

- the Abstract Categorial Grammar method

  abstract syntax, divergent compositional translations:

  $\lceil\cdot\rceil^{string}$ string semantics

  $\lceil\cdot\rceil^{sem}$ meaning assembly

- logical $+$ structural reasoning

  derivations $\Gamma \vdash A$ with alternating logical and structural phases

  meaning assembly: Elim/Intro logical constants

  surface form: meaning preserving structural transformations

2-MCFG

iets willen zeggen	$(something\ want\ say)$	=want to say sth
haar iets laten zeggen	$(her\ something\ let\ say)$	=let her say sth

k-dimensional Multiple CFG: CFG with non-terminals ranging over string tuples.

- infinitival phrases $INFP$: string tuples (*rest*, *verb(cluster)*)
- verb raising triggers IVR_i: head-adjunction [Pollard, 1984]

$$
\begin{array}{llcll}
f_0 & INFP(\epsilon, x) & \leftarrow & INF_0(x) & \\
f_1 & INFP(x, y) & \leftarrow & NP(x)\ INF_1(y) & \\
 & & & & \\
g_0 & INFP(y, x \cdot z) & \leftarrow & INFP(y, z)\ IVR_0(x) & \\
g_1 & INFP(y \cdot z, x \cdot w) & \leftarrow & NP(y)\ INFP(z, w)\ IVR_1(x) & \\
 & & & & \\
d_0 & INF_0(\text{vertrekken}) & \leftarrow & & leave \\
d_1 & INF_1(\text{zeggen}) & \leftarrow & & say \\
d_2 & IVR_0(\text{willen}) & \leftarrow & & want \\
d_3 & IVR_1(\text{laten}) & \leftarrow & & let
\end{array}
$$

VR triggers: modal/temporal auxiliaries (g_0); verbs of perception, causatives (g_1)

2-MCFG (cont'd)

Some VR triggers select an infinitival complement with the particle 'te'. An alternative construction is extraposition, which has the verbal complement intact at the end of the clause. Interestingly, some verbs allow the two realizations.

iets proberen te zeggen	(*something try to say*)	VR
proberen iets te zeggen	(*try something to say*)	extra

$$
\begin{array}{lrcll}
h_0 & INFP(y, x \cdot z) & \leftarrow & TIP(y,z)\ IVR_2(x) & \\
h_1 & TIP(x, \text{te}\cdot y) & \leftarrow & INFP(x,y) & \\
d_4 & IVR_2(\text{proberen}) & \leftarrow & & \textit{try} \\
x_0 & INFP_x(\epsilon, x) & \leftarrow & INF_2(x) & \\
x_1 & INFP_x(y, x\cdot z) & \leftarrow & INFP_x(y,z)\ IVR_0(x) & \\
x_2 & INFP_x(x\cdot z, y\cdot w) & \leftarrow & OBJ(x)\ INFP_x(z,w)\ IVR_1(y) & \\
d_7 & INF_2(\text{proberen}) & \leftarrow & & \textit{try} \\
r_0 & S(\text{hij}\cdot\text{zal}\cdot x\cdot y) & \leftarrow & INFP(x,y) & \\
x_3 & S(\text{hij}\cdot\text{zal}\cdot x\cdot y\cdot z\cdot w) & \leftarrow & INFP_x(x,y)\ TIP(z,w) &
\end{array}
$$

Friday you'll see how BERT performs on these patterns ...

Verb clusters: the ACG method

The ACG method is easily adapted to our **NL** source: words as abstract constants.

Simple combinatorics, inflated type homomorphism String semantics: higher-order modelling of tuples

$$\lceil INFP \rceil = (\sigma \multimap \sigma \multimap \sigma) \multimap \sigma \qquad \triangleq \sigma^{(2)}$$

ACG method (cont'd)

Abstract syntax The syntax types don't yield the surface string, but the closest you can get using logical rules only.

$$\dfrac{\dfrac{\text{haar}}{NP} \quad \dfrac{\dfrac{\dfrac{\text{iets}}{NP} \quad \dfrac{\text{zeggen}}{NP\backslash INFP}}{\text{iets}\cdot\text{zeggen} \vdash INFP}\backslash E \quad \dfrac{\text{laten}}{INFP\backslash(NP\backslash INFP)}}{(\text{iets}\cdot\text{zeggen})\cdot\text{laten} \vdash NP\backslash INFP}\backslash E}{\text{haar}\cdot((\text{iets}\cdot\text{zeggen})\cdot\text{laten}) \vdash INFP}\backslash E \quad \dfrac{\text{willen}}{INFP\backslash INFP}$$

$$\dfrac{\qquad}{\dagger \qquad (\text{haar}\cdot((\text{iets}\cdot\text{zeggen})\cdot\text{laten}))\cdot\text{willen} \vdash INFP}\backslash E$$

$$\begin{array}{rcl@{\quad}c@{\quad}l}
\lceil\text{zeggen}\rceil^{string} & = & \lambda x\lambda f.(f\ x\ \texttt{zeggen}) & :: & \sigma \multimap \sigma^{(2)} \\
\lceil\text{willen}\rceil^{string} & = & \lambda q\lambda f.(q\ \lambda y\lambda z.(f\ y\ \texttt{willen}\cdot z)) & :: & \sigma^{(2)} \multimap \sigma^{(2)} \\
\lceil\text{laten}\rceil^{string} & = & \lambda q\lambda x\lambda f.(q\ \lambda z\lambda w.(f\ x\cdot z\ \texttt{laten}\cdot w)) & :: & \sigma^{(2)} \multimap \sigma \multimap \sigma^{(2)}
\end{array}$$

$$\lceil\dagger\rceil^{string} = \lambda f.(f\ \texttt{haar}\cdot\texttt{iets}\ \ \texttt{willen}\cdot\texttt{laten}\cdot\texttt{zeggen})$$

compare $\lceil\dagger\rceil^{sem} = \textsc{want}\ (\textsc{let}\ (\textsc{say}\ \textsc{something})\ \textsc{her})$

Dependency enhancement

function types $A\backslash B \rightsquigarrow \Diamond_d A\backslash B$

vc: verbal complement

$$\frac{\text{iets}}{np}$$

$$\frac{np}{\langle\text{iets}\rangle^{obj} \vdash \Diamond_{obj} np}\;\Diamond I \qquad \frac{\text{zeggen}}{\Diamond_{obj} np\backslash inf}$$

$$\frac{\langle\text{iets}\rangle^{obj} \vdash \Diamond_{obj} np \qquad \Diamond_{obj} np\backslash inf}{\langle\text{iets}\rangle^{obj} \cdot \text{zeggen} \vdash inf}\;\backslash E$$

$$\frac{\langle\text{iets}\rangle^{obj} \cdot \text{zeggen} \vdash inf}{\langle\langle\text{iets}\rangle^{obj} \cdot \text{zeggen}\rangle^{vc} \vdash \Diamond_{vc} inf}\;\Diamond I \qquad \frac{\text{laten}}{\Diamond_{vc} inf\backslash(\Diamond_{obj} np\backslash inf)}$$

$$\frac{\langle\langle\text{iets}\rangle^{obj} \cdot \text{zeggen}\rangle^{vc} \vdash \Diamond_{vc} inf \qquad \Diamond_{vc} inf\backslash(\Diamond_{obj} np\backslash inf)}{\langle\langle\text{iets}\rangle^{obj} \cdot \text{zeggen}\rangle^{vc} \cdot \text{laten} \vdash \Diamond_{obj} np\backslash inf}\;\backslash E$$

$$\frac{\dfrac{\text{haar}}{np}}{\langle\text{haar}\rangle^{obj} \vdash \Diamond_{obj} np}\;\Diamond I$$

$$\frac{\langle\text{haar}\rangle^{obj} \vdash \Diamond_{obj} np \qquad \langle\langle\text{iets}\rangle^{obj} \cdot \text{zeggen}\rangle^{vc} \cdot \text{laten} \vdash \Diamond_{obj} np\backslash inf}{\langle\text{haar}\rangle^{obj} \cdot (\langle\langle\text{iets}\rangle^{obj} \cdot \text{zeggen}\rangle^{vc} \cdot \text{laten}) \vdash inf}\;\backslash E$$

$$\frac{\langle\text{haar}\rangle^{obj} \cdot (\langle\langle\text{iets}\rangle^{obj} \cdot \text{zeggen}\rangle^{vc} \cdot \text{laten}) \vdash inf}{\langle\langle\text{haar}\rangle^{obj} \cdot (\langle\langle\text{iets}\rangle^{obj} \cdot \text{zeggen}\rangle^{vc} \cdot \text{laten})\rangle^{vc} \vdash \Diamond_{vc} inf}\;\Diamond I \qquad \frac{\text{willen}}{\Diamond_{vc} inf\backslash inf}$$

