Variablenfreie Semantik

Ziel:

kompositionale Interpretation (nicht-indizierter) syntaktischer Strukturen, die
  o direkt ist (d. h. keinen Gebrauch einer zusätzlichen Ebene zwischen Syntax
    und Semantik macht)
  o lokal ist (eine Herausforderung dafür stellen lange Abhängigkeiten wie Ex-
    traktionen und pronomiale Bindung dar)

(1) Wen glaubst du, dass Maria liebt ____?
(2) Jeder Mensch hofft, dass Maria denkt, dass er die Welt verstan-
    den hat.

Idee:

Bedeutungen sind unabhängig von Variablenbelegungen. Variablen sind kein es-
  sentialer Bestandteil von Ausdrücken oder deren Bedeutung, sondern spielen nur
für repräsentationelle Zwecke eine Rolle: sie sind metasprachliche Hilfsmittel, um
das implizite semantische Potential von Ausdrücken explizit zu machen. Auf Va-
riablen und Abstraktion über diese kann zugunsten von funktionalen Operationen
(Kombinatoren) verzichtet werden.

Kombinatorische Logik:

Grundobjekte sind Funktionen und die Grundoperation ist funktionale Applika-
  tion. Es gibt einige spezielle Operatoren (Kombinatoren), die Funktionen mani-
pulierven.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>[ B f g x = f(g(x)) ]</td>
</tr>
<tr>
<td>S</td>
<td>[ S f g x = f(x(g(x)) ]</td>
</tr>
<tr>
<td>G</td>
<td>[ G f g x = f(g(x)) x ]</td>
</tr>
<tr>
<td>T</td>
<td>[ T x f = f x ]</td>
</tr>
<tr>
<td>W</td>
<td>[ W f x = f x x ]</td>
</tr>
<tr>
<td>I</td>
<td>[ I x = x ]</td>
</tr>
</tbody>
</table>

\[ \lambda f \lambda g \lambda x [f(g(x))] \]
1 Der technische Apparat (Jacobson)

Definition 1.0.1 (syntaktische Kategorien)

- Basiskategorien: $S, NP$
- Sind $A$ und $B$ syntaktische Kategorien, so sind auch $(B/LA), (B/RA)$ und $A^B$ syntaktische Kategorien.

Definition 1.0.2 (semantische Typen)

Ausdrücke der Kategorie $S$ sind vom Typ $t$, Ausdrücke der Kategorie $NP$ sind vom Typ $e$ und Ausdrücke der Kategorie $(A/B)$ sowie Ausdrücke der Kategorie $A^B$ sind vom Typ $\langle b, a \rangle$ (wobei $b$ der semantische Typ von Ausdrücken der Kategorie $B$ und $a$ der semantische Typ von Ausdrücken der Kategorie $A$ ist).

Definition 1.0.3 (g: Geach-Regel, $g =$ unäres B)

- Syntax: $g((B/A) = B^C/A^C$
- Semantik: Ist $f$ eine Funktion vom Typ $\langle a, b \rangle$, dann ist $g(f)$ eine Funktion vom Typ $\langle \langle c, a \rangle, \langle c, b \rangle \rangle$, wobei $g(f) = \lambda V[\lambda C[f(V(C))]]$ mit $V$ vom Typ $\langle c, a \rangle$ und $C$ vom Typ $c$.

Definition 1.0.4 (z: Bindung, $z = G$)

- Syntax: $z((B/NP)/A) = (B/NP)/A^{NP}$
- Semantik: $z(g) = \lambda f \lambda x[g(f(x))(x)]$ mit $g$ vom Typ $\langle a, \langle e, b \rangle \rangle$ und $f$ vom Typ $\langle c, a \rangle$

Definition 1.0.5 (l: Kategorien-/Typenanhebung, $l = T$)

- Syntax: $l_p(A) = B/(B/A)$ oder $B/(B/A)$
- Semantik: $l_b(a') = \lambda P[P(a')]$ mit $P$ vom Typ $\langle a, b \rangle$

Definition 1.0.6 (Verkettung)

Verkettung als Applikation:

$$\frac{A/RB : a'}{B : \emptyset} \quad \frac{B : \emptyset}{A : a'(\emptyset)}$$

Verkettung als Komposition (= binäres B):

$$\frac{A/RB : a'}{A/R : \emptyset} \quad \frac{B/R : \emptyset}{A/LB : a'}$$

$$\frac{A/R : \emptyset}{A/R : \emptyset} \quad \frac{B/R : \emptyset}{A/LB : a'}$$
2 Pronominale Bindung

2.1 Reflexiva (Szabolcsi)

Reflexivpronomen sind Duplikatoren (W), sind also Operatoren, die zweistellige Prädikate durch Argumentidentifizierung in einstellige Prädikate überführen.

*Beispiel:*

\[
\begin{array}{c|c}
\text{Mary} & \text{likes} \\
\text{NP} & \text{Lex} \\
\text{m} & \quad (S/LNP)/RNP \\
\text{like}' & \lambda y \lambda x [\text{like}'(y)(x)] \\
\hline
\text{herself} & \text{Lex} \\
\text{W} & (S/LNP)/(S/LNP)/RNP \\
\lambda f \lambda x [f(x)(x)] & \quad A \\
\hline
\text{W(like')(m)} & \quad A \\
\text{like'(m)(m)} & \quad A
\end{array}
\]

2.2 Nicht-reflexive Pronomina (Jacobson)

Pronomen sind von der syntaktischen Kategorie NP, also vom semantischen Typ \( \langle e, e \rangle \). Sie denotieren die Identitätsfunktion über Individuen.

Bindung ist keine Beziehung zwischen sprachlichen Ausdrücken (DPn,...), sondern zwischen Argumentslots, und wird mithilfe der Typenanhebungsoperationen \( g \) und \( z \) modelliert.

2.3 Probleme

Prinzip A- und Prinzip B-Effekte, d.h. Lokalitätsbeschränkungen, die die (in vielen Fällen komplementäre) Distribution von Reflexivpronomen und nicht-reflexiven Pronomen steuern

**Prinzip A**

Ein Reflexivpronomen muss in seiner Bindungsdomäne gebunden sein.

**Prinzip B**

Ein nicht-reflexives Pronomen muss in seiner Bindungsdomäne frei sein.

(3) Mary believes that John loves her/’herself.
3 Extraktion

Beispiel:
(the woman) who John says Bill loves ___

Mit Variablen:

\[ \lambda x_1[\text{say}'(\text{love}'(x_1)(b))(j)] \]

who\_1 \quad \lambda x_1 \quad \text{say}'(\text{love}'(x_1)(b))(j)

John \quad \lambda x[\text{say}'(\text{love}'(x_1)(b))(x)]

says \quad \lambda p \lambda x[\text{say}'(x)(p)]

\quad \text{loves} \quad \lambda x[\text{love}'(x_1)(y)]

\quad \text{loves} \quad \lambda y \lambda x[\text{love}'(y)(x)]

\quad b \quad t_1 \quad x_1

Ohne Variablen:

Das Prinzip ist analog zu pronominaler Bindung.

Relativpronomen sind Pronomen, d.h. genauso Identitätsfunktionen über Individuen: \( \lambda x[x] \) (Kategorie \( NP^NP \) und Typ \( \langle e, e \rangle \)).

Literatur


John thinks Mary said he lost.

\[
\begin{align*}
\text{he} \quad & \quad \frac{\text{lost}}{\text{Lex}} \\
NP^{NP} : \lambda x[x] \quad & \quad \frac{S/1NP : \text{lost}'}{\text{Lex}} \\
& \quad \frac{S^{NP}/1NP^{NP} : \lambda f \lambda y[\text{lost}')(f(y))]}{g} \\
& \quad \frac{S^{NP} : \lambda y[\text{lost}')(y)] = \text{lost}'}{A}
\end{align*}
\]

Mary said he lost

\[
\begin{align*}
\text{Mary} \quad & \quad \frac{NP : m}{\text{Lex}} \\
S/1(S/LNP) : \lambda P(m) \quad & \quad \frac{\lambda f \lambda x[\text{say}'(f(x))(m)]}{g} \\
& \quad \frac{(S/1NP)^{NP}/1S^{NP} : \lambda f \lambda x[\text{say}'(\text{lost}')(x)](z)}{A}
\end{align*}
\]

\[
\begin{align*}
S^{NP} : \lambda x[\text{say}'(\text{lost}')(x)(m)]
\end{align*}
\]

John said Mary he lost

\[
\begin{align*}
\text{John} \quad & \quad \frac{(S/1NP)/R^{S} : \lambda P \lambda z[\text{think}')(p)(z)]}{\text{Lex}} \\
(S/LNP)/R^{SNP} : \lambda P \lambda x[\text{think}')(P(x)(x)] \quad & \quad \frac{z \quad \text{Mary said he lost}}{S^{NP} : \lambda x[\text{say}'(\text{lost}')(x)(m)]}
\end{align*}
\]

\[
\begin{align*}
S/1LNP : \lambda x[\text{think}')(\text{say}'(\text{lost}')(x)(m)(x)]
\end{align*}
\]

\[
\begin{align*}
S : \text{think}')(\text{say}'(\text{lost}')(j)(m)(j)) = B(T(m))(B'(B'(\text{B'}(\text{I}))))
\end{align*}
\]