Lambek Categorial Grammars Day 2

Stepan Kuznetsov

Steklov Mathematical Institute, RAS for ESSLLI '15 in Barcelona

August 4, 2015

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 の�?

THE : $(e \rightarrow t) \rightarrow e$ "if a predicate has exactly one object on which it is true, return this object" (partial function)

THE : $(e \rightarrow t) \rightarrow e$ "if a predicate has exactly one object on which it is true, return this object" (partial function)

```
AND: t \rightarrow (t \rightarrow t), OR: t \rightarrow (t \rightarrow t)
(written u \& v, u \lor v)
```

THE : $(e \rightarrow t) \rightarrow e$ "if a predicate has exactly one object on which it is true, return this object" (partial function)

$$ext{AND}: t
ightarrow (t
ightarrow t), ext{ OR}: t
ightarrow (t
ightarrow t)$$
 (written $u \And v, \ u \lor v$)

Mapping syntactic types to semantic ones: noun $n \mapsto (e \to t)$ noun phrase $np \mapsto e$ sentence $s \mapsto t$

The Cut Rule

$\frac{\Gamma \to A \quad \Gamma A \Delta \to C}{\Gamma \Pi \Delta \to C}$

(corresponds to substitution of λ -terms)

 $\begin{array}{ccc} \text{The} & \text{book} & \text{fell.} \\ (np / n) & n & (np \setminus s) & \rightarrow s \\ \text{THE} & \text{BOOK} & \text{FALL} \end{array}$

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

Thebookfell.
$$(np / n)$$
 n $(np \setminus s)$ $\rightarrow s$ THEBOOKFALL

thebookwhichfell
$$(np/n)$$
 n $(n \setminus n)/(np \setminus s)$ $np \setminus s \to np$ THEBOOK $\lambda x^{e \to t} . \lambda y^{e \to t} . \lambda z^e . (x(z) \& y(z))$ FALL

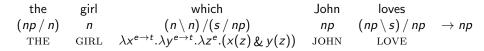
◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

Semantic value: THE $\lambda z.(BOOK(z) \& FALL(z))$

JohnlovesMary.np $(np \setminus s) / np$ np $\rightarrow s$ JOHNLOVEMARY

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

JohnlovesMary.np $(np \setminus s) / np$ np $\rightarrow s$ JOHNLOVEMARY



Semantic value: THE $\lambda z.(GIRL(z) \& LOVE(z)(JOHN))$.

JohnlovesMary.np $(np \setminus s) / np$ np $\rightarrow s$ JOHNLOVEMARY

the girl which John loves

$$(np/n)$$
 n $(n \setminus n)/(s/np)$ np $(np \setminus s)/np \to np$
THE GIRL $\lambda x^{e \to t} . \lambda y^{e \to t} . \lambda z^{e} . (x(z) \& y(z))$ JOHN LOVE

Semantic value: THE $\lambda z.(GIRL(z) \& LOVE(z)(JOHN))$.

Actually, $L \vdash np (np \setminus s) / np \rightarrow s / np$ ("John loves" reduces to s / np). Not derivable in AB.

Dependent Clauses: Limitations

John read *Ulysses* a month ago. the book which John read [] a month ago

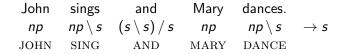
◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

Dependent Clauses: Limitations

John read *Ulysses* a month ago. the book which John read [] a month ago

John sings and loves Mary. *the girl which John sings and loves

Coordination



Semantic value: SING(JOHN) & DANCE(MARY).

Coordination: Going Polymorphic

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

Semantic value: SING(JOHN) ∨ DANCE(JOHN).

Coordination: Going Polymorphic

In general:

$$T = T_1 \rightarrow \ldots \rightarrow (T_k \rightarrow t)$$

$$\mathfrak{or}_T: T \to (T \to T)$$

$$\mathfrak{or}_{T} = \lambda x^{T} \cdot \lambda y^{T} \cdot \lambda z_{1}^{T_{1}} \ldots \lambda z_{k}^{T_{k}} \cdot (xz_{1} \ldots z_{k} \vee yz_{1} \ldots z_{k})$$

◆□ ▶ ◆□ ▶ ◆三 ▶ ◆□ ▶ ◆□ ▶

Coordination Between Noun Phrases

John and Pete sing.

John or Pete sings.

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

Coordination Between Noun Phrases: AND

$$np^* \longrightarrow$$
 plural noun phrase.
 $np^* \mapsto (e \rightarrow t)$ (in other words, $\mathcal{P}(e)$)

John	and	Pete	sing.	
np	$(\textit{np} \setminus \textit{np}^*) / \textit{np}$	np	$\textit{np}^* \setminus \textit{s}$	ightarrow s
JOHN	PAIR	PETE	SING^*	

$$\{x^{e}, y^{e}\}^{e \to t} = \lambda z^{e} . (z = x \lor z = y)$$

PAIR = $\lambda x^{e} . \lambda y^{e} . \{x, y\}$
SING^{*} = $\lambda w^{e \to t} . \forall^{(e \to t) \to t} \lambda x^{e} . (w(x) \Rightarrow \text{SING}(x)).$

◆□ ▶ < 圖 ▶ < 圖 ▶ < 圖 ▶ < 圖 • 의 Q @</p>

Coordination Between Noun Phrases: AND

$$np^* \longrightarrow$$
 plural noun phrase.
 $np^* \mapsto (e \rightarrow t)$ (in other words, $\mathcal{P}(e)$)

John	and	Pete	sing.	
np	$(\textit{np} \setminus \textit{np}^*) / \textit{np}$	np	$\textit{np}^* \setminus \textit{s}$	ightarrow s
JOHN	PAIR	PETE	SING^*	

$$\{x^{e}, y^{e}\}^{e \to t} = \lambda z^{e} . (z = x \lor z = y)$$

PAIR = $\lambda x^{e} . \lambda y^{e} . \{x, y\}$
SING^{*} = $\lambda w^{e \to t} . \forall^{(e \to t) \to t} \lambda x^{e} . (w(x) \Rightarrow \text{SING}(x)).$

This solution needs some first-order logic (\forall) ...

▲□▶▲圖▶▲圖▶▲圖▶ 圖 めへぐ

Coordination Between Noun Phrases: AND

$$np^* \longrightarrow ext{plural noun phrase.} \ np^* \longmapsto (e o t) ext{ (in other words, } \mathcal{P}(e))$$

John	and	Pete	sing.	
np	$(\textit{np} \setminus \textit{np}^*) / \textit{np}$	np	$\textit{np}^* \setminus \textit{s}$	ightarrow s
JOHN	PAIR	PETE	SING^*	

$$\{x^{e}, y^{e}\}^{e \to t} = \lambda z^{e} . (z = x \lor z = y)$$

PAIR = $\lambda x^{e} . \lambda y^{e} . \{x, y\}$
SING^{*} = $\lambda w^{e \to t} . \forall^{(e \to t) \to t} \lambda x^{e} . (w(x) \Rightarrow \text{SING}(x)).$

This solution needs some first-order logic (\forall)... and doesn't work for OR.

・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・
 ・

 $L \vdash p \rightarrow q / (p \setminus q)$



$$L \vdash p \rightarrow q / (p \setminus q)$$

$$x^{\sigma(p)} \mapsto \lambda f^{\sigma(p) o \sigma(q)} . f(x)$$

$$L \vdash p \rightarrow q / (p \setminus q)$$

$$x^{\sigma(p)} \mapsto \lambda f^{\sigma(p) \to \sigma(q)} . f(x)$$

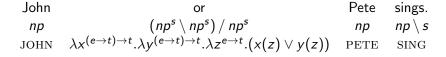
Now we can transform x : e into $\lambda f^{e \to t} f(x) : (e \to t) \to t$.

$$L \vdash p \rightarrow q / (p \setminus q)$$

$$x^{\sigma(p)} \mapsto \lambda f^{\sigma(p) \to \sigma(q)} . f(x)$$

Now we can transform x : e into $\lambda f^{e \to t} f(x) : (e \to t) \to t$. Again, this doesn't work in AB. Coordination of Noun Phrases: OR

$$np^{s} = s / (np \setminus s).$$



Coordination of Noun Phrases: OR

$$np^s = s / (np \setminus s).$$

JohnorPetesings.np $(np^s \setminus np^s) / np^s$ np $np \setminus s$ JOHN $\lambda x^{(e \to t) \to t} . \lambda y^{(e \to t) \to t} . \lambda z^{e \to t} . (x(z) \lor y(z))$ PETESING