$$\frac{\langle\langle\text{haar}\rangle^{obj} \cdot (\langle\langle\text{iets}\rangle^{obj} \cdot \text{zeggen}\rangle^{vc} \cdot \text{laten})\rangle^{vc} \vdash \Diamond_{vc} inf \qquad \Diamond_{vc} inf\backslash inf}{\langle\langle\text{haar}\rangle^{obj} \cdot (\langle\langle\text{iets}\rangle^{obj} \cdot \text{zeggen}\rangle^{vc} \cdot \text{laten})\rangle^{vc} \cdot \text{willen} \vdash inf}\;\backslash E$$

$$(\text{WANT}\ \ \triangle^{vc}((\text{LET}\ \ \triangle^{vc}(\text{SAY}\ \ \triangle^{obj}\ \text{SOMETHING}))\ \ \triangle^{obj}\ \text{HER}))$$

Logical plus structural reasoning

Alternating logical/structural phases $|\Gamma|$: antecedent structure term with yield Γ

$$\dfrac{\dfrac{\vdots}{|\text{haar iets zeggen laten willen}| \vdash INFP}\ \textit{logical phase}}{\dfrac{?}{|\text{haar iets willen laten zeggen}| \vdash INFP}\ \textit{structural phase}}$$

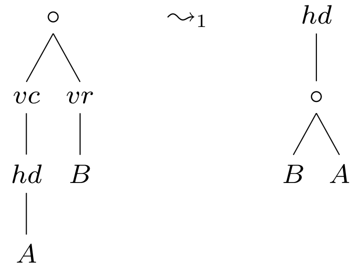
Challenge To reach the end sequent, the VR triggers have to break into the dependency domain of their infinitival complements.

As with Pollard's Head Grammars, we need head marking:

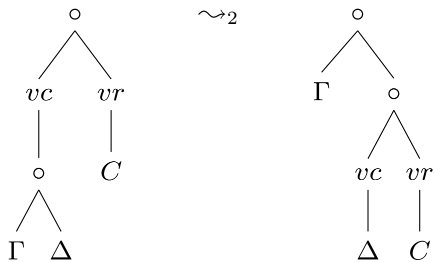
$$\begin{array}{rcl}
\text{zeggen} :: \Diamond_{obj} np \backslash inf & \rightsquigarrow & \Box_{hd}(\Diamond_{obj} np \backslash inf) \\
\text{willen} :: \Diamond_{vc} inf \backslash inf & \rightsquigarrow & \Box_{vr}(\Diamond_{vc} inf \backslash inf) \\
\text{laten} :: \Diamond_{vc} inf \backslash (\Diamond_{obj} np \backslash inf) & \rightsquigarrow & \Box_{vr}(\Diamond_{vc} inf \backslash (\Diamond_{obj} np \backslash inf))
\end{array}$$

Structural transformations

- vr trigger in construction with the head hd of its vc merges into complex hd:



- controlled asso: restructuring disassembles complex verbal complements:



iets willen zeggen

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\text{iets}}{np}
}{\langle\text{iets}\rangle^{obj} \vdash \Diamond_{obj}\, np}\,\Diamond I
\qquad
\dfrac{
\dfrac{\text{zeggen}}{\Box_{hd}(\Diamond_{obj}\, np \backslash inf)}
}{\langle\text{zeggen}\rangle^{hd} \vdash \Diamond_{obj}\, np \backslash inf}\,\Box E
}{\langle\text{iets}\rangle^{obj} \cdot \langle\text{zeggen}\rangle^{hd} \vdash inf}\,\backslash E
}{\langle\langle\text{iets}\rangle^{obj} \cdot \langle\text{zeggen}\rangle^{hd}\rangle^{vc} \vdash \Diamond_{vc}\, inf}\,\Diamond I
\qquad
\dfrac{
\dfrac{\text{willen}}{\Box_{vr}(\Diamond_{vc}\, inf \backslash inf)}
}{\langle\text{willen}\rangle^{vr} \vdash \Diamond_{vc}\, inf \backslash inf}\,\Box E
}{
\dfrac{
\dfrac{\langle\langle\text{iets}\rangle^{obj} \cdot \langle\text{zeggen}\rangle^{hd}\rangle^{vc} \cdot \langle\text{willen}\rangle^{vr} \vdash inf}
{\langle\text{iets}\rangle^{obj} \cdot (\langle\langle\text{zeggen}\rangle^{hd}\rangle^{vc} \cdot \langle\text{willen}\rangle^{vr}) \vdash inf}\,R2
}{\langle\text{iets}\rangle^{obj} \cdot \langle\text{willen} \cdot \text{zeggen}\rangle^{hd} \vdash inf}\,R1
}\,\backslash E
$$

$$(\blacktriangledown^{vr}\text{WANT}\ \ \triangle^{vc}(\blacktriangledown^{hd}\text{SAY}\ \ \triangle^{obj}\ \text{SOMETHING}))$$

- joint effect: verb raising trigger left-adjoined to the head of its verbal complement
- more complex clusters: $R1 \circ R2^{+}$

Conclusion

- the ACG method
  - ▷ full burden is put on the type homomorphism
  - ☹ $\lceil inf \rceil = (\sigma \multimap \sigma \multimap \sigma) \multimap \sigma$: 3rd order
  - ☺ combinatorics simple: application (plus β simplification)
- logical plus structural reasoning
  - ▷ division of labour between logical and structural inferences
  - ▷ logical phase: direct input to semantic composition
  - ▷ structural phase: meaning-preserving transformations

What's next dependency-enhanced type logic in action:

- supertagging, neural parsing — tomorrow
- experimental evaluation — Friday

□

References

Adriana D. Correia, Henk T. C. Stoof, and Michael Moortgat. Putting a spin on language: A quantum interpretation of unary connectives for linguistic applications. In *Proceedings 17th International Conference on Quantum Physics and Logic*, volume 340 of *EPTCS*, pages 114–140, 2020. doi: 10.4204/EPTCS.340.6. URL https://doi.org/10.4204/EPTCS.340.6.

Philippe de Groote and Sylvain Pogodalla. On the expressive power of abstract categorial grammars: Representing context-free formalisms. *J. Log. Lang. Inf.*, 13 (4):421–438, 2004. doi: 10.1007/s10849-004-2114-x. URL https://doi.org/10.1007/s10849-004-2114-x.

Herman Hendriks. The logic of tune - A proof-theoretic analysis of intonation. In *Proceedings Logical Aspects of Computational Linguistics LACL'97*, volume 1582 of *Lecture Notes in Computer Science*, pages 132–159. Springer, 1997. doi: 10.1007/3-540-48975-4_7. URL https://doi.org/10.1007/3-540-48975-4_7.

Konstantinos Kogkalidis and Michael Moortgat. Geometry-aware supertagging with heterogeneous dynamic convolutions. *CoRR*, abs/2203.12235, 2022. doi: 10.48550/arXiv.2203.12235. URL https://doi.org/10.48550/arXiv.2203.12235.

Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. Æthel: Automatically extracted typelogical derivations for dutch. In *Proceedings of The 12th Language*

Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020, pages 5257–5266. European Language Resources Association, 2020a. URL https://aclanthology.org/2020.lrec-1.647/.

Konstantinos Kogkalidis, Michael Moortgat, and Richard Moot. Neural proof nets. In Raquel Fernández and Tal Linzen, editors, *Proceedings of the 24th Conference on Computational Natural Language Learning, CoNLL 2020, Online, November 19-20, 2020*, pages 26–40. Association for Computational Linguistics, 2020b. doi: 10.18653/v1/2020.conll-1.3. URL https://doi.org/10.18653/v1/2020.conll-1.3.

Natasha Kurtonina and Michael Moortgat. Structural control. In Patrick Blackburn and Maarten de Rijke, editors, *Specifying Syntactic Structures*, pages 75–113. CSLI, Stanford, 1997.

Michael Moortgat and Gijs Wijnholds. Lexical and derivational meaning in vector-based models of relativisation. *CoRR*, abs/1711.11513, 2017. URL http://arxiv.org/abs/1711.11513.

Michael Moortgat, Mehrnoosh Sadrzadeh, and Gijs Wijnholds. A Frobenius algebraic analysis for parasitic gaps. *J of Applied Logics*, 7(5):823–852, 2020. URL http://collegepublications.co.uk/ifcolog/?00041.

Richard Moot and Christian Retoré. *The Logic of Categorial Grammars - A Deductive Account of Natural Language Syntax and Semantics*, volume 6850 of *Lecture Notes* *in Computer Science*. Springer, 2012. ISBN 978-3-642-31554-1. doi: 10.1007/978-3-642-31555-8. URL https://doi.org/10.1007/978-3-642-31555-8.

Carl Pollard. Head grammars, generalized phrase structure grammars, and natural language, 1984.

Mehrnoosh Sadrzadeh and Reinhard Muskens. Static and dynamic vector semantics for lambda calculus models of natural language. *J. Lang. Model.*, 6(2):319–351, 2018. doi: 10.15398/jlm.v6i2.228. URL https://doi.org/10.15398/jlm.v6i2.228.