Lambek Categorial Grammars
Day 4

Stepan Kuznetsov

Steklov Mathematical Institute, RAS
for ESSLLI '15 in Barcelona

August 6, 2015

Gentzen-style Lambek Calculus

$$A \to A$$

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

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

$$\frac{\Gamma\, A\, B\, \Delta \to C}{\Gamma\, (A \cdot B)\, \Delta \to C} \qquad \frac{\Gamma \to A \quad \Delta \to B}{\Gamma\, \Delta \to A \cdot B}$$

Gentzen-style Lambek Calculus

$$A \to A$$

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

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

$$\frac{\Gamma\, A\, B\, \Delta \to C}{\Gamma\, (A \cdot B)\, \Delta \to C} \qquad \frac{\Gamma \to A \quad \Delta \to B}{\Gamma\, \Delta \to A \cdot B}$$

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

L-models

$$w\colon \mathrm{Tp} \to \mathcal{P}(\Sigma^+)$$

$$w(A \cdot B) = w(A) \cdot w(B) = \{uv \mid u \in w(A),\ v \in w(B)\}$$
$$w(A \setminus B) = w(A) \setminus w(B) = \{u \in \Sigma^+ \mid (\forall v \in w(A))\, vu \in w(B)\}$$
$$w(B \,/\, A) = w(B) \,/\, w(A) = \{u \in \Sigma^+ \mid (\forall v \in w(A))\, uv \in w(B)\}$$
$$w(A_1 \ldots A_n) = w(A_1) \cdot \ldots \cdot w(A_n)$$

L-models

$$w \colon \mathrm{Tp} \to \mathcal{P}(\Sigma^+)$$

$$\begin{aligned}
&w(A \cdot B) = w(A) \cdot w(B) = \{uv \mid u \in w(A),\ v \in w(B)\}\\
&w(A \setminus B) = w(A) \setminus w(B) = \{u \in \Sigma^+ \mid (\forall v \in w(A))\, vu \in w(B)\}\\
&w(B \,/\, A) = w(B) \,/\, w(A) = \{u \in \Sigma^+ \mid (\forall v \in w(A))\, uv \in w(B)\}\\
&w(A_1 \ldots A_n) = w(A_1) \cdot \ldots \cdot w(A_n)
\end{aligned}$$

$$\Sigma, w \vDash \Gamma \to C \iff w(\Gamma) \subseteq w(C).$$

Completeness Theorem

Theorem (Pentus '95)

$\mathrm{L} \vdash \Gamma \to C \iff (\forall \Sigma, w)\, \Sigma, w \models \Gamma \to C.$

Completeness Proof in the Product-Free Case

(Buszkowski)

$\Sigma = \mathrm{Tp}_m$
$w(A) = \{\Gamma \mid \mathrm{L} \vdash \Gamma \to A\}$

Doesn't Work with the Product

$$(p \cdot q) \in w(p \cdot q)$$
$$(p \cdot q) \notin w(p) \cdot w(q)$$

Open Problem

Formulate an L-complete extension of L with the Kleene iteration (as a unary connective).

More on Grammars

- The trick to add the empty word: $\mathcal{G}[p_S := ((r \setminus r) \setminus ((s \setminus s) \setminus q) \setminus q]$, where $\mathcal{G}$ comes from Greibach normal form in which S does not appear in right-hand sides of the rules.

More on Grammars

- The trick to add the empty word: $\mathcal{G}[p_S := ((r \setminus r) \setminus ((s \setminus s) \setminus q) \setminus q]$, where $\mathcal{G}$ comes from Greibach normal form in which S does not appear in right-hand sides of the rules.
- Lambek grammars with one primitive type: $\mathcal{G}[p_k := \big(p^{k+1} \cdot ((p \setminus (p \setminus p)) \setminus p\big) \cdot p^{n-k+1}\big) \setminus p]_{k=1}^{n}$.

More on Grammars

- The trick to add the empty word: $\mathcal{G}[p_S := ((r \setminus r) \setminus ((s \setminus s) \setminus q) \setminus q]$, where $\mathcal{G}$ comes from Greibach normal form in which S does not appear in right-hand sides of the rules.
- Lambek grammars with one primitive type: $\mathcal{G}[p_k := \left(p^{k+1} \cdot ((p \setminus (p \setminus p)) \setminus p\right) \cdot p^{n-k+1}) \setminus p]_{k=1}^{n}$.
- (Safiullin '07) Every context-free language is generated by a Lambek grammar which assigns a unique type to each letter.