Perform type raising for "John" and "Pete": John or Pete np^{s} $(np^{s} \setminus np^{s}) / np^{s}$ $np^{s} \to np^{s}$ $\lambda f.f(JOHN)$ $\lambda x. \lambda y. \lambda z. (x(z) \lor y(z))$ $\lambda f.f(PETE)$

Coordination of Noun Phrases: OR

$$np^s = s / (np \setminus s).$$

JohnorPetesings.np $(np^s \setminus np^s) / np^s$ np $np \setminus s$ JOHN $\lambda x^{(e \to t) \to t} . \lambda y^{(e \to t) \to t} . \lambda z^{e \to t} . (x(z) \lor y(z))$ PETESING

Perform type raising for "John" and "Pete": John or Pete np^{s} $(np^{s} \setminus np^{s}) / np^{s}$ $np^{s} \rightarrow np^{s}$ $\lambda f.f(\text{JOHN}) \quad \lambda x. \lambda y. \lambda z. (x(z) \lor y(z)) \quad \lambda f.f(\text{PETE})$

John or Pete sings. $s/(np \setminus s)$ $np \setminus s \rightarrow s$ $\lambda f.(f(JOHN) \lor f(PETE))$ SING Final semantic value: SING(JOHN) \lor SING(PETE). Gentzen-style Product-Free Lambek Calculus

 $A \rightarrow A$

$$\frac{A\Pi \to B}{\Pi \to A \setminus B} \text{ , } \Pi \text{ is not empty} \qquad \frac{\Pi A \to B}{\Pi \to B / A} \text{ , } \Pi \text{ is not empty}$$

$$\frac{\Pi \to A \quad \Gamma B \Delta \to C}{\Gamma \Pi (A \setminus B) \Delta \to C} \qquad \frac{\Pi \to A \quad \Gamma B \Delta \to C}{\Gamma (B / A) \Pi \Delta \to C}$$

$$\frac{\Pi \to A \quad \Gamma A \Delta \to C}{\Gamma \Pi \Delta \to C} \text{ (cut)}$$

<□ > < @ > < E > < E > E のQ @

Cut Elimination

Theorem (Lambek '58)

Every sequent derivable in L can be derived without using the (cut) rule.

◆□▶ ◆□▶ ◆臣▶ ◆臣▶ 臣 のへぐ

Cut Elimination

Theorem (Lambek '58)

Every sequent derivable in ${\rm L}$ can be derived without using the (cut) rule.

Proof.

Routine induction on the complexity of the sequent.

Cut Elimination

Theorem (Lambek '58)

Every sequent derivable in ${\rm L}$ can be derived without using the (cut) rule.

Proof.

Routine induction on the complexity of the sequent.

Benefits:

- Subformula property
- Decidability (more precisely, the derivability problem for L belongs to NP)

Lambek Grammars and Context-Free Grammars

Theorem (Gaifman '61)

Every context-free language without the empty word can be generated by an AB-grammar.

Theorem (Pentus '92)

Every language generated by a Lambek categorial grammar is context-free.

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQ@

Greibach Normal Form

A context-free grammar is in Greibach normal from if every rule of this grammar has one of the following forms:

- $\blacktriangleright A \rightarrow a$
- $A \rightarrow aB$
- $A \rightarrow aBC$

Theorem (Greibach '65)

Every context-free language without the empty word can be generated by a context-free grammar in Greibach normal form.

From Context-Free Grammars to AB-grammars: Proof

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ のQ@

$$N \ni A \rightsquigarrow p_A \in \Pr$$
.

Categorial vocabulary:

 $\begin{array}{ll} A \rightarrow a & \langle p_A, a \rangle \\ A \rightarrow aB & \langle p_A \, / \, p_B, a \rangle \\ A \rightarrow aBC & \langle (p_A \, / \, p_C) \, / \, p_B, a \rangle \end{array}$

 $H = p_S$



http://www.mi.ras.ru/~sk/lehre/esslli2015

