Type-Logical Grammar and Natural Language Syntax

Yusuke Kubota

University of Tsukuba
kubota.yusuke.fn@u.tsukuba.ac.jp

LACompLing 2018
Outline

Overview of **Hybrid Type-Logical Grammar**

- A version of Type-Logical Grammar jointly developed with Bob Levine (OSU)
Outline

Overview of **Hybrid Type-Logical Grammar**

▶ A version of Type-Logical Grammar jointly developed with Bob Levine (OSU)

Outline of presentation

▶ Motivations
▶ Basic architecture
▶ Linguistic application – Gapping in English
Outline

Overview of Hybrid Type-Logical Grammar

▶ A version of Type-Logical Grammar jointly developed with Bob Levine (OSU)

Outline of presentation

▶ Motivations
▶ Basic architecture
▶ Linguistic application – Gapping in English
▶ Larger issues, open questions
  ▶ Comparison with some recent HPSG work
  ▶ Formal properties of Hybrid TLG
  ▶ Parsing
Hybrid Type-Logical Grammar [Kubota, 2010, Kubota and Levine, 2015]

Motivations:

- Logic-based version of CG
- (Relatively) easy to use for linguists
- Wide empirical coverage
Hybrid Type-Logical Grammar [Kubota, 2010, Kubota and Levine, 2015]

Motivations:

- Logic-based version of CG
- (Relatively) easy to use for linguists
- Wide empirical coverage

Hybrid TLG builds on two lines of research in Type-Logical Grammar:

- **Lambek calculus** [Lambek, 1958] and its extensions
  ‘Syntax can be done (mostly) with order-sensitive implication’

  ‘Get rid of word order from syntax’
  local combinatorics non-local dependencies
  (e.g. coordination) (e.g. extraction, scope)
Hybrid Type-Logical Grammar [Kubota, 2010, Kubota and Levine, 2015]

Motivations:

▶ Logic-based version of CG
▶ (Relatively) easy to use for linguists
▶ Wide empirical coverage

Hybrid TLG builds on two lines of research in Type-Logical Grammar:

▶ **Lambek calculus** [Lambek, 1958] and its extensions
  ‘Syntax can be done (mostly) with order-sensitive implication’
  ‘Get rid of word order from syntax’
Hybrid Type-Logical Grammar [Kubota, 2010, Kubota and Levine, 2015]

Motivations:

▶ Logic-based version of CG
▶ (Relatively) easy to use for linguists
▶ Wide empirical coverage

Hybrid TLG builds on two lines of research in Type-Logical Grammar:

▶ **Lambek calculus** [Lambek, 1958] and its extensions
  ‘Syntax can be done (mostly) with order-sensitive implication’
  ‘Get rid of word order from syntax’

<table>
<thead>
<tr>
<th></th>
<th>local combinatorics (e.g. coordination)</th>
<th>non-local dependencies (e.g. extraction, scope)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lambek calculus</td>
<td>✓</td>
<td>*</td>
</tr>
<tr>
<td>λ grammars</td>
<td>*</td>
<td>✓</td>
</tr>
</tbody>
</table>
Hybrid Type-Logical Grammar [Kubota, 2010, Kubota and Levine, 2015]

Motivations:

- Logic-based version of CG
- (Relatively) easy to use for linguists
- Wide empirical coverage

Hybrid TLG builds on two lines of research in Type-Logical Grammar:

- **Lambek calculus** [Lambek, 1958] and its extensions
  ‘Syntax can be done (mostly) with order-sensitive implication’
  ‘Get rid of word order from syntax’

<table>
<thead>
<tr>
<th></th>
<th>local combinatorics (e.g. coordination)</th>
<th>non-local dependencies (e.g. extraction, scope)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lambek calculus</td>
<td>✓</td>
<td>*</td>
</tr>
<tr>
<td>λ grammars</td>
<td>*</td>
<td>✓</td>
</tr>
</tbody>
</table>

- Hybrid TLG \(\approx\) Lambek calculus + λ grammar
Empirical results:

- coordination
  - nonconstituent coordination [Kubota and Levine, 2015]
  - Gapping [Kubota and Levine, 2016a]
- scopal operators *(same/different, respectively)* [Kubota and Levine, 2016b]
- ellipsis
  - pseudogapping [Kubota and Levine, 2017]
  - stripping [Puthawala, 2018]
  - comparatives [Vaikšnoraitė, 2018]
Lambek calculus
Syntactic types

- $\mathcal{A} := \{ \text{N, NP, S, ...} \}$ (atomic type)
- $\mathcal{T} := \mathcal{A} \mid \mathcal{T}\mathcal{T} \mid \mathcal{T}/\mathcal{T}$ (type)
Lambek calculus

Syntactic types

- ▶ \( \mathcal{A} := \{ \text{N, NP, S, . . .} \} \) (atomic type)
- ▶ \( \mathcal{T} := \mathcal{A} | \mathcal{T} \setminus \mathcal{T} | \mathcal{T} / \mathcal{T} \) (type)

Syntactic rules of the Lambek calculus

**Forward Slash Elimination**

\[
\frac{A/B \quad B}{A} \quad /E
\]

**Backward Slash Elimination**

\[
\frac{B \quad B \setminus A}{A} \quad \setminus E
\]

**Forward Slash Introduction**

\[
\vdash [A]^n
\]

**Backward Slash Introduction**

\[
\vdash [A]^n
\]
Sample derivation

(1)

\[
\begin{array}{c}
\frac{(NP\backslash S)/NP}{NP} & [NP]^1 \\
\frac{NP}{NP\backslash S} & /E \\
\frac{S}{S/\text{NP}} & /I^1 \\
\frac{((S/\text{NP})\backslash S)/N}{N} & (S/\text{NP})\backslash S \\
\frac{S}{S/\text{NP}} & /I^1 \\
\end{array}
\]

\[NP (NP\backslash S)/NP ((S/\text{NP})\backslash S)/N N \vdash S\]

John saw every student
Derivation with prosodic term labelling (cf. [Morrill, 1994])

(2)

\[
\begin{align*}
\text{saw;} & \quad \varphi; \\
\text{saw;} & \quad x; \\
\text{(NP\S)/NP} & \quad [\text{NP}] \\
\text{john; saw \bullet \varphi;} & \quad \text{every; student;} \\
\text{j; NP saw}(x); \text{NP\S} & \quad \forall; \quad \text{student;} \\
\text{\frac{\text{john \bullet saw \bullet \varphi;}}{	ext{saw}(x)(j); S}} & \quad \frac{((\text{S/NP}\S)/\text{N})/\text{N}}{	ext{every \bullet student;}} \\
\text{\frac{\text{\lambda x. saw}(x)(j); S/\text{NP}}{	ext{john \bullet saw;}} & \quad \frac{\forall_{\text{student;}}}{(\text{S/NP}\S)}} \quad \forall_{\text{student;}} (\text{S/NP}\S) \\
\text{\frac{\text{\lambda x. saw}(x)(j); S/\text{NP}}{\text{john \bullet saw \bullet every \bullet student;}} & \quad \text{\forall_{\text{student;}} (\lambda x. \text{saw}(x)(j)); S}} \\
\text{\lambda x. saw}(x)(j); S/\text{NP} & \quad \forall_{\text{student;}} (\lambda x. \text{saw}(x)(j)); S
\end{align*}
\]
Lambek calculus, with semantic and prosodic term-labelling

**Forward Slash Introduction**

\[
\begin{array}{c}
\vdash [\varphi; x; A]^n \\
\vdash \\
\hline
b \cdot \varphi; \mathcal{F}; \! \! B \\
b; \lambda x.\mathcal{F}; \! \! B/A
\end{array}
\]

**Forward Slash Elimination**

\[
\begin{array}{c}
a; \mathcal{F}; A/B \\
b; \mathcal{G}; B
\end{array}
\]

\[
\frac{a \cdot b; \mathcal{F}(\mathcal{G}); A}{/E}
\]

\text{Note: the prosodic terms are not proof terms in this setup.}

\text{Labelled deduction for notating the prosody (cf. [Morrill, 1994, Oehrle, 1994]).}
Lambek calculus, with semantic and prosodic term-labelling

Forward Slash Introduction

\[
\frac{[\varphi; x; A]^n}{b \bullet \varphi; \mathcal{F}; B} \\
\frac{b; \lambda x.\mathcal{F}; B/A}{|n}
\]

Forward Slash Elimination

\[
\frac{a; \mathcal{F}; A/B \quad b; \mathcal{G}; B}{a \bullet b; \mathcal{F}(\mathcal{G}); A} \quad /E
\]

Backward Slash Introduction

\[
\frac{[\varphi; x; A]^n}{\varphi \bullet b; \mathcal{F}; B} \\
\frac{b; \lambda x.\mathcal{F}; A/B}{|n}
\]

Backward Slash Elimination

\[
\frac{b; \mathcal{G}; B \quad a; \mathcal{F}; B/A}{b \bullet a; \mathcal{F}(\mathcal{G}); A} \quad \backslash E
\]
Lambek calculus, with semantic and prosodic term-labelling

Forward Slash Introduction

\[
\vdash [\varphi; x; A]^n
\]

\[
\vdash \vdash
\]

\[
\frac{\varphi \bullet \varphi; \mathcal{F}; B}{b; \lambda x. \mathcal{F}; B/A}
\]

\[
\frac{a; \mathcal{F}; A/B}{a \bullet \varphi; \mathcal{F}; B}
\]

Forward Slash Elimination

\[
\frac{a; \mathcal{F}; A/B \quad b; \mathcal{G}; B}{b; \lambda x. \mathcal{F}; B/A}
\]

\[
\frac{a \bullet b; \mathcal{F}(\mathcal{G}); A}{}/E
\]

Backward Slash Introduction

\[
\vdash [\varphi; x; A]^n
\]

\[
\vdash \vdash
\]

\[
\frac{\varphi \bullet b; \mathcal{F}; B}{b; \lambda x. \mathcal{F}; A/B}
\]

\[
\frac{b; \mathcal{G}; B \quad a; \mathcal{F}; B\setminus A}{b \bullet a; \mathcal{F}(\mathcal{G}); A}
\]

\[
\frac{b \bullet a; \mathcal{F}(\mathcal{G}); A}{}/E
\]

- Labelled deduction for notating the prosody (cf. [Morrill, 1994, Oehrle, 1994]).

Note: the prosodic terms are not proof terms in this setup.
Lambek calculus

Forward Slash Introduction

\[
\begin{array}{c}
\vdash [\phi; x; A]^n \\
\vdash \vdash \\
\hline
b \bullet \phi; \mathcal{F}; B \\
b; \lambda x. \mathcal{F}; B/A \\
\end{array}
\]

Backward Slash Introduction

\[
\begin{array}{c}
\vdash [\phi; x; A]^n \\
\vdash \vdash \\
\hline
\phi \bullet b; \mathcal{F}; B \\
b; \lambda x. \mathcal{F}; A\backslash B \\
\end{array}
\]

Forward Slash Elimination

\[
\begin{array}{c}
a; \mathcal{F}; A/B \\
b; \mathcal{G}; B \\
\hline
a \bullet b; \mathcal{F}(\mathcal{G}); A \\
\end{array}
\]

Backward Slash Elimination

\[
\begin{array}{c}
b; \mathcal{G}; B \\
a; \mathcal{F}; B\backslash A \\
\hline
b \bullet a; \mathcal{F}(\mathcal{G}); A \\
\end{array}
\]

But... What about syntactic movement?

(3) a. I don't know who [John met i at the party].

b. John met every student yesterday.
Lambek calculus

Forward Slash Introduction

\[
\begin{array}{c}
\phi; x; A^n \\
\hline
\phi; b; F; B \\
\hline
b; \lambda x. F; B/A
\end{array}
\]

Backward Slash Introduction

\[
\begin{array}{c}
\phi; x; A^n \\
\hline
\phi; b; F; B \\
\hline
b; \lambda x. F; A\setminus B
\end{array}
\]

But...

Forward Slash Elimination

\[
\frac{a; F; A\setminus B}{a \cdot b; F(G); A}
\]

Backward Slash Elimination

\[
\frac{b; G; B}{b \cdot a; F(G); A}
\]
**Lambek calculus**

**Forward Slash Introduction**

\[
\frac{\phi; \lambda x. F; B/A}{b \cdot \phi; F; B} /^n
\]

**Forward Slash Elimination**

\[
\frac{a; F; A/B \quad b; G; B}{a \cdot b; F(G); A} /^E
\]

**Backward Slash Introduction**

\[
\frac{\phi; \lambda x. F; B/A}{b \cdot a; F; B} \backslash^n
\]

**Backward Slash Elimination**

\[
\frac{b; G; B \quad a; F; B\backslash A}{b \cdot a; F(G); A} \backslash^E
\]

But... What about syntactic movement?

(3)  
a. I don’t know \textit{who}_i [John met ___i at the party].

b. John met \textit{every student} yesterday.
Syntactic movement in the Lambek calculus?

(4)

```
S  
   /\  
  S  S  
     /\  
    λy S  
       /\  
      NP VP  
      /\ \  
     John VP PP  
      /\  
     V NP yesterday  
      /\  
    met NP y
```

everyone

Yusuke Kubota  Type-Logical Grammar and Natural Language Syntax  10/44
Syntactic movement in the Lambek calculus?

(4) $\lambda y$ NP $\rightarrow$ VP
    NP John $\rightarrow$ VP
       V met $\rightarrow$ PP
          NP yesterday

(5) $\frac{\text{met; (NP \backslash S)/NP } t \; ; \; NP}{\text{met } \bullet t \; ; \; NP \backslash S \quad /E \quad \text{yesterday; (NP \backslash S)\backslash(NP \backslash S) \quad \backslash E}}$

$\frac{\text{john; NP}}{\text{met } \bullet t \bullet \text{yesterday; NP \backslash S \quad \backslash E}}$

$\frac{\text{everyone; NP}}{\text{john } \bullet \text{met } \bullet t \bullet \text{yesterday; S \quad \backslash E}}$

Yusuke Kubota Type-Logical Grammar and Natural Language Syntax 10/44
‘Vertical Slash’ for movement (cf. [Oehrle, 1994])

(6) John saw everyone yesterday.

