Semantik und Pragmatik

20. Mai 2008

Gerhard Jäger

Typengetriebene Interpretation

Regelformate

- bislang drei Arten von Regeln:
  - $X \rightarrow Y, Z :: \|X\| = \|Y\|(\|Z\|)$
  - $X \rightarrow Y, Z :: \|X\| = \|Z\|(\|Y\|)$
  - $X \rightarrow Y, Z, W :: \|X\| = \|Z\|(\|Y\|)(\|W\|)$
- Gemeinsamkeiten:
  - eines der Elemente auf der rechten Seite denotiert Funktion
  - andere Elemente auf der rechten Seite denotieren Argumente für diese Funktion
  - Bedeutung des Mutterknotens: Resultat der Anwendung des Funktors auf die Argumente
  - **semantische Operation ist immer Funktionsanwendung**
  - **Es gibt immer nur eine Möglichkeit, die Bedeutung eines der Tochterknoten auf die Bedeutung des/der anderen Tochterknoten anzuwenden**

$\Rightarrow$ semantische Operation ergibt sich aus Definitionsbereich der beteiligten Funktionen

Typengetriebene Interpretation

- Typ einer Funktion: *Definitionsbereich, Wertebereich*
- allgemeine semantische Kompositionsregel:

Prinzip der typengetriebenen Interpretation
Die Bedeutung des Mutterknotens ist das Resultat der Anwendung der Bedeutung eines der Tochterknoten auf die Bedeutung(en) des/der anderen Tochterknoten. Aufgrund der Typen der beteiligten Funktionen ist diese Operation immer eindeutig definiert.

- semantische Regel ergibt sich damit immer eindeutig aus syntaktischer Regel

$\rightsquigarrow$ semantische Regel muss nicht eigens angegeben werden

Argumentstruktur und Lambda-Präfixe

- Verben – ein paar Beispiele:
  - *regnen* $\rightsquigarrow \lambda s.\text{RAIN'}(s)$
  - *schlafen* $\rightsquigarrow \lambda x \lambda s.\text{SLEEP'}(s, x)$
  - *lesen* $\rightsquigarrow \lambda y \lambda x \lambda s.\text{READ'}(s, x, y)$
  - *geben* $\rightsquigarrow \lambda z \lambda y \lambda x \lambda s.\text{GIVE'}(s, x, y, z)$
- Muster: Die Interpretation eines *n*-stelligen Verbs hat immer $n + 1$ Lambdas (ein Lambda pro Argumentstelle, plus ein Lambda für die Situationsvariable).
- Argumentstruktur ist also direkt aus Bedeutung ablesbar

Indefinite Ellipse

- bei (manchen) transitiven Verben kann das Objekt weggelassen werden
- z.B.
  - *Peter liest Anna Karenina.* $\Rightarrow$
  - *Peter liest.*
- Resultat der Ellipse folgt immer logisch aus nicht-elliptischer Variante

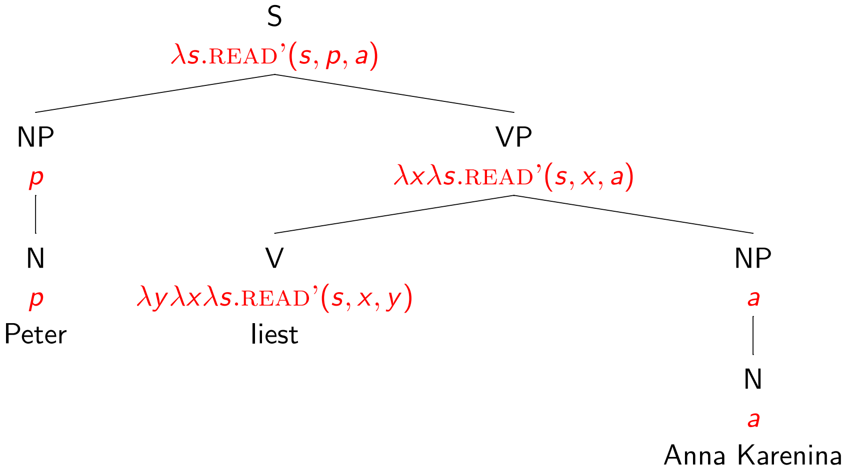
Diathesen und lexikalische Regeln

Indefinite Ellipse

- es gibt zwei Verben *lesen*, ein transitives und ein intransitives, die semantisch verwandt sind
- **Lexikalische Regel:** Wenn V ein transitives Verb ist mit der Bedeutung α, dann ist V auch ein intransitives Verb, mit der Bedeutung $\lambda x \lambda s.\exists y(\alpha(y)(x)(s))$
- also:
  - Bedeutung von transitivem *lesen*: $\lambda y \lambda x \lambda s.\text{READ'}(s, x, y)$
  - Bedeutung von *lesen* als intransitives Verb ist

$$\lambda x \lambda s.\exists y(\text{READ'}(s, x, y))$$

Indefinite Ellipse



Indefinite Ellipse

- S: $\lambda s.\exists y(\textsc{read}'(s, p, y))$
  - NP: p
    - N: p
      - Peter
  - VP: $\lambda x \lambda s.\exists y(\textsc{read}'(s, x, y))$
    - V: $\lambda x \lambda s.\exists y(\textsc{read}'(s, x, y))$
      - liest

$$\| \textit{Peter liest Anna Karenina} \| \subseteq \| \textit{Peter liest} \|$$

$$\textit{Peter liest Anna Karenina} \Rightarrow \textit{Peter liest}$$

Passiv

- Passiv:
  - *Peter liest Anna Karenina*
  - *Anna Karenina wird gelesen*
- Passiv wandelt transitives (zweistelliges) Verb in intransitives (einstelliges) Partizip um
- Partizip muss aus syntaktischen Gründen mit Hilfsverb auftreten

Diathesen und lexikalische Regeln

Passiv

- **Lexikalische Regel:** Wenn V ein transitives Verb ist mit der Bedeutung α, dann hat das Partizip II von V die Bedeutung $\lambda x \lambda s.\exists y(\alpha(x)(y)(s))$
- $\|gelesen\| = \lambda x \lambda s.\exists y(\textsc{read}'(s, y, x))$
- Das Hilfsverb trägt nichts zur Bedeutung bei:[1]

$$\|wird\| = \lambda P \lambda x.P(x)$$

- syntaktische Kategorie von Hilfsverben: T
- Syntax-Regel:

$$S \rightarrow NP, T, VP$$

[1] Abgesehen von Tempus und Modus, die wir bislang noch ignorieren.

Passiv

S
$\lambda s.\exists y(\textsc{read}'(s, y, a))$

NP
a

N
a

Anna Karenina

T
$\lambda P \lambda x.P(x)$
wird

VP
$\lambda x \lambda s.\exists y(\textsc{read}'(s, y, x))$

V
$\lambda x \lambda s.\exists y(\textsc{read}'(s, y, x))$
gelesen

$\|$*Peter liest Anna Karenina*$\| \subseteq \|$*Anna Karenina wird gelesen*$\|$

Peter liest Anna Karenina $\Rightarrow$ *Anna Karenina wird gelesen*

Quantoren

Einführung

- bisher nur eine Art von NP: Eigennamen (*Peter, Hans, Anna Karenina, ...*)
- daneben gibt es eine Vielzahl weiterer NPs:
  - *niemand, jeder, keiner, alle, jemand, ...*
  - