Categorial Grammar

and

Formal Semantics

Richard Moot & Christian Retoré

LaBRI-CNRS, U Bordeaux & LIRMM, U Montpellier

ESSLLI 2014, Tübingen, 18–22 August 2014

Categorial Grammar

and

Formal Semantics

Richard Moot & Christian Retoré

LaBRI-CNRS, U Bordeaux & LIRMM, U Montpellier

ESSLLI 2014, Tübingen, 18–22 August 2014

B Proof Theory

B.1. Proof Theory

- Normalization
- Principal Branches
- Subformula property, decidability and parsing
- Curry-Howard isomorphism

B.2. Lambek grammars

$$\frac{A/B \quad B}{A}\,[/E] \qquad \frac{\begin{array}{c}\ldots [B]^n \\ \vdots \\ A\end{array}}{A/B}\,[/I]^n$$

$$\frac{B \quad B\backslash A}{A}\,[\backslash E] \qquad \frac{\begin{array}{c}[B]^n \ldots \\ \vdots \\ A\end{array}}{B\backslash A}\,[\backslash I]^n$$

B.3. Lambek grammars — Natural deduction, sequent style

$$\frac{\Gamma \vdash A/B \quad \Delta \vdash B}{\Gamma, \Delta \vdash A} \; [/E] \qquad\qquad \frac{\Gamma, B \vdash A}{\Gamma \vdash A/B} \; [/I]$$

$$\frac{\Gamma \vdash B \quad \Delta \vdash B\backslash A}{\Gamma, \Delta \vdash A} \; [\backslash E] \qquad\qquad \frac{B, \Gamma \vdash A}{\Gamma \vdash B\backslash A} \; [\backslash I]$$

B.4. Normalisation

Natural deduction proofs can contain “detours”, where an implication is eliminated directly after its introduction.

For example:

$$\dfrac{np \qquad \dfrac{\dfrac{[np]^1 \qquad np\backslash s}{s}\,\backslash E}{np\backslash s}\,\backslash I_1}{s}\,\backslash E$$

Normalisation removes such detours.

B.5. Normalization

$$
\dfrac{\dfrac{\begin{matrix}\ldots [B]^n \\ \vdots\ \delta' \\ A\end{matrix}}{A/B}\ /I_n \qquad \begin{matrix}\vdots\ \delta \\ B\end{matrix}}{A}\ /E
\quad\rightsquigarrow\quad
\begin{matrix} \vdots\ \delta \\ \ldots\ B \\ \vdots\ \delta' \\ A \end{matrix}
$$

B.6. Normalization

$$
\dfrac{\begin{array}{c}\vdots\;\delta\\ B\end{array} \qquad \dfrac{\begin{array}{c}[B]^n \ldots\\ \vdots\;\delta'\\ A\end{array}}{B\backslash A}\,\backslash I_n}{A}\,\backslash E
\quad\rightsquigarrow\quad
\begin{array}{c}\vdots\;\delta\\ B \quad \ldots\\ \vdots\;\delta'\\ A\end{array}
$$

B.7. Strong normalisation & confluence

Strong normalisation: however you choose to reduce reduce a proof normalisation terminates, because the number of rule in the proof decreases.

Confluence: whenever a proof π has two redexes whose reduction leads to π' and π'', both π' and π'' can be reduced to a single π° (because redexes can overlap, their traces can be found in π' and π'').

B.8. Principal Branch

Let us call a *principal branch* leading to F a sequence $H_0, \ldots, H_n = F$ of formulae of a natural deduction tree such that:

- H_0 is a free hypothesis
- H_i is the principal premise — the one carrying the eliminated symbol — of an elimination rule whose conclusion is H_{i+1}
- H_n is F

B.9. Subformula Property

Proposition 1 *Let d be a normal natural deduction, then:*

1. *if d ends with an elimination then there is a principal branch leading to its conclusion*
2. *each formula in d is the subformula of a free hypothesis or of the conclusion*

B.10. Corollary: decidability (in natural deduction)

As a corollary there exists a procedure that decides whether a sequent is provable or not (in particular parsing is decidable).

Look for a normal proof:

1) Inverse introduction rules.

2) Choose a possible head variable and check all the possible partitions of the sequences of formulae on its left and on its right.

Many choices may need to be explored: deduction (parsing) is an NP complete problem (Pentus, 2006, with the additional rules for product, Savateev, 2009, for the rules we have seen here).

B.11. Lambek grammars generate all context free languages

A measure of nesting:

$$
\begin{aligned}
o(p) &= 0 && \textit{when } p \textit{ is an atomic type}\\
o(A\backslash B) &= \max(o(A)+1, o(B))\\
o(B/A) &= \max(o(A)+1, o(B))
\end{aligned}
$$

Thus, the order of $(np\backslash s)/np$ is 1, but the order of $s/(np\backslash s)$ is 2.

B.12. Lambek grammars generate all context free languages

Proposition 2 (Cohen) *Let* $A_1, \ldots, A_n \vdash p$ *be provable in the Lambek calculus with* $o(A_i) \leq 1$ *and* p *a primitive type (and therefore of order zero). Then it is provable with* $[\backslash E]$ *and* $[/E]$ *only — in other words AB derivations and L derivations coincide when types are of order at most one.*

Proposition 3 *Given a context free grammar there exists a Lambek grammar that generates the same language.*

B.13. Lambek grammars generate all context free languages — proof

One may assume w.l.o.g Greibach normal form i.e. all rules are $X \to a$ or $X \to aY$ or $X \to aYZ$, with $a \in T$ and $X, Y, Z \in NT$

As we saw yesterday, there is an AB grammar that generates the very same langage. Turn

$X \to a$ into $a : X$

$X \to aY$ into $a : X/Y$

$X \to aYZ$ into $a : (X/Z)/Y$

Given that all types are of order 1, and s is of order 0 normal proofs (that are enough) of $A_1, \ldots, A_n \vdash s$ coincides because of previous property [Cohen].

B.14. Lambek grammars generate only context free languages

Chomsky conjecture of 1963 proved by Pentus in 1992.

Too difficult and lengthy to be proved here (see e.g. “The logic of categorial grammars”).

Hence parsing can be achieved in cubic time (w.r.t. the length of the parsed sentence) but with a corresponding enormous context free grammar (that yields a huge constant in front of the cubic polynomial).