(7)

\[
\text{saw; see; } (\text{NP}\backslash\text{S})/\text{NP} \quad [\varphi; \quad \begin{array}{c}
[\ x; \\
\text{NP} \end{array}]^1 \\
\text{//E } \quad \text{yesterday;}
\]

\[
\text{john; see}(x); \text{NP}\backslash\text{S} \quad (\text{NP}\backslash\text{S})\backslash(\text{NP}\backslash\text{S})
\]

\[
\text{j; NP} \quad \text{saw} \bullet \varphi \bullet \text{yesterday; yest}((\text{see}(x))); \text{NP}\backslash\text{S}
\]

\[
\text{john} \bullet \text{saw} \bullet \varphi \bullet \text{yesterday; yest}((\text{see}(x)))(j); \text{S}
\]
‘Vertical Slash’ for movement (cf. [Oehrle, 1994])

(6) John saw everyone yesterday.

(7)\[
\begin{array}{c}
\text{saw;} \\
\text{see;} \\
(NP\backslash S)/NP
\end{array}
\]

\[
\begin{array}{c}
\varphi; \\
x; \\
NP
\end{array}
\]

\[
\begin{array}{c}
\text{john;} \\
j; NP
\end{array}
\]

\[
\begin{array}{c}
\text{see}(x); NP\backslash S \\
(NP\backslash S)\backslash (NP\backslash S)
\end{array}
\]

\[
\begin{array}{c}
saw \cdot \varphi; \\
yest(\text{see}(x)); NP\backslash S
\end{array}
\]

\[
\begin{array}{c}
\text{john} \cdot \text{saw} \cdot \varphi \cdot \text{yesterday}; \text{yest}(\text{see}(x))(j); S \\
(\varphi; \text{yesterday}; \text{yest}(\text{see}(x))(j); S)\downarrow
\end{array}
\]

\[
\begin{array}{c}
\lambda \varphi. \text{john} \cdot \text{saw} \cdot \varphi \cdot \text{yesterday}; \lambda x. \text{yest}(\text{see}(x))(j); S\uparrow
\end{array}
\]
‘Vertical Slash’ for movement (cf. [Oehrle, 1994])

(6)  John saw everyone yesterday.

(7)  \[\lambda \sigma \sigma \text{(everyone)}; \ V_{\text{person}}; S \downarrow (S \uparrow \text{NP}) \]

\[\lambda \varphi. \text{john} \bullet \text{saw} \bullet \varphi \bullet \text{yesterday}; \ \lambda x. \text{yest}(\text{see}(x))(j); S \uparrow \text{NP} \]

\[\lambda \sigma. [\sigma \text{(everyone)}](\lambda \varphi. \text{john} \bullet \text{saw} \bullet \varphi \bullet \text{yesterday}); \ V_{\text{person}}(\lambda x. \text{yest}(\text{see}(x))(j)); S \]

where \[V_{\text{person}} = \text{def} \ \lambda P[\forall x. \text{person}(x) \rightarrow P(x)]\]
‘Vertical Slash’ for movement (cf. [Oehrle, 1994])

(6) John saw everyone yesterday.

\[
\lambda \sigma. \sigma (\text{everyone}); \quad V_{\text{person}}; S \rhd (S \downarrow \text{NP})
\]

\[
\quad \lambda \varphi. \text{john} \cdot \text{saw} \cdot \varphi \cdot \text{yesterday}; \quad \lambda x. \text{yest} (\text{see} (x)) (j); \quad S
\]

\[
\quad \lambda \varphi. [\text{john} \cdot \text{saw} \cdot \varphi \cdot \text{yesterday}] (\text{everyone}); \quad V_{\text{person}} (\lambda x. \text{yest} (\text{see} (x)) (j)); \quad S
\]

where \( \quad V_{\text{person}} = \text{def} \quad \lambda P [\forall x. \text{person} (x) \rightarrow P (x)] \)
‘Vertical Slash’ for movement (cf. [Oehrle, 1994])

(6) John saw everyone yesterday.

\[
\begin{align*}
\lambda \sigma. \sigma(everyone); \\
V_{person}; S \vdash (S|NP) \\
\lambda \phi. \lambda P[\forall x. person(x) \rightarrow P(x)]
\end{align*}
\]

\[
\begin{align*}
\lambda \phi. john \bullet saw \bullet \phi \bullet yesterday; \ yest(see(x))(j); \ S \\
S \vdash NP \setminus S \\
S \setminus NP
\end{align*}
\]
‘Vertical Slash’ for movement (cf. [Oehrle, 1994])

(6) John saw everyone yesterday.

(7) \[\lambda \sigma. \sigma(\text{everyone}); V_{\text{person}}; S \mid (S \mid \text{NP}) \]

\[\lambda \sigma.[\sigma(\text{everyone})](\lambda \varphi. \text{john} \bullet \text{saw} \bullet \varphi \bullet \text{yesterday}); V_{\text{person}}(\lambda x. \text{yest}(\text{see}(x))(j)); S\]

where \( V_{\text{person}} =_{\text{def}} \lambda P[\forall x. \text{person}(x) \rightarrow P(x)] \)

\(\to\) (vertical slash): order-insensitive implication
Scope ambiguity

(8)

\[
\begin{align*}
\lambda \sigma. \sigma(\text{someone}); & \quad \exists \text{person}; \\
S \upharpoonright (S \upharpoonright NP) & \quad S \upharpoonright (S \upharpoonright NP) \\
\end{align*}
\]

\[
\begin{align*}
\lambda \sigma. \sigma(\text{everyone}); & \quad \forall \text{person}; \\
S \upharpoonright (S \upharpoonright NP) & \quad S \upharpoonright (S \upharpoonright NP) \\
\end{align*}
\]

\[
\begin{align*}
\lambda \varphi_1. \text{someone} \bullet \text{talked} \bullet \text{to} \bullet \varphi_1 \bullet \text{yesterday}; & \quad \lambda x_1. \forall \text{person}(\lambda x_2. \text{yest}(\text{talked-to}(x_1)(x_2))); S \\
\end{align*}
\]

\[
\begin{align*}
\varphi_2 \bullet \text{talked} \bullet \text{to} \bullet \varphi_1; & \quad \text{yest}(\text{talked-to}(x_1)(x_2)); S \\
\end{align*}
\]

\[
\begin{align*}
\varphi_2 \bullet \text{talked} \bullet \text{to} \bullet \varphi_1 \bullet \text{yesterday}; & \quad S \upharpoonright NP \\
\end{align*}
\]

\[
\begin{align*}
\varphi_2 \bullet \text{talked} \bullet \text{to} \bullet \varphi_1; & \quad \text{yest}; S \upharpoonright S \\
\end{align*}
\]

\[
\begin{align*}
\varphi_2 \bullet \text{talked} \bullet \text{to} \bullet \varphi_1 \bullet \text{yesterday}; & \quad \text{yest}; S \upharpoonright S \\
\end{align*}
\]

\[
\begin{align*}
\varphi_2 \bullet \text{talked} \bullet \text{to} \bullet \varphi_1 \bullet \text{yesterday}; & \quad S \upharpoonright NP \\
\end{align*}
\]
Rules for the vertical slash

Vertical Slash Introduction

\[
\vdash [\varphi; x; A]^n \vdash \\
\vdash b; F; B \quad \lambda \varphi. b; \lambda x. F; B \upharpoonright A
\]

Vertical Slash Elimination

\[
\frac{a; F; A \upharpoonright B \quad b; G; B}{a(b); F(G); A} \quad [E]
\]
Syntactic types in Hybrid TLG

- $A := \{ S, NP, N, \ldots \}$ (atomic type)
- $D := A | D \setminus D | D / D$ (directional type)

Some examples:
- $S / NP \in T$
- $S \restriction (S \restriction NP) \in T$
- $S \restriction (S / NP) \in T$
- $S / (S \restriction NP) \notin T$

Note: A vertical slash cannot occur under a Lambek slash.
Syntactic types in Hybrid TLG

- $\mathcal{A} := \{ S, \text{NP}, N, \ldots \}$ (atomic type)
- $\mathcal{D} := \mathcal{A} | \mathcal{D}\setminus\mathcal{D} | \mathcal{D}/\mathcal{D}$ (directional type)
- $\mathcal{T} := \mathcal{D} | \mathcal{T}\setminus\mathcal{T}$ (type)

Some examples:
- $S/\text{NP} \in \mathcal{T}$
- $S \setminus (S \setminus \text{NP}) \in \mathcal{T}$
- $S \setminus (S/\text{NP}) \notin \mathcal{T}$

Note: A vertical slash cannot occur under a Lambek slash.
Syntactic types in Hybrid TLG

- $\mathcal{A} := \{ \text{S, NP, N, \ldots} \}$ (atomic type)
- $\mathcal{D} := \mathcal{A} \mid \mathcal{D}\setminus\mathcal{D} \mid \mathcal{D}/\mathcal{D}$ (directional type)
- $\mathcal{T} := \mathcal{D} \mid \mathcal{T}\mid\mathcal{T}$ (type)

Some examples:

- $\text{S/NP} \in \mathcal{T}$
- $\text{S}(\text{S/NP}) \in \mathcal{T}$
- $\text{S}(\text{S/NP}) \not\in \mathcal{T}$

Note: A vertical slash cannot occur under a Lambek slash.
Syntactic types in Hybrid TLG

- \( \mathcal{A} := \{ \text{S, NP, N, ...} \} \) (atomic type)
- \( \mathcal{D} := \mathcal{A} | \mathcal{D}\backslash\mathcal{D} | \mathcal{D}/\mathcal{D} \) (directional type)
- \( \mathcal{T} := \mathcal{D} | \mathcal{T}\upharpoonright\mathcal{T} \) (type)

Some examples:

- \( \text{S/NP} \in \mathcal{T} \)
- \( \text{S}\upharpoonright(\text{S}\upharpoonright\text{NP}) \in \mathcal{T} \)
- \( \text{S}\upharpoonright(\text{S/NP}) \in \mathcal{T} \)
- \( \text{S}/(\text{S}\upharpoonright\text{NP}) \notin \mathcal{T} \)

**Note:** A vertical slash cannot occur under a Lambek slash.
## Rules in Hybrid TLG

<table>
<thead>
<tr>
<th>Connective</th>
<th>Introduction</th>
<th>Elimination</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{\varphi; x; A}{b \cdot \varphi; \mathcal{F}; B} )</td>
<td>( b; \lambda x.\mathcal{F}; B/A )</td>
<td>( \frac{a; \mathcal{F}; A/B}{b; \mathcal{G}; B} )</td>
</tr>
<tr>
<td>( \frac{\varphi; x; A}{\varphi \cdot b; \mathcal{F}; B} )</td>
<td>( b; \lambda x.\mathcal{F}; A\backslash B )</td>
<td>( \frac{b; \mathcal{G}; B}{a; \mathcal{F}; B\backslash A} )</td>
</tr>
<tr>
<td>( \frac{\varphi; x; A}{b; \mathcal{F}; B} )</td>
<td>( \lambda \varphi. b; \lambda x.\mathcal{F}; B\backslash A )</td>
<td>( \frac{a; \mathcal{F}; A\backslash B}{a(b); \mathcal{F}(\mathcal{G}); A} )</td>
</tr>
</tbody>
</table>
## Rules in Hybrid TLG

### Lambda calculus

<table>
<thead>
<tr>
<th>Connective</th>
<th>Introduction</th>
<th>Elimination</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>$\vdash [\varphi; x; A]^n$</td>
<td>$\frac{a; \mathcal{F}; A/B \quad b; \mathcal{G}; B}{a \cdot b; \mathcal{F}(\mathcal{G}); A}$ /E</td>
</tr>
<tr>
<td>(\lambda x.\mathcal{F}; B/A)</td>
<td>$\frac{b \cdot \varphi; \mathcal{F}; B}{b; \lambda x.\mathcal{F}; B/A}$ /I$^m$</td>
<td>$\frac{b; \mathcal{G}; B \quad a; \mathcal{F}; B\backslash A}{b \cdot a; \mathcal{F}(\mathcal{G}); A}$ \E</td>
</tr>
<tr>
<td>(\varphi \cdot b; \mathcal{F}; B)</td>
<td>$\frac{\varphi \cdot b; \mathcal{F}; B}{b; \lambda x.\mathcal{F}; A\backslash B}$ \I$^m$</td>
<td></td>
</tr>
</tbody>
</table>

\[\vdash [\varphi; x; A]^n\]

\[a; \mathcal{F}; A/B \quad b; \mathcal{G}; B\]

\[a \cdot b; \mathcal{F}(\mathcal{G}); A\]

\[b; \lambda x.\mathcal{F}; B/A\]
Rules in Hybrid TLG

Connective Introduction Elimination

\[\lambda\text{ grammar} \quad \vdash \quad \begin{align*}
\varphi; x; A^n & \quad : \\
\vdash & \quad : \\
\vdash & \quad : \\
\vdash & \quad : \\
b; \mathcal{F}; B & \quad \vdash \\
\vdash & \quad \vdash \\
\lambda\varphi. b; \lambda x. \mathcal{F}; B|A & \quad \vdash^n \\
\end{align*} \]

a; \mathcal{F}; A|B \quad b; \mathcal{G}; B \quad \vdash^n \\
a(b); \mathcal{F}(\mathcal{G}); A
Rules in Hybrid TLG

<table>
<thead>
<tr>
<th>Connective</th>
<th>Introduction</th>
<th>Elimination</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \vdash )</td>
<td>( [\varphi; x; A]^n )</td>
<td>( \frac{a; \mathcal{F}; A/B \quad b; \mathcal{G}; B}{a \cdot b; \mathcal{F}(\mathcal{G}); A} )</td>
</tr>
<tr>
<td>/</td>
<td>( b \cdot \varphi; \mathcal{F}; B )</td>
<td>( \frac{a; \mathcal{F}; A/B \quad b; \mathcal{G}; B}{b \cdot \lambda x.\mathcal{F}; B/A} )</td>
</tr>
<tr>
<td>( b; \lambda x.\mathcal{F}; B/A )</td>
<td>/( l^m )</td>
<td>/( l^m )</td>
</tr>
<tr>
<td>/</td>
<td>( \varphi \cdot b; \mathcal{F}; B )</td>
<td>( \frac{b; \mathcal{G}; B \quad a; \mathcal{F}; B/A}{b \cdot a; \mathcal{F}(\mathcal{G}); A} )</td>
</tr>
<tr>
<td>( b; \lambda x.\mathcal{F}; A/B )</td>
<td>/( l^m )</td>
<td>/( l^m )</td>
</tr>
<tr>
<td>( \uparrow )</td>
<td>( \vdash )</td>
<td>( a; \mathcal{F}; A/B )</td>
</tr>
<tr>
<td>( b; \mathcal{F}; B )</td>
<td>( \frac{a; \mathcal{F}; A/B \quad b; \mathcal{G}; B}{a(b); \mathcal{F}(\mathcal{G}); A} )</td>
<td></td>
</tr>
</tbody>
</table>
Gapping

Simple Gapping examples

(9)  a. Robin *speaks* French, and Leslie, ∅ German.

Scope anomaly in Gapping [Siegel, 1984, Oehrle, 1987]

(10) a. Mrs. J can't live in LA and Mr. J ∅ in Boston. (= ¬ [ϕ ∧ ψ])

b. Kim didn't play bingo or Sandy ∅ sit at home all evening. (= ¬ [ϕ ∨ ψ])
Gapping

Simple Gapping examples

(9)  

a. Robin **speaks** French, and Leslie, $\emptyset$ German.

b. Robin **wants to speak** French, and Leslie, $\emptyset$ German.

c. To Robin **Chris gave** the book, and to Leslie, $\emptyset$ the magazine.
Gapping

Simple Gapping examples

(9)  
   a. Robin \textbf{speaks} French, and Leslie, ∅ German.
   
   b. Robin \textbf{wants to speak} French, and Leslie, ∅ German.
   
   c. To Robin \textbf{Chris gave} the book, and to Leslie, ∅ the magazine.

Scope anomaly in Gapping [Siegel, 1984, Oehrle, 1987]

(10)  
   a. Mrs. J \textbf{can’t live} in LA and Mr. J ∅ in Boston.
       \hspace{0.4cm} (= \neg \lozenge [\varphi \land \psi])
       
   b. Kim \textbf{didn’t} play bingo or Sandy ∅ sit at home all evening.
       \hspace{0.4cm} (= \neg [\varphi \lor \psi])
Gapping in Hybrid TLG

Deriving a gapped sentence via hypothetical reasoning

\[(11)\]

\[
\begin{array}{c}
\frac{\varphi_1; \begin{array}{c} P; \text{VP/NP} \end{array}^1 \text{leslie; l; NP}}{\text{robin; r; NP \quad \varphi_1 \bullet \text{leslie}; P(l); \text{VP}}}
\end{array}
\]

\[
\begin{array}{c}
\frac{\text{robin} \bullet \varphi_1 \bullet \text{leslie}; P(l)(r); \text{S}}{\lambda \varphi_1.\text{robin} \bullet \varphi_1 \bullet \text{leslie}; \lambda P.P(l)(r); \text{S}\upharpoonright(\text{VP}/\text{NP})}
\end{array}
\]
Gapping as like-category coordination (cont.)

(12) \( \lambda \varphi_1. \text{robin} \bullet \varphi_1 \bullet \text{leslie}; \lambda \varphi.P.P(\text{l})(\text{r}); S\upharpoonright TV \)

(13) \( \lambda \varphi_1. \text{sandy} \bullet \varphi_1 \bullet \text{kim}; \lambda \varphi.P.P(\text{k})(\text{s}); S\upharpoonright TV \)

Conjunction entry for Gapping:

(14) \( \lambda \sigma_2 \lambda \sigma_1 \lambda \varphi_0. \sigma_1(\varphi_0) \bullet \text{and} \bullet \sigma_2(\varepsilon); \lambda \mathcal{W} \lambda \mathcal{V}. \mathcal{V} \sqcap \mathcal{W}; (S\upharpoonright TV)\upharpoonright (S\upharpoonright TV)\upharpoonright (S\upharpoonright TV) \)
Gapping as like-category coordination (cont.)

(15)

\[ \lambda \sigma_2 \lambda \sigma_1 \lambda \varphi_0. \]
\[ \sigma_1(\varphi_0) \cdot \text{and} \cdot \sigma_2(\varepsilon); \]
\[ \lambda \mathcal{W} \mathcal{V} \mathcal{V} \sqcap \mathcal{W}; \]
\[ (S \upharpoonright \mathcal{V}) \sqcap (S \upharpoonright \mathcal{V}) \sqcap (S \upharpoonright \mathcal{V}) \]

\[ \lambda \varphi_1. \text{sandy} \cdot \varphi_1 \cdot \text{kim}; \]
\[ \lambda P.P(k)(s); \]
\[ S \upharpoonright \mathcal{V} \]

\[ \lambda \varphi_1. \text{robin} \cdot \varphi_1 \cdot \text{leslie}; \]
\[ \lambda P.P(l)(r); \]
\[ S \upharpoonright \mathcal{V} \]

\[ \lambda \varphi_0[\text{robin} \cdot \varphi_0 \cdot \text{leslie} \cdot \text{and} \cdot \text{sandy} \cdot \varepsilon \cdot \text{kim}]; \]
\[ \lambda P.P(l)(r) \sqcap \lambda P.P(k)(s); S \upharpoonright \mathcal{V} \]
Gapping as like-category coordination (cont.)

(15)

\[
\lambda \sigma_2 \lambda \sigma_1 \lambda \varphi_0. \\
\sigma_1(\varphi_0) \bullet \text{and} \bullet \sigma_2(\varepsilon); \\
\lambda \mathcal{W} \lambda \mathcal{V}. \mathcal{V} \sqcap \mathcal{W}; \\
(S \upharpoonright \text{TV}) \sqcup (S \upharpoonright \text{TV}) \sqcup (S \upharpoonright \text{TV}) \\
: \\
\lambda \varphi_1. \text{sandy} \bullet \varphi_1 \bullet \text{kim}; \\
\lambda \mathcal{P}. \mathcal{P}(k)(s); \\
\text{S} \upharpoonright \text{TV}
\]

| \lambda \varphi_1. \text{robin} \bullet \varphi_1 \bullet \text{leslie}; | \lambda \sigma_1 \lambda \varphi_0. \sigma_1(\varphi_0) \bullet \text{and} \bullet \text{sandy} \bullet \varepsilon \bullet \text{kim}; |
| \lambda \mathcal{P}. \mathcal{P}(l)(r); | \lambda \mathcal{V}. \mathcal{V} \sqcap \lambda \mathcal{P}. \mathcal{P}(k)(s); |
| \text{S} \upharpoonright \text{TV} | (S \upharpoonright \text{TV}) \sqcup (S \upharpoonright \text{TV}) |

| \text{met; } | \lambda \varphi_0[\text{robin} \bullet \varphi_0 \bullet \text{leslie} \bullet \text{and} \bullet \text{sandy} \bullet \varepsilon \bullet \text{kim}]; |
| \text{TV} | \lambda \mathcal{P}. \mathcal{P}(l)(r) \sqcap \lambda \mathcal{P}. \mathcal{P}(k)(s); \text{S} \upharpoonright \text{TV} |

\[
\text{robin} \bullet \text{met} \bullet \text{leslie} \bullet \text{and} \bullet \text{sandy} \bullet \varepsilon \bullet \text{kim}; \\
\text{meet}(l)(r) \sqcap \text{meet}(k)(s); \text{S}
\]
Explaining scope anomaly in Gapping

