Introduction to Abstract Categorial Grammars

Fourth course
• Preliminary notions:
  ▶ coherence theorem
  ▶ closure properties
  ▶ definition of membership problems
• Complexity of membership problems:
  ▶ \( G(3, n) \)
  ▶ \( G(2, n) \)
• Non-linear ACGs
The coherence theorem
Properties of the linear lambda-calculus:
Properties of the linear lambda-calculus:

- Subject expansion
Properties of the linear lambda-calculus:

• **Subject expansion**

• **Every linear lambda-term is simply typable**
Properties of the linear lambda-calculus:

- Subject expansion
- Every linear lambda-term is simply typable
- The principal type of a linear lambda-term specifies uniquely this term (coherence)
Let's take $\Sigma = (\{o\}, \{a\}, \{a : o \rightarrow o\})$

$\lambda x.a(a(x))$ has $o \rightarrow o$ as principal type
**Linearity and coherence**

Let's take $\Sigma = (\{o\}, \{a\}, \{a : o \rightarrow o\})$

$\lambda x.a(a(x))$ has $o \rightarrow o$ as principal type

But so does $\lambda x.a(a(a(x)))$
Recovering linearity

Relabelings: lexicons of complexity 1
Recovering linearity

Relabelings: lexicons of complexity 1

Example:

\[ \Sigma' = (\{o_1, o_2, o_3\}, \{a_1, a_2\}, \{a_1 : o_2 \rightarrow o_1, a_2 : o_3 \rightarrow o_2\}) \]

\[ \Sigma = (\{o\}, \{a\}, \{a : o \rightarrow o\}) \]

\[ \mathcal{R}(o_i) = o \]

\[ \mathcal{R}(a_i) = a \]
Recovering linearity

Relabelings: lexicons of complexity 1

Example:

\[ \Sigma' = (\{o_1, o_2, o_3\}, \{a_1, a_2\}, \{a_1 : o_2 \rightarrow o_1, a_2 : o_3 \rightarrow o_2\}) \]

\[ \Sigma = (\{o\}, \{a\}, \{a : o \rightarrow o\}) \]

\[ R(o_i) = o \quad \lambda x.a_1(a_2 x) \text{ is the only term of type } o_3 \rightarrow o_1 \text{ in } \Lambda_{\Sigma'} \]
Recovering linearity

Relabelings: lexicons of complexity 1

Example:

\[ \Sigma' = (\{o_1, o_2, o_3\}, \{a_1, a_2\}, \{a_1 : o_2 \rightarrow o_1, a_2 : o_3 \rightarrow o_2\}) \]

\[ R(\Sigma') = (\{o\}, \{a\}, \{a : o \rightarrow o\}) \]

\[ R(o_i) = o \quad \lambda x.a_1(a_2 x) \text{ is the only term of type } o_3 \rightarrow o_1 \text{ in } \Lambda_{\Sigma'} \]

\[ R(a_i) = a \]

\[ R(\lambda x.a_1(a_2 x)) = \lambda x.a(a x) \]
**Recovering linearity**

**Proposition (coherence):**

For every term $t$ in $\Lambda_\Sigma$ there is:

- A signature $\Sigma'$
- A relabeling $\mathcal{R} : \Sigma' \rightarrow \Sigma$
- A term $t'$ in $\Lambda^\alpha_\Sigma$, verifying
Proposition (coherence):

for every term $t$ in $\Lambda_\Sigma$ there is:

a signature $\Sigma'$

a relabeling $R : \Sigma' \to \Sigma$

a term $t'$ in $\Lambda_\Sigma^\alpha$

$\text{verifying}$

$t'$ is the unique term of type $\alpha$ in $\Lambda_\Sigma'$
Recovering linearity

**Proposition (coherence):**

For every term \( t \) in \( \Lambda_{\Sigma} \) there is:

- A signature \( \Sigma' \)
- A relabeling \( \mathcal{R} : \Sigma' \to \Sigma \)
- A term \( t' \) in \( \Lambda^\alpha_{\Sigma'} \)

**Verifying**

\( t' \) is the unique term of type \( \alpha \) in \( \Lambda^\alpha_{\Sigma'} \)

\( \mathcal{R}(t') = t \)
Interpreting coherence

\[ a : o \rightarrow o \]

\[ \lambda x. a(a(a(a(x)))) \]

\[ a_i : o_{i+1} \rightarrow o_i \]

\[ \lambda x. a_1(a_2(a_3(a_4(a_5(x)))) : o_6 \rightarrow o_1 \]

\[ o_1 \ o_2 \ o_3 \ o_4 \ o_5 \ o_6 \]
Interpreting coherence

\[ a : o \rightarrow o \quad a_i : o_{i+1} \rightarrow o_i \]

\[ \lambda x.a(a(a(a(x)))) \quad \lambda x.a_1(a_2(a_3(a_4(a_5(x)))) : o_6 \rightarrow o_1 \]

\[ o_1 \quad o_2 \quad o_3 \quad o_4 \quad o_5 \quad o_6 \]

