Residuation, Structural Rules and Context Freeness

Gerhard Jäger
jaeger@ling.uni-potsdam.de
TAG+6
Venice
May 22, 2002

Outline of talk

- generative capacity of Categorial Grammars
- NL
- adding modalities
- context freeness of NL◊-grammars
- generalizations
- comparison to Kandulski
### Generative capacity of Categorial Grammars

#### Basic systems

<table>
<thead>
<tr>
<th></th>
<th>AB</th>
<th>CF</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ residuation</td>
<td>NL</td>
<td>CF</td>
<td>Kandulski (1988)</td>
</tr>
<tr>
<td>+ associativity</td>
<td>L</td>
<td>CF</td>
<td>Pentus (1993)</td>
</tr>
<tr>
<td>+ further structural rules</td>
<td>LP, LC, ...</td>
<td>⊆ CF</td>
<td>van Bentham (1991)</td>
</tr>
</tbody>
</table>

#### Multimodal systems

<table>
<thead>
<tr>
<th></th>
<th>NL + $\diamond_i, \Box^i_i$</th>
<th>CF</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+ associativity</td>
<td>L + $\diamond_i, \Box^i_i$</td>
<td>CF</td>
<td>Jäger (2001)</td>
</tr>
<tr>
<td>+ further structural rules</td>
<td>NL + postulates</td>
<td>Turing-complete</td>
<td>Carpenter (1999)</td>
</tr>
</tbody>
</table>
The non-associative Lambek Calculus

- Reasoning over binary trees
- root and leaves are labeled with formulas over \, \bullet, /
- Natural Deduction format:

\[ \Rightarrow \quad A \quad \text{id} \]
\[ \Rightarrow \quad B \quad \text{Cut} \]

\[ \Rightarrow \quad A \quad \text{id} \]
\[ \Rightarrow \quad B \quad \text{Cut} \]

\[ \Rightarrow \quad A \quad \text{id} \]
\[ \Rightarrow \quad B \quad \text{Cut} \]

\[ \Rightarrow \quad A \quad \text{id} \]
\[ \Rightarrow \quad B \quad \text{Cut} \]

\[ \Rightarrow \quad A \quad \text{id} \]
\[ \Rightarrow \quad B \quad \text{Cut} \]

\[ \Rightarrow \quad A \quad \text{id} \]
\[ \Rightarrow \quad B \quad \text{Cut} \]

\[ \Rightarrow \quad A \quad \text{id} \]
\[ \Rightarrow \quad B \quad \text{Cut} \]
NL

- conservative extension of NL

- logical vocabulary: additionally two unary modal operators, ◇ and □↓

- reasoning over unary/binary trees
From Logic to Grammar

- **NL◊-grammars** consist of
  - Lexicon: finite set of elementary non-branching trees with
    
    \[
    \begin{array}{c}
    \text{Category on top and terminal item on bottom:} \\
    \text{Cat} \\
    \text{lex}
    \end{array}
    \]
  - finite set of designated categories

- A string of terminals is recognized by the grammar iff it is the yield of a tree with a designated category as root label.
Generative Capacity

Lemma 1 Every context free language \( L \) is recognized by some \( \text{NL}^\diamond \)-grammar.

Proof:

- Bar-Hillel et al. (1960): Every context free language is recognized by some first order \( \text{AB} \)-grammar
- for first order grammars, the difference between \( \text{AB} \) and \( \text{NL}^\diamond \) doesn’t matter
- basically Cohen’s 1967 proof for inclusion of cfl in \( \text{L} \)-languages

Sequent presentation of \( \text{NL}^\diamond \)

\[
\frac{\text{id}}{A \Rightarrow A} \quad \frac{X \Rightarrow A \quad Y[A] \Rightarrow B}{Y[X] \Rightarrow B} \quad \text{Cut}
\]

\[
\frac{X[A \circ B] \Rightarrow C}{X[A \bullet B] \Rightarrow C} \quad \text{\( \bullet L \)} \quad \frac{X \Rightarrow A \quad Y \Rightarrow B}{X \circ Y \Rightarrow A \bullet B} \quad \text{\( \bullet R \)}
\]

\[
\frac{X \Rightarrow A \quad Y[B] \Rightarrow C}{Y[B/A \circ X] \Rightarrow C} \quad \text{\( / L \)} \quad \frac{X \circ A \Rightarrow B}{X \Rightarrow B/A} \quad \text{\( / R \)}
\]
Lemma 2 Let $X[Y] \Rightarrow A$ be a theorem of $\text{NL}\Diamond$. Then there is a type $B$ such that

1. $\text{NL}\Diamond \vdash Y \Rightarrow B$

2. $\text{NL}\Diamond \vdash X[B] \Rightarrow A$

3. There is a type occurring in $X[Y] \Rightarrow A$ which contains at least as many connectives as $B$.

Proof: Induction over sequent derivations.
Constructing a CFG from an NL◊-grammar

- **NL◊(n)**: fragment of NL◊ where each formula contains at most n connectives

- Let n be the maximal number of connectives occurring in an NL◊-grammar G.

- Equivalent cfg (construction inspired by Pentus 1993):

  \[
  \text{NL◊}(n) \vdash A \circ B \Rightarrow C \quad \leadsto \quad C \rightarrow A, B
  \]

  \[
  \text{NL◊}(n) \vdash \langle A \rangle \Rightarrow B \quad \leadsto \quad B \rightarrow A
  \]

  \[
  \text{NL◊}(n) \vdash A \Rightarrow B \quad \leadsto \quad B \rightarrow A
  \]

  \[
  \text{Cat} \quad \leadsto \quad \text{Cat} \rightarrow \text{lex}
  \]

  \[
  \text{lex} \quad \leadsto \quad A \rightarrow S
  \]
Lemma 3 Every NL\(\Diamond\)-recognizable language is context free.

Proof: By lemma 2 and the construction above.

Theorem 1 NL\(\Diamond\)-grammars recognize exactly the context free languages.

Proof: Immediate.

Generalization I: Connectives with arbitrary arity

- Both \(\setminus\), \(\cdot\), \(/\) and \(\Diamond\), \(\Box\uparrow\) form families of residuated operators
- Can be generalized to products/implications of arbitrary arity
- Logic of Pure Residuation (LPR): Generalization of NL to infinite number of families of residuated connectives of any arity
- Lemma 2 holds for LPR

Theorem 2 LPR-grammars recognize exactly the context free languages.
Generalization II: Structural Rules

- **LPR** can be strengthened by adding structural rules
- Lemma 2 remains valid if Permutation and/or Expansion are added
- Thus grammars based on these logics only recognize context free languages

\[
\frac{X[Y \circ Z] \Rightarrow A}{X[Z \circ Y] \Rightarrow A} \quad P \\
\frac{X[Y] \Rightarrow A}{X[Y \circ Y] \Rightarrow A} \quad E
\]

- Associativity, Contraction and Weakening do not preserve lemma 2

Relation to Kandulski’s work

- Differences:
  - Kandulski restricts designated categories to atomic categories
  - entirely different proof strategy: based on proof normalization in axiomatic version of LPR
References