(16) *robin* • ϕ1 • *leslie*

\[ \lambda \sigma_2 \lambda \sigma_1 \lambda \varphi_0. \]
\[ \sigma_1(\varphi_0) \text{ and } \sigma_2(\varepsilon); \]
\[ \lambda \psi. \lambda \nu. \nu \sqcap \nu; \]
\[ (S \upharpoonright \text{TV}) \upharpoonright (S \upharpoonright \text{TV}) \upharpoonright (S \upharpoonright \text{TV}) \]

\[ S \upharpoonright \text{TV} \]

\[ \lambda \varphi_1. \text{ sandy } \bullet \varphi_1 \bullet \text{ kim}; \]
\[ \lambda P.P(k)(s); \]
\[ (S \upharpoonright \text{TV}) \]

(17) a. Mrs. J can’t live in LA and Mr. J in Boston. ¬◊ > ∧

b. Kim didn’t play bingo or Sandy sit at home all evening. ¬ > ∨
Explaining scope anomaly in Gapping

(16)  

(17)  

Rough story:

▶ In examples like (17), the auxiliary is gapped.
▶ Thus, the auxiliary is outside the coordinate structure ‘at LF’.
▶ Auxiliary wide scope is then immediately predicted.
Explaining scope anomaly, specifics

Auxiliary as a scope-taking expression

(18) Robin must discover a solution.

(19) $\lambda \sigma. \sigma(\text{must}); \lambda \mathcal{F}. \Box \mathcal{F}(\text{id}_{et}); S \upharpoonright (S \upharpoonright (VP/VP))$

(where $\text{id}_{et} = \text{def} \lambda P_{et}.P$)
Auxiliaries in English (cont.)

(20) Someone must be present at the meeting.  \((\Box > \exists)\)

(21)

\[
\begin{array}{c}
\lambda \sigma. \sigma(\text{someone});
\forall \text{person};
\end{array}
\]

\[
\begin{array}{c}
\lambda \sigma. \sigma(\text{must});
\lambda \mathcal{T}. \Box \mathcal{T} (\text{id}_{et});
\end{array}
\]

\[
\begin{array}{c}
\text{someone} \bullet \varphi_1 \bullet \text{be} \bullet \text{present};
\Box \forall \text{person}(\lambda x . f (\text{present})(x))(x); S
\end{array}
\]

\[
\begin{array}{c}
\lambda \varphi_1 . \text{someone} \bullet \varphi_1 \bullet \text{be} \bullet \text{present};
\lambda f . \forall \text{person}(\lambda x . f (\text{present})(x))(x); S \upharpoonright (\text{VP/VP})
\end{array}
\]

\[
\begin{array}{c}
\lambda \varphi_1 . \text{someone} \bullet \varphi_1 \bullet \text{be} \bullet \text{present};
\forall \text{person}(\lambda x . f (\text{present})(x)); S
\end{array}
\]

\[
\begin{array}{c}
\lambda \sigma. \sigma(\text{must});
\lambda \mathcal{T}. \Box \mathcal{T} (\text{id}_{et});
\end{array}
\]

\[
\begin{array}{c}
\text{someone} \bullet \varphi_1 \bullet \text{be} \bullet \text{present};
\square \forall \text{person}(\lambda x . \text{present}(x))(x); S
\end{array}
\]
Deriving the VP/VP entry

(22) can’t; λPλx.¬◊P(x); VP/VP
Deriving the VP/VP entry

(22)  can’t; $\lambda P \lambda x. \neg \Diamond P(x); \text{VP/VP}$

(23)  $\varphi_1; x; \text{NP}$

<table>
<thead>
<tr>
<th>Rule</th>
<th>Derivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{\varphi_2; g; \text{VP/VP}^2}{\varphi_2 \bullet \varphi_3; g(P); \text{VP}}$</td>
<td>$\varphi_3; P; \text{VP}^3$</td>
</tr>
<tr>
<td>$\varphi_1 \bullet \varphi_2 \bullet \varphi_3; g(P)(x); S$</td>
<td>$\text{VP}$</td>
</tr>
<tr>
<td>$\lambda \sigma_0. \sigma_0(\text{can’t});$</td>
<td>$\varphi_1 \bullet \varphi_2 \bullet \varphi_3; g(P)(x); S$</td>
</tr>
<tr>
<td>$\lambda \mathcal{F}. \neg \Diamond \mathcal{F}(\text{id}_{et});$</td>
<td>$\varphi_2 \bullet \varphi_3; g(P)(x); S$</td>
</tr>
<tr>
<td>$\text{S}(\text{S</td>
<td>VP/VP})$</td>
</tr>
</tbody>
</table>
Auxiliary Gapping, auxiliary wide-scope

(24) \[ \begin{array}{c}
\varphi_1; \quad \text{eat} \bullet \text{steak}; \\
\left[ \begin{array}{c}
\varphi_1; \\
f; \text{VP/VP}
\end{array} \right] \end{array} \]

\[ \begin{array}{c}
\varphi_1 \bullet \text{eat} \bullet \text{steak}; \\
j; \text{NP} \\
f(eat(s)); \\
\text{VP}
\end{array} \]

\[ \begin{array}{c}
\text{john;} \\
f(eat(s)(j)); \\
S \end{array} \]

\[ \begin{array}{c}
\lambda \varphi_1. \text{john} \bullet \varphi_1 \bullet \text{eat} \bullet \text{steak}; \\
\lambda f.f(eat(s))(j); \\
S \upharpoonright (\text{VP/VP})
\end{array} \]

\[ \begin{array}{c}
\lambda \varphi_0. \text{john} \bullet \varphi_0 \bullet \text{eat} \bullet \text{steak} \bullet \text{and} \bullet \text{mary} \bullet \varepsilon \bullet \text{eat} \bullet \text{pizza}; \\
\lambda f.f(eat(s))(j) \sqcap \lambda g.g(eat(p))(m); \\
S \upharpoonright (\text{VP/VP})\]

\[ \begin{array}{c}
\lambda \varphi_0. \text{john} \bullet \varphi_0 \bullet \text{eat} \bullet \text{steak} \bullet \text{and} \bullet \text{mary} \bullet \varepsilon \bullet \text{eat} \bullet \text{pizza}; \\
\lambda f.f(eat(s))(j) \sqcap \lambda g.g(eat(p))(m); \\
S \upharpoonright (\text{VP/VP})\]
(25)

\[ \lambda \varphi_0. \text{john} \cdot \varphi_0 \cdot \text{eat} \cdot \text{steak} \cdot \text{and} \cdot \text{mary} \cdot \varepsilon \cdot \text{eat} \cdot \text{pizza}; \]

\[ \lambda f.f(\text{eat}(s))(j) \sqcap \lambda g.G(\text{eat}(p))(m); \]

\[ S \upharpoonright (S \upharpoonright (\text{VP}/\text{VP})) \]

\[ \neg \Diamond [\text{eat}(s)(j) \land \text{eat}(p)(m)]; S \]
Summary

Hybrid TLG

- \( \approx \) Lambek calculus + \( \lambda \) grammar

- enables simple analyses of complex linguistic phenomena, capturing potentially interesting empirical generalizations

- can, at a certain level of abstraction, be thought of as a formalization of the ‘movement-based’ syntax-semantics interface

Caveat: Many (supposedly) central properties of mainstream syntax are not part of Hybrid TLG.

- E.g.: island constraints

But note that the exact status of island constraints is unclear in the minimalist program also (cf. [Boeckx, 2012]).
Summary

Hybrid TLG

- ≈ Lambek calculus + λ grammar
- enables simple analyses of complex linguistic phenomena, capturing potentially interesting empirical generalizations
- can, at a certain level of abstraction, be thought of as a formalization of the ‘movement-based’ syntax-semantics interface

 indefinitely Caveat: Many (supposedly) central properties of mainstream syntax are not part of Hybrid TLG.
- E.g.: island constraints
- But note that the exact status of island constraints is unclear in the minimalist program also (cf. [Boeckx, 2012]).
Open questions

- Comparison with some recent HPSG work
  - complex combinatoric component (Hybrid TLG) vs. underspecification (HPSG)

- Formal properties of Hybrid TLG
  - decidability
  - relations to other variants of TLG

- Parsing
Comparison with HPSG using underspecification

Some responses to the Hybrid TLG analyses of coordination from the HPSG community:

- [Yatabe and Tam, 2017] on NCC, building on and extending [Yatabe, 2001]'s earlier work
- [Park et al., 2018] propose an analysis of Gapping in Lexical Resource Semantics addressing the scope anomaly problem

Both of these proposals crucially involve underspecification.
Comparison with HPSG using underspecification (cont.)

[Park et al., 2018]'s analysis of Gapping

```
S
[EXCONT 0]

2 ≪ α
```

```
S
[EXCONT 1]

INCONT 2

John can’t live in LA
```

```
S
[EXCONT 3]

4 ≪ β
```

```
Coord
[EXCONT 3]α ∧ β

and
```

```
S
[EXCONT 4]

and

Mary in New York
```
Comparison with HPSG using underspecification (cont.)

Some observations:

- [Park et al., 2018]’s HPSG analysis:
  - surface-oriented syntax
  - underspecification in semantics

- Hybrid TLG:
  - rigid mapping from (underlying) syntax to semantics
  - mapping from underlying syntax to surface syntax is somewhat complex

Some questions:

- What's the relationship between the two research strategies?
- Any reason to prefer one over the other?
Comparison with HPSG using underspecification (cont.)

Some observations:

▶ [Park et al., 2018]’s HPSG analysis:
  ▶ surface-oriented syntax
  ▶ underspecification in semantics

▶ Hybrid TLG:
  ▶ rigid mapping from (underlying) syntax to semantics
  ▶ mapping from underlying syntax to surface syntax is somewhat complex

Some questions:

▶ What’s the relationship between the two research strategies?
▶ Any reason to prefer one over the other?
Some known and unknown properties of Hybrid TLG

Some recent work investigating the formal properties of Hybrid TLG:

- Richard Moot’s work on translating Hybrid TLG to an implicative fragment of linear logic with first-order quantifiers (MILL1) [Moot, 2014]
  - proof normalization
  - proof search in Hybrid TLG is decidable
  - parsing with Hybrid TLG is NP complete

- Chris Worth, Jordan Needle and Carl Pollard’s ongoing work on embedding Hybrid TLG in Linear Categorial Grammar [Worth, 2016] (continued by ongoing work by Needle/Pollard)

This work may shed light on the relationship between Hybrid TLG and $\lambda$ grammars.
Some known and unknown properties of Hybrid TLG

Some recent work investigating the formal properties of Hybrid TLG:

▶ Richard Moot’s work on translating Hybrid TLG to an implicative fragment of linear logic with first-order quantifiers (MILL1) [Moot, 2014]
  ▶ proof normalization
  ▶ proof search in Hybrid TLG is decidable
  ▶ parsing with Hybrid TLG is NP complete

▶ Chris Worth, Jordan Needle and Carl Pollard’s ongoing work on embedding Hybrid TLG in Linear Categorial Grammar [Worth, 2016] (continued by ongoing work by Needle/Pollard)
  ▶ This work may shed light on the relationship between Hybrid TLG and $\lambda$ grammars.
Some known and unknown properties of Hybrid TLG

- It would be illuminating to study the formal properties of the proof theory of Hybrid TLG itself.
- This is largely left for future work.

Open questions:

- Proof normalization in natural deduction?
- Cut elimination?
- Decidability?
Parser for Hybrid TLG: LinearOne [Moot, 2014, Moot, 2015]

- Theorem prover for MILL1
- Implemented in SWI Prolog
- Available at https://github.com/RichardMoot/LinearOne
- Works as a parser for Hybrid TLG due to the one-to-one correspondence between proofs in first-order linear logic and derivations in Hybrid TLG
Wide coverage parsing?

Should we care? There are already several excellent CCG parsers employing state-of-the-art NLP techniques, e.g.,

- C&C parser [Clark and Curran, 2004]
- EasyCCG [Lewis and Steedman, 2014]
- depCCG [Yoshikawa et al., 2017]
Wide coverage parsing?

Should we care? There are already several excellent CCG parsers employing state-of-the-art NLP techniques, e.g.,

- C&C parser [Clark and Curran, 2004]
- EasyCCG [Lewis and Steedman, 2014]
- depCCG [Yoshikawa et al., 2017]

Some reasons we may want to care about robust parsing in TLG:

- In a sense, TLG is closer to mainstream Chomskian syntax than CCG, due to its less surface-oriented nature.
- So, whether (or to what extent) large-scale parsing is possible with TLG is an inherently interesting question.
- It may give us some ideas about how one might go about implementing parsers based on movement-based theories of grammar.
Challenge for wide coverage parsing

- Hypothetical reasoning is a very general rule. **Note**: CCG effectively decomposes hypothetical reasoning to a set of local composition rules.