Intervals:

\[ \lambda x.a_i(\cdots(a_{i+k}x)\cdots) : o_{i+k+1} \rightarrow o_i \]
**Interpreting Coherence**

\[ a : o \rightarrow o \]
\[ \lambda x. a(a(a(a(x)))) \]
\[ \lambda x. a_1(a_2(a_3(a_4(a_5(x)))) : o_6 \rightarrow o_1 \]
\[ o_1 \ o_2 \ o_3 \ o_4 \ o_5 \ o_6 \]

**Intervals:**
\[ \lambda x. a_i(\cdots (a_{i+k} x ) \cdots ) : o_{i+k+1} \rightarrow o_i \]

**String contexts:**
\[ \lambda s. a_i + \cdots + a_{i+k} + s + a_{i+k+l} + \cdots + a_{i+k+l+m} : (o_{i+k+1} \rightarrow o_{i+k+l}) \rightarrow (o_{i+k+l+m} \rightarrow o_i) \]
Closure properties
Closure under inverse relabeling

Proposition:

Let \( G = \langle \Sigma_1, \Sigma_2, \mathcal{L}, s \rangle \) of \( G(m, n) \)
\( R : \Sigma_3 \rightarrow \Sigma_1 \)
\( \alpha \) a type in \( \Sigma_3 \) such that \( R(\alpha) = \mathcal{L}(s) \)

There is \( G' = \langle \Sigma'_1, \Sigma_3, \mathcal{L}', s' \rangle \) of \( G(m, n) \) such that

\[
\mathcal{O}(G') = R^{-1}(\mathcal{O}(G)) \cap \Lambda^\alpha_{\Sigma_3}
\]
A pictorial view

\[ \Lambda \Sigma_2 \rightarrow O(G) \]

\[ \mathcal{R}^{-1}(O(G)) \rightarrow \Lambda^\alpha_{\Sigma_3} \]

\[ \Lambda \Sigma_3 \cap O(G') \]
**Other closure properties**

\[ L(m, n) \text{ is closed under union} \]

\[ L_{\text{string}}(m, n) \text{ is closed under intersection with regular sets of strings} \]

\[ L_{\text{tree}}(m, n) \text{ is closed under intersection with regular sets of trees} \]
Membership problems
Two kinds of problems

Universal membership:

Fixed data: a class of grammar

Input: a grammar $G$ and an object $t$

Output: Yes/No when $t \in L(G)$ / $t \notin L(G)$
Two kinds of problems

Universal membership:
Fixed data: a class of grammar
Input: a grammar $G$ and an object $t$
Output: Yes/No when $t \in L(G)$ / $t \notin L(G)$

Membership:
Fixed data: a grammar $G$
Input: an object $t$
Output: Yes/No when $t \in L(G)$ / $t \notin L(G)$
<table>
<thead>
<tr>
<th></th>
<th>Membership</th>
<th>Universal Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFGs / TAGs</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Lambek Grammars</td>
<td>P</td>
<td>NP-complete</td>
</tr>
<tr>
<td>MCFGs</td>
<td>P</td>
<td>PSPACE-complete / EXPTIME-complete</td>
</tr>
</tbody>
</table>
Complexity of $G(3, n)$
The emptiness problem in $G(3, n)$ is Turing-equivalent to proof search in MELL.

So is the membership problem.
Proof search in MELL is Turing-equivalent to the emptiness problem of $\Lambda_\Sigma^\alpha$

The emptiness problem in $G(3, n)$ is Turing-equivalent to proof search in MELL

So is the membership problem
Membership and MELL

\[ \Lambda^s_{\Sigma_1} \rightarrow A(G) \]
\[ \Lambda^{L(s)}_{\Sigma_2} \rightarrow O(G) \]

\[ t \in \Lambda^{L(s)}_{\Sigma_2} \quad t \in O(G) \]
Membership and MELL

Coherence

\[ \Lambda_{\Sigma_1} = \Lambda_{\Sigma_2} \]

\[ \Lambda^\alpha_{\Sigma_3} = \{ t' \} \]

\[ t \in \Lambda_{\Sigma_2}^{L(s)} \]

\[ t \in \mathcal{O}(G) \]

\[ \mathcal{R}(t') = t \]
Membership and MELL

Coherence

Closure under inverse relabeling

\[ \Lambda^s_{\Sigma_1} \quad \Lambda^{L(s)}_{\Sigma_2} \quad \Lambda^\alpha_{\Sigma_3} = \{ t' \} \]

\[ O(G') \quad R^{-1}(O(G)) \]

\[ t \in \Lambda^{L(s)}_{\Sigma_2} \quad t \in O(G) \quad R(t') = t \]
MEMBERSHIP AND MELL

Exercise:

Reduce emptiness of an ACG to the membership problem
MEMBERSHIP AND MELL

Exercise:

