

# Unterspezifikation in der Semantik

## Quantoren in Objektposition

Laura Kallmeyer  
Heinrich-Heine-Universität Düsseldorf  
Wintersemester 2011/2012

---

Quantoren in Objektposition 1 24. Oktober 2011

### Overview

1. Problem
2. Flexible Typen (Ambiguität im Lexikon)
3. Quantifying In
4. Quantifier Raising
5. Cooper Storage

[Heim and Kratzer, 1998]

---

Quantoren in Objektposition 2 24. Oktober 2011

### Problem (1)

Beispiel: Quantor in Subjektposition

(1) every linguist admires Chomsky

- *linguist* denotiert ein einstelliges Prädikat, das die Menge aller Linguisten charakterisiert.

$$[[\text{linguist}]] = \lambda x. \text{linguist}'(x)$$

- *admires Chomsky* denotiert ebenfalls ein einstelliges Prädikat, die Menge aller Bewunderer von Chomsky.

$$[[\text{admires Chomsky}]] = \lambda x. \text{admire}'(x, \text{Chomsky}')$$

- *every* ist ein generalisierter Quantor, der auf die beiden Prädikate angewendet wird:

$$[[\text{every linguist admires Chomsky}]] =$$

$$[[\text{every}]]([[ \text{linguist} ]], [[ \text{admires Chomsky} ]]) =$$

$$\text{every}'(\lambda x. \text{linguist}'(x), \lambda x. \text{admire}'(x, \text{Chomsky}'))$$

---

Quantoren in Objektposition 3 24. Oktober 2011

### Problem (2)

Problem: Wie interpretieren wir Quantoren in Objektposition?

(2) Chomsky inspires every linguist

Ziel:

- *Chomsky inspires* denotiert ein einstelliges Prädikat, die Menge aller, die von Chomsky inspiriert werden.

$$[[\text{Chomsky inspires}]] = \lambda x. \text{inspire}'(\text{Chomsky}', x)$$

- *every* ist ein generalisierter Quantor, der auf die beiden Prädikate  $[[\text{linguist}]]$  und  $[[\text{Chomsky inspires}]]$  angewendet wird:

$$[[\text{Chomsky inspires every linguist}]] =$$

$$[[\text{every}]]([[ \text{linguist} ]], [[ \text{Chomsky inspires} ]]) =$$

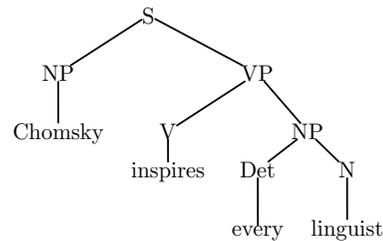
$$\text{every}'(\lambda x. \text{linguist}'(x), \lambda x. \text{inspire}'(\text{Chomsky}', x))$$

---

Quantoren in Objektposition 4 24. Oktober 2011

**Problem (3)**

Problem: Wir müssen die Semantik aufgrund der Syntax berechnen.  
Es gibt aber keine Konstituente *Chomsky inspires*.

**Flexible Typen (1)**

Mögliche Lösung: Flexible Typen, je nachdem, ob ein Quantor in Subjekt- oder Objektposition steht.

- Quantifizierte DP in Subjektposition: Typ  $\langle\langle e, t \rangle, t\rangle$   
[[everybody]] =  $\lambda P. every'(\lambda x. person'(x), \lambda x. P(x))$
- Quantifizierte DP in Objektposition: Typ  $\langle\langle e, \langle e, t \rangle \rangle, \langle e, t \rangle\rangle$   
[[everybody]] =  $\lambda P \lambda x. every'(\lambda y. person'(y), \lambda y. P(x, y))$

Dabei wird angenommen, dass bei  $S \rightarrow NP VP$  das Denotat der NP auf das der VP angewendet wird, während bei  $VP \rightarrow V NP$  das Denotat der NP auf das des V angewendet wird. Die *funktionale Applikation* hat also jeweils eine andere Richtung.

**Flexible Typen (2)**

Unbefriedigend, da

- Quantoren zwei Lexikoneinträge (oder sogar mehr, für die Kombination mit dreistelligen Verben) benötigen,
- Quantoren, die nicht dort Skopus nehmen, wo sie stehen, für diesen Ansatz problematisch sind.

(3) somebody likes everybody

- $some'(\lambda x. person'(x), \lambda x. every'(\lambda y. person'(y), \lambda y. like'(x, y)))$
- $every'(\lambda y. person'(y), \lambda y. some'(\lambda x. person'(x), \lambda x. like'(x, y)))$

Eine Interpretation in situ mit flexiblen Typen würde nur (3a.) vorhersagen.

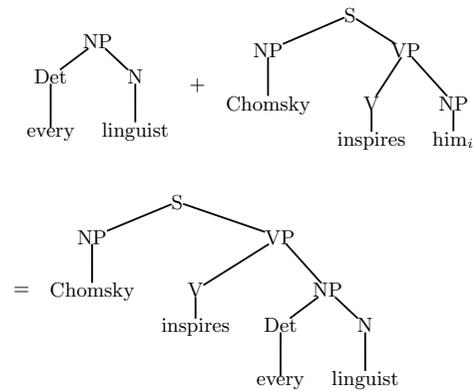
**Quantifying In (1)**

[Montague, 1974, Dowty et al., 1981]

Idee: Es gibt nicht nur Funktionale Applikation, sondern auch andere Verknüpfungen, die in der Syntax aus zwei Teilbäumen einen neuen herstellen und in der Semantik irgendeine Operation durchführen.

*Quantifying In* syntaktisch: Ein zunächst offenes Pronomen wird durch eine quantifizierende NP ersetzt.

D.h., die NP wird gewissermaßen auf den Satz ohne das noch fehlende Argument angewendet.

**Quantifying In (2)**

Sei  $\alpha$  die NP,  $\phi$  der S-Baum mit Pronomenindex  $i$ , dann sei  $Syn_{Qu-in.i}(\alpha, \phi)$  das Resultat von *Quantifying In*.

**Quantifying In (3)**

*Quantifying In* semantisch

Für  $\alpha$  vom Typ  $\langle\langle e, t \rangle, t\rangle$  und  $\phi$  vom Typ  $t$  gilt

$$[[Syn_{Qu-in.i}(\alpha, \phi)]] = [[\alpha]](\lambda x. [[\phi]]^{x/i})$$

(dabei bedeutet das Superscript  $x/i$ , dass das Pronomen mit Index  $i$  durch  $x$  ersetzt wird)

$$[[\text{Chomsky inspires every linguist}]]$$

$$= [[Syn_{Qu-in.i}(\text{every linguist}, \text{Chomsky inspires } x_i)]]$$

$$= \text{every}'(\lambda x. \text{linguist}'(x), \lambda x. \text{inspire}'(\text{Chomsky}', x))$$

Mit dieser Analyse werden Skopusambiguitäten zu derivationellen Ambiguitäten: Die Reihenfolge, in der die Quantoren mit *Quantifying In* hinzugefügt werden, bestimmt, wer über wen Skopus hat. (Je später desto höher.)

**Quantifier Raising (1)**

GB und Minimalist Grammar:

- Die Semantik ergibt sich nicht unmittelbar aus der Oberflächenstruktur, sondern zwischen Oberflächenstruktur und *Logischer Form (LF)* kann es noch Transformationen geben.
- Quantoren stehen zunächst, in der Oberflächenstruktur, in situ. Auf dem Weg zur LF können Sie nach oben bewegt werden und je nach "Landeplatz" Skopus nehmen.

**Quantifier Raising (2)**