- Some attempts in the TLG literature:
  - supertagging [Moot, 2017]
  - proof net parsing [Morrill, 2010]
  - using word vectors to eliminate structural ambiguity [Moot, 2017]

- Incorporating these results in Hybrid TLG parsing is another major remaining issue.
Conclusion

- Hybrid TLG is a new version of Type-Logical Grammar which
  - has wide coverage on complex empirical phenomena
  - models the notion of ‘syntactic movement’ with lambda binding in
    the prosodic component
- Computational and formal properties of Hybrid TLG still remain
  to be explored.
  - some basic formal properties (e.g. decidability, parsing complexity)
    need to be clarified.
  - robust parser?
Acknowledgement

- The present research is supported by JSPS KAKENHI Grant Number 15K16732.
Appendix A: Prosodic and semantic types

Mapping from syntactic to prosodic types

Directional type:

For any directional type $A$,

$\textbf{Pros}(A) = \text{st}$  
(with $\text{st}$ for ‘string’)
Appendix A: Prosodic and semantic types

Mapping from syntactic to prosodic types

Directional type:

For any directional type $A$,

- $\text{Pros}(A) = \text{st}$
  
  (with st for ‘string’)

Non-directional complex type:

For any complex syntactic type $A \upharpoonright B$,

- $\text{Pros}(A \upharpoonright B) = \text{Pros}(B) \to \text{Pros}(A)$

Note:

- For the mapping from syntactic types to prosodic types, only $\upharpoonright$ is effectively interpreted as functional.
Appendix A: Prosodic and semantic types

Mapping from syntactic to semantic types

Atomic types:

- $\text{Sem}(\text{NP}) = e$
- $\text{Sem}(\text{S}) = t$
- $\text{Sem}(\text{N}) = e \rightarrow t$

Complex types:

$\text{Sem}(A/B) = \text{Sem}(B\backslash A) = \text{Sem}(A\upharpoonright B) = \text{Sem}(B) \rightarrow \text{Sem}(A)$.

- The semantic types are ‘read off’ from syntactic types by taking all of $\mathcal{S}, \mathcal{L}, \mathcal{R}$ to be type constructors for functions.
Appendix A: Prosodic and semantic types

Some examples

- $\text{Sem}(S \upharpoonright \text{NP}) = e \rightarrow t$
- $\text{Pros}(S \upharpoonright \text{NP}) = \text{st} \rightarrow \text{st}$

- $\text{Sem}(S \upharpoonright (S \upharpoonright \text{NP})) = (e \rightarrow t) \rightarrow t$
- $\text{Pros}(S \upharpoonright (S \upharpoonright \text{NP})) = (\text{st} \rightarrow \text{st}) \rightarrow \text{st}$
Appendix A: Prosodic and semantic types

Some examples

- \( \text{Sem}(S \upharpoonright \text{NP}) = e \rightarrow t \)
- \( \text{Pros}(S \upharpoonright \text{NP}) = \text{st} \rightarrow \text{st} \)

- \( \text{Sem}(S \upharpoonright (S \upharpoonright \text{NP})) = (e \rightarrow t) \rightarrow t \)
- \( \text{Pros}(S \upharpoonright (S \upharpoonright \text{NP})) = (\text{st} \rightarrow \text{st}) \rightarrow \text{st} \)

- \( \text{Sem}(S / \text{NP}) = e \rightarrow t \)
- \( \text{Pros}(S / \text{NP}) = \text{st} \)

- \( \text{Sem}(S \upharpoonright (S / \text{NP})) = (e \rightarrow t) \rightarrow t \)
- \( \text{Pros}(S \upharpoonright (S / \text{NP})) = \text{st} \rightarrow \text{st} \)
Appendix B: (A bit) more on decidability

- Proving Cut elimination seems relatively straightforward.
- The tricky part is that proofs come with term labelling (where these terms are not proof terms).
- And it is not immediately clear whether there is a deterministic algorithm for specifying all the possible terms for each node of the proof tree when going from root to leaves.
- Cf. [Oehrle, 1994]:

  Does the proof [of decidability of LP] carry over directly to the system of labeled deduction considered here? There is one sticking point: the label of the left-hand premise of the rule \( L \rightarrow \) is not determined by the sub-terms of its conclusion.

\[
L \rightarrow \\
\Gamma, t: A \rightarrow B \leadsto t', \Delta \vdash Y \cup F_A(A) v: C \leadsto v' \\
\Gamma \cup X u: A \leadsto u' \ (tu): B \leadsto (t'u'), \Delta \vdash Y \cup F_A(A) v: C \leadsto v'
\]
Appendix C: $\lambda$ grammars and coordination


Coordination

(26) a. John walks and talks.
    b. John bought and ate the fish.

(27) John sent [Sandy a letter] and [Jane a postcard].

(28) [John bought], and [Sandy sold], some very expensive books.
Appendix C: $\lambda$ grammars and coordination (cont.)

Coordination of $\text{st} \rightarrow \text{st}$ functions:

(29) $\lambda \varphi. \varphi \bullet \text{walks; walk; } S \upharpoonright \text{NP}$
    $\lambda \varphi. \varphi \bullet \text{talks; talk; } S \upharpoonright \text{NP}$

(30) $\lambda \sigma_1 \lambda \sigma_2 \lambda \varphi. \varphi \bullet \sigma_1(\varepsilon) \bullet \text{and} \bullet \sigma_2(\varepsilon); \lambda P \lambda Q. P \sqcap Q;
    (S \upharpoonright \text{NP}) \upharpoonright (S \upharpoonright \text{NP}) \upharpoonright (S \upharpoonright \text{NP})$

(31) \[ \begin{align*}
    \lambda \sigma_1 \lambda \sigma_2 \lambda \varphi. \varphi & \bullet \sigma_1(\varepsilon) \bullet \text{and} \bullet \sigma_2(\varepsilon); \\
    \lambda \varphi. \varphi & \bullet \text{walks; } \lambda P \lambda Q. P \sqcap Q; \\
    \text{walk; } S \upharpoonright \text{NP} & (S \upharpoonright \text{NP}) \upharpoonright (S \upharpoonright \text{NP}) \upharpoonright (S \upharpoonright \text{NP})
\end{align*} \]

\[ \begin{align*}
    \lambda \sigma_2 \lambda \varphi. \varphi & \bullet \text{walks} \bullet \text{and} \bullet \sigma_2(\varepsilon); \\
    \lambda Q. \text{walk} & \sqcap Q; \\
    (S \upharpoonright \text{NP}) \upharpoonright (S \upharpoonright \text{NP})
\end{align*} \]

\[ \begin{align*}
    \lambda \varphi. \varphi & \bullet \text{talks; talk; } S \upharpoonright \text{NP}
\end{align*} \]

\[ \begin{align*}
    \lambda \varphi. \varphi & \bullet \text{walks} \bullet \text{and} \bullet \text{talks; walk} \sqcap \text{talk; } S \upharpoonright \text{NP}
\end{align*} \]
Appendix C: $\lambda$ grammars and coordination (cont.)

(32) 
\[
\begin{align*}
\lambda \varphi_1 \lambda \varphi_2 & \cdot \text{met} \cdot \varphi_1; \quad [\varphi_1; x; \text{NP}]^1 \\
\text{meet}; (S \upharpoonright \text{NP}) \upharpoonright \text{NP} & \\
\text{sandy; } \lambda \varphi_2 \cdot \text{met} \cdot \varphi_1; \\
\text{s; NP} & \\
\text{meet}(x); S \upharpoonright \text{NP} & \\
\text{sandy } \cdot \text{met} \cdot \varphi_1; \text{ meet}(x)(s); S & \\
\lambda \varphi_1 \cdot \text{sandy } \cdot \text{met} \cdot \varphi_1; \lambda x \cdot \text{meet}(x)(s); S \upharpoonright \text{NP} & \end{align*}
\]

(33) *John [[walks] and [Sandy met ___]].

(34) *[[Sandy met] and [walks]] John.

(35) *John bought and ate the fish.
‘John bought the fish and the fish ate John.’
References


Lambek, J. (1958).
The mathematics of sentence structure.