NP-completeness

Theorem (Pentus '03)

The derivability problem for L *is NP-complete.*

NP-completeness: Sketch of Proof

Reduce SAT to derivability in L.

NP-completeness: Sketch of Proof

Reduce SAT to derivability in L.

- Let $c_1 \wedge \ldots \wedge c_m$ be a Boolean formula in conjunctive normal form.

NP-completeness: Sketch of Proof

Reduce $\textsc{sat}$ to derivability in L.

- Let $c_1 \wedge \ldots \wedge c_m$ be a Boolean formula in conjunctive normal form.
- Construct such formulae $E_1(t_1)$, ..., $E_n(t_n)$, and G, that $\mathrm{L} \vdash E_1(t_1) \ldots E_n(t_n) \to G$ iff $\langle t_1, \ldots, t_n \rangle$ is a satisfying assignment for $c_1 \wedge \cdots \wedge c_m$.

NP-completeness: Sketch of Proof

Reduce SAT to derivability in L.

- Let $c_1 \wedge \ldots \wedge c_m$ be a Boolean formula in conjunctive normal form.
- Construct such formulae $E_1(t_1)$, ..., $E_n(t_n)$, and G, that $\mathrm{L} \vdash E_1(t_1) \ldots E_n(t_n) \to G$ iff $\langle t_1, \ldots, t_n \rangle$ is a satisfying assignment for $c_1 \wedge \cdots \wedge c_m$.
- Construct such formulae F_1, ..., F_n, that $\mathrm{L} \vdash F_i \to E_i(t)$ $(t \in \{0, 1\})$ and if $\mathrm{L} \vdash F_1 \ldots F_n \to G$, then $\mathrm{L} \vdash E_1(t_1) \ldots E_n(t_n) \to G$ for some $t_1, \ldots, t_n$.

NP-completeness: Sketch of Proof

Reduce SAT to derivability in L.

- Let $c_1 \wedge \ldots \wedge c_m$ be a Boolean formula in conjunctive normal form.
- Construct such formulae $E_1(t_1)$, …, $E_n(t_n)$, and G, that $\mathrm{L} \vdash E_1(t_1) \ldots E_n(t_n) \to G$ iff $\langle t_1, \ldots, t_n \rangle$ is a satisfying assignment for $c_1 \wedge \cdots \wedge c_m$.
- Construct such formulae F_1, …, F_n, that $\mathrm{L} \vdash F_i \to E_i(t)$ ($t \in \{0, 1\}$) and if $\mathrm{L} \vdash F_1 \ldots F_n \to G$, then $\mathrm{L} \vdash E_1(t_1) \ldots E_n(t_n) \to G$ for some $t_1, \ldots, t_n$.
- $\mathrm{L} \vdash F_1 \ldots F_n \to G$ iff $c_1 \wedge \cdots \wedge c_n$ is satisfiable.

NP-completeness: Sketch of Proof

Reduce SAT to derivability in L.

- Let $c_1 \wedge \ldots \wedge c_m$ be a Boolean formula in conjunctive normal form.
- Construct such formulae $E_1(t_1)$, ..., $E_n(t_n)$, and G, that $\mathrm{L} \vdash E_1(t_1) \ldots E_n(t_n) \to G$ iff $\langle t_1, \ldots, t_n \rangle$ is a satisfying assignment for $c_1 \wedge \cdots \wedge c_m$.
- Construct such formulae F_1, ..., F_n, that $\mathrm{L} \vdash F_i \to E_i(t)$ $(t \in \{0, 1\})$ and if $\mathrm{L} \vdash F_1 \ldots F_n \to G$, then $\mathrm{L} \vdash E_1(t_1) \ldots E_n(t_n) \to G$ for some $t_1, \ldots, t_n$.
- $\mathrm{L} \vdash F_1 \ldots F_n \to G$ iff $c_1 \wedge \cdots \wedge c_n$ is satisfiable.

