Lambek Categorial Grammars
Day 1

Stepan Kuznetsov

Steklov Mathematical Institute, RAS
for ESSLLI '15 in Barcelona

August 3, 2015

Course Outline

- Lambek calculus & categorial grammars

Course Outline

- Lambek calculus & categorial grammars
- Montague semantics

Course Outline

- Lambek calculus & categorial grammars
- Montague semantics
- Lambek grammars are context-free

Course Outline

- Lambek calculus & categorial grammars
- Montague semantics
- Lambek grammars are context-free
- Extensions of the formalism

Course Outline

- Lambek calculus & categorial grammars
- Montague semantics
- Lambek grammars are context-free
- Extensions of the formalism
- Interpretation of the Lambek calculus on the algebra of formal languages; completeness results

Course Outline

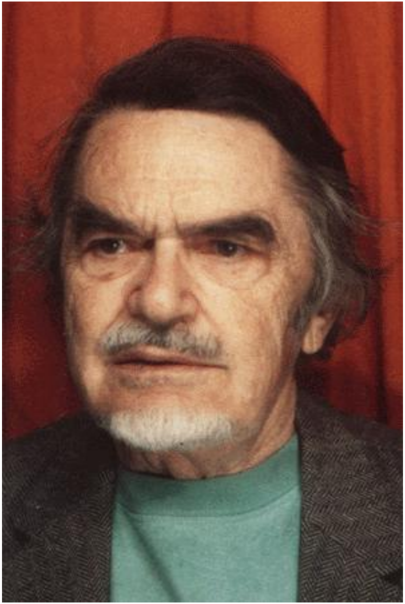
- Lambek calculus & categorial grammars
- Montague semantics
- Lambek grammars are context-free
- Extensions of the formalism
- Interpretation of the Lambek calculus on the algebra of formal languages; completeness results
- Algorithmic complexity

Course Outline

- Lambek calculus & categorial grammars
- Montague semantics
- Lambek grammars are context-free
- Extensions of the formalism
- Interpretation of the Lambek calculus on the algebra of formal languages; completeness results
- Algorithmic complexity
- Natural language examples

Kazimierz Ajdukiewicz

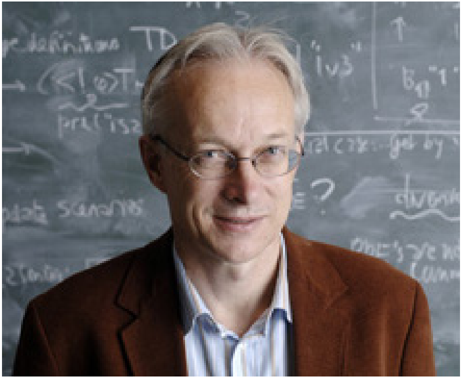
Joachim Lambek



Richard Montague

Haskell Curry & William Howard

Johan van Benthem



Mati Pentus

The λ-calculus

The λ-calculus

Types & Terms

Types:

1. Basic types.
2. If A and B are types, then $(A \to B)$ is also a type.

Types & Terms

Types:

1. Basic types.
2. If A and B are types, then $(A \to B)$ is also a type.

Terms: each term has a type $(t : A)$.

1. Constants (a countable set for each type)
2. Variables (a countable set for each type)
3. *Application*: if $t : A \to B$ and $s : A$, then $(t \cdot s) : B$.
4. λ*-abstraction:* if $t : B$, then $\lambda x^A.t : (A \to B)$.

Reductions

β-reduction

$$(\lambda x.u)v \to_\beta u[x := v]$$

Reductions

β-reduction

$$(\lambda x.u)v \to_\beta u[x := v]$$

η-reduction

$$\lambda x.(fx) \to_\eta f$$

$(x \notin \mathrm{FV}(f))$

Currying (Emulating Multiargument Functions)

Use $A \rightarrow (B \rightarrow C)$ instead of $(A \times B) \rightarrow C$.

Montague Semantics: Compositionality (Idea)

The semantic value assigned to a sentence is a composition of semantic values assigned to its words.

Montague Semantics: Compositionality (Idea)

The semantic value assigned to a sentence is a composition of semantic values assigned to its words.

Useful, but generally invalid (consider idiomatic phrases).

Montague Semantics: Compositionality (Idea)

The semantic value assigned to a sentence is a composition of semantic values assigned to its words.

Useful, but generally invalid (consider idiomatic phrases).

Semantic values are λ-terms.

John loves Mary.

John loves Mary.

Semantic value: LOVE(MARY)(JOHN).

John loves Mary.

Semantic value: $\text{LOVE}(\text{MARY})(\text{JOHN})$.

$\text{JOHN} : e$, $\text{MARY} : e$, $\text{LOVE} : e \to (e \to t)$

Narcissus loves himself.

Narcissus loves himself.

Semantic value: LOVE(NARCISSUS)(NARCISSUS).

Narcissus loves himself.

Semantic value: $\textsc{love}(\textsc{narcissus})(\textsc{narcissus})$.

$\textsc{narcissus} : e$, $\textsc{love} : e \to (e \to t)$

Narcissus loves himself.

Semantic value: $\textsc{love}(\textsc{narcissus})(\textsc{narcissus})$.

$\textsc{narcissus} : e$, $\textsc{love} : e \to (e \to t)$
$\sigma(\text{himself}) : (e \to (e \to t)) \to (e \to t)$.

Narcissus loves himself.

Semantic value: $\textsc{love}(\textsc{narcissus})(\textsc{narcissus})$.

$\textsc{narcissus} : e$, $\textsc{love} : e \to (e \to t)$
$\sigma(\text{himself}) : (e \to (e \to t)) \to (e \to t)$.

$\sigma(\text{himself}) = \lambda y.\lambda x.y(x)(x)$

Narcissus loves himself.

Semantic value: $\textsc{love}(\textsc{narcissus})(\textsc{narcissus})$.

$\textsc{narcissus} : e$, $\textsc{love} : e \to (e \to t)$
$\sigma(\text{himself}) : (e \to (e \to t)) \to (e \to t)$.

$\sigma(\text{himself}) = \lambda y.\lambda x.y(x)(x)$

$(\lambda y.\lambda x.y(x)(x))(\textsc{love})(\textsc{narcissus}) \to_\beta$
$(\lambda x.\textsc{love}(x)(x))\textsc{narcissus} \to_\beta \textsc{love}(\textsc{narcissus})(\textsc{narcissus})$

The Curry – Howard Correspondence

Theorem

Let A, B_1, ..., B_k be types. Then there exists such a term $u : A$ with free variables $x_1^{B_1}$, ..., $x_k^{B_k}$ iff $B_1, \dots, B_k \vdash A$ is intuitionistically valid.

$\mathrm{Int}_{\to}$ (Natural Deduction)

$$\Gamma, A \vdash A$$

$$\frac{\Gamma, B \vdash A}{\Gamma \vdash (A \to B)}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash (A \to B)}{\Gamma \vdash B}$$

$\mathrm{Int}_{\rightarrow}$ (Natural Deduction)

$$\Gamma, A \vdash A$$

$$\frac{\Gamma, B \vdash A}{\Gamma \vdash (A \rightarrow B)} \qquad x : B, u : A \rightsquigarrow \lambda x.u : (A \rightarrow B)$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash (A \rightarrow B)}{\Gamma \vdash B} \qquad u : (A \rightarrow B), v : A \rightsquigarrow (uv) : B$$

The Calculus for Syntax: Going Substructural

Describe syntax using a logical calculus (a fragment of $\mathrm{Int}_{\rightarrow}$) that will yield semantic values.

The Calculus for Syntax: Going Substructural

Describe syntax using a logical calculus (a fragment of $\mathrm{Int}_{\rightarrow}$) that will yield semantic values.

- non-commutative: instead of $(A \rightarrow B)$ we now have $(A \setminus B)$ and $(B \,/\, A)$

The Calculus for Syntax: Going Substructural

Describe syntax using a logical calculus (a fragment of $\mathrm{Int}_{\rightarrow}$) that will yield semantic values.

- non-commutative: instead of $(A \rightarrow B)$ we now have $(A \setminus B)$ and $(B \mathbin{/} A)$
- no contraction (A, A is different from A)

The Calculus for Syntax: Going Substructural

Describe syntax using a logical calculus (a fragment of $\mathrm{Int}_{\rightarrow}$) that will yield semantic values.

- non-commutative: instead of $(A \rightarrow B)$ we now have $(A \setminus B)$ and $(B \,/\, A)$
- no contraction (A, A is different from A)
- no weakening (not allowed to add garbage into the sentence)

Product-Free Lambek Calculus (Natural Deduction)

$$A \to A$$

$$\frac{A\Gamma \to B}{\Gamma \to A \backslash B}\text{, } \Gamma \text{ is not empty} \qquad \frac{\Gamma A \to B}{\Gamma \to B / A}\text{, } \Gamma \text{ is not empty}$$

$$\frac{\Gamma \to A \quad \Delta \to A \backslash B}{\Gamma\Delta \to B} \qquad \frac{\Delta \to B / A \quad \Gamma \to A}{\Delta\Gamma \to B}$$

Why Non-Empty Left-Hand Side?

BOOK : n

INTERESTING : $(n \,/\, n)$

VERY : $((n \,/\, n) \,/ (n \,/\, n))$

Why Non-Empty Left-Hand Side?

$\text{BOOK} : n$
$\text{INTERESTING} : (n \,/\, n)$
$\text{VERY} : ((n \,/\, n) \,/(n \,/\, n))$

$\mathrm{L} \vdash ((n \,/\, n) \,/(n \,/\, n)) \; (n \,/\, n) \; n \to n$	"very interesting book"
$\mathrm{L} \nvdash ((n \,/\, n) \,/(n \,/\, n)) \; n \to n$	"very book"

Ajdukiewicz – Bar-Hillel Calculus

Note: in our examples we actually didn't use λ-abstraction.
We can use a calculus that only regulates the order of applications.

Ajdukiewicz – Bar-Hillel Calculus

Note: in our examples we actually didn't use λ-abstraction.
We can use a calculus that only regulates the order of applications.

$$A \to A$$

$$\frac{\Gamma \to A \quad \Delta \to A \setminus B}{\Gamma\Delta \to B} \qquad \frac{\Delta \to B \,/\, A \quad \Gamma \to A}{\Delta\Gamma \to B}$$

Categorial Grammar

Σ is a finite alphabet, $H \in \mathrm{Tp}$.

Categorial vocabulary: $\mathcal{D} \subset \mathrm{Tp} \times \Sigma \times \mathrm{Tm}_\lambda$.

$w = a_1 \dots a_n \in \Sigma^+$
$\langle A_i, a_i, u_i \rangle \in \mathcal{D}$
$\mathrm{L} \vdash A_1 \dots A_n \to H$ (or: $\mathrm{AB} \vdash \dots$).

Thanks and see you tomorrow