Reduce emptiness of an ACG to the membership problem

Hint 1: prove that sets of pure terms of a given type is finite
**Membership and MELL**

**Exercise:**

Reduce emptiness of an ACG to the membership problem

**Hint 1:** prove that sets of pure terms of a given type is finite

**Hint 2:** prove that the sets of pure terms built with types of the form

\[ T ::= (\sigma \rightarrow \sigma) \mid (T \rightarrow T) \]

are not empty
Lexicalized grammars

\[ G = \langle \Sigma_1, \Sigma_2, \mathcal{L}, s \rangle \text{ is lexicalized if for every } c \in \Sigma_1 \]
\[ \mathcal{L}(c) \text{ contains at least one object constant} \]
**Lexicalized Grammars**

\[
G = \langle \Sigma_1, \Sigma_2, \mathcal{L}, s \rangle \text{ is lexicalized if for every } c \in \Sigma_1 \\
\mathcal{L}(c) \text{ contains at least one object constant}
\]

<table>
<thead>
<tr>
<th></th>
<th>Membership</th>
<th>Universal Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>(G(3, 1))</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
<tr>
<td>(G(3, n))</td>
<td>NP-complete</td>
<td>NP-complete</td>
</tr>
</tbody>
</table>
Complexity of $G(2, n)$
<table>
<thead>
<tr>
<th>(G(2, 1))</th>
<th>(P)</th>
<th>(P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(G(2, n) \quad n &gt; 1)</td>
<td>(P)</td>
<td>PSPACE-complete</td>
</tr>
</tbody>
</table>
Let $G = (N, T, R, s)$ be a CFG

$a_1 \ldots a_n$
Let $G = (N, T, R, s)$ be a CFG

$\ a_1 \ldots a_n$

**Items** $(Y,i,j)$ **Meaning** $Y \rightarrow^* a_{i+1} \ldots a_j$
CYK algorithm for CFGs

Let $G = (N, T, R, s)$ be a CFG

$a_1 \ldots a_n$

Items $(Y, i, j)$ Meaning $Y \rightarrow^* a_{i+1} \ldots a_j$

$X \rightarrow w_1X_1 \ldots w_nX_nw_{n+1}$

$w_1 = a_i \ldots a_j \ (X_1, j + 1, k)$

\[ \vdots \]

$w_n = a_i \ldots a_m \ (X_n, m + 1, p)$

$w_{n+1} = a_{p+1} \ldots a_q$

$(X, i, q)$
Let $G = \langle \Sigma_1, \Sigma_2, \mathcal{L}, s \rangle$ be in $G(2, n)$ and $t \in \Lambda^{\mathcal{L}(s)}_{\Sigma_2}$

Coherence: $\Sigma_3, \mathcal{R}, \alpha, t'$

Item: $(\beta, \gamma)$ with $\beta \in \Sigma_1, \gamma \in \Sigma_2$ and $\mathcal{L}(\beta) = \mathcal{R}(\gamma)$

Meaning: there is $t \in \Lambda^\beta_{\Sigma_1}$ such that $\mathcal{R}^{-1}(\mathcal{L}(t)) \cap \Lambda^\gamma_{\Sigma_3} \neq \emptyset$

\[
(\beta_1, \gamma_1) \cdots (\beta_n, \gamma_n)
\]

\[
\mathcal{R}^{-1}(\mathcal{L}(c)) \cap \Lambda_{\Sigma_3}^{\gamma_1} \cdots \gamma_n \gamma \neq \emptyset
\]

$(\beta, \gamma)$
Non linear ACGs
ABSTRACT
SYNTACTIC
STRUCTURE

\[ \Sigma_0 \]
abstract syntactic structure

Σ₀

Σ₁

SYNTACTIC FORM

ABSTRACT SYNTACTIC STRUCTURE
ABSTRACT SYNTACTIC STRUCTURE

\[ \Sigma_0 \]

\[ \Sigma_1 \]
SYNTACTIC FORM

\[ \Sigma_2 \]
SEMANTIC FORM
Abstract syntactic structure

\[ \Sigma_0 \]

\[ L_{\text{synt}} \]

\[ \Sigma_1 \]

Syntactic form

\[ \Sigma_2 \]

Semantic form
ABSTRACT
SYNTACTIC
STRUCTURE

$\Sigma_0$

$\mathcal{L}_{\text{synt}}$

$\Sigma_1$

SYNTACTIC
FORM

$\mathcal{L}_{\text{sem}}$

$\Sigma_2$

SEMANTIC
FORM
Decidability and non-linearity

\( G_{\text{non-lin}}(3, n) \): undecidable membership problem

\( G_{\text{non-lin}}(2, n) \): decidable membership problem

\( \exists x. \text{unicorn } x \land \text{find } j x \)
Decidability of $G_{\text{non-lin}}(2,n)$

- Recognizability for lambda-terms
- Singleton sets are recognizable
- Recognizable sets are closed under:
  - Boolean operations
  - Inverse homomorphisms (lexicons)