[M. Pentus. Complexity of the Lambek Calculus and Its Fragments. Proc. AiML '10]

NP-complete Fragments

Theorem (Savateev '08-'09)

Derivability problems for $\mathrm{L}(\backslash, /)$ *and* $\mathrm{L}(\cdot, \backslash)$ *are NP-complete.*

Complexity of Fragments of the Lambek Calculus

	L	L^*	$\mathrm{L}(p_1)$	$\mathrm{L}^*(p_1)$
$\cdot, \backslash, /$	NP	NP	NP	NP
$\cdot, \backslash$	NP	NP	NP	NP
$\backslash, /$	NP	NP	NP	NP
$\backslash$	P	P	P	P

Savateev's Proof Nets

$$\mathrm{Atn} = \{p^{(i)} \mid p \in \mathrm{Pr},\ i \in \mathbb{N}\}.$$

Translate the given sequent into a string of elements of Atn.

Savateev's Proof Nets

$\mathrm{Atn} = \{p^{(i)} \mid p \in \mathrm{Pr},\ i \in \mathbb{N}\}.$

Translate the given sequent into a string of elements of Atn.

$\gamma, \bar{\gamma} \colon \mathrm{Tp}(\backslash) \to \mathrm{Atn}^+.$

$$\begin{array}{cc} \gamma(p) = p^{(1)} & \bar{\gamma}(p) = p^{(2)} \\ \gamma(A \backslash B) = \bar{\gamma}(A)\gamma(B) & \bar{\gamma}(A \backslash B) = \bar{\gamma}(B)\big(\gamma(A)\big)^{+2} \end{array}$$

Translate $A_1 \ldots A_n \to B$ into $\gamma(A_1) \ldots \gamma(A_n)\bar{\gamma}(B)$.

Savateev's Proof Nets

$\mathrm{Atn} = \{p^{(i)} \mid p \in \mathrm{Pr},\ i \in \mathbb{N}\}.$

Translate the given sequent into a string of elements of Atn.

$\gamma, \bar{\gamma} \colon \mathrm{Tp}(\backslash) \to \mathrm{Atn}^{+}.$

$$\begin{array}{cc} \gamma(p) = p^{(1)} & \bar{\gamma}(p) = p^{(2)} \\ \gamma(A \setminus B) = \bar{\gamma}(A)\gamma(B) & \bar{\gamma}(A \setminus B) = \bar{\gamma}(B)\big(\gamma(A)\big)^{+2} \end{array}$$

Translate $A_1 \ldots A_n \to B$ into $\gamma(A_1) \ldots \gamma(A_n)\bar{\gamma}(B)$.

This translation is injective.

Savateev's Proof Nets

$\mathrm{Atn} = \{p^{(i)} \mid p \in \mathrm{Pr},\ i \in \mathbb{N}\}$.

Translate the given sequent into a string of elements of Atn.

$\gamma, \bar{\gamma} \colon \mathrm{Tp}(\backslash) \to \mathrm{Atn}^{+}$.

$$\begin{array}{cc} \gamma(p) = p^{(1)} & \bar{\gamma}(p) = p^{(2)} \\ \gamma(A \setminus B) = \bar{\gamma}(A)\gamma(B) & \bar{\gamma}(A \setminus B) = \bar{\gamma}(B)\big(\gamma(A)\big)^{+2} \end{array}$$

Translate $A_1 \dots A_n \to B$ into $\gamma(A_1) \dots \gamma(A_n)\bar{\gamma}(B)$.

This translation is injective.

Example. $(r \setminus p)\ ((s \setminus p) \setminus t) \to (s \setminus r) \setminus t$.

$r^{(2)} p^{(1)} p^{(2)} s^{(3)} t^{(1)} t^{(2)} s^{(4)} r^{(3)}$

Savateev's Proof Nets

A sequent is derivable in $\mathrm{L}(\backslash)$ iff on its translation to Atn^{+} there exists such pairing of letters, that

- a pair consists of $p^{(i)}$ and $p^{(i+1)}$, and $p^{(i)}$ stays to the left;
- the links connecting pairs can be drawn in the upper semiplane without intersections;
- if the superscript of the left atom in the pair is even, then there is an atom with a less superscript between the elements of the pair.

Using Proof Nets

- Making easier to obtain proof-theoretical results.

Using Proof Nets

- Making easier to obtain proof-theoretical results.
- For $\mathrm{L}(\backslash)$: polynomial decision procedure (dynamic programming à la CYK).

Using Proof Nets

- Making easier to obtain proof-theoretical results.
- For $\mathrm{L}(\backslash)$: polynomial decision procedure (dynamic programming à la CYK).
- ... and polynomial translation into context-free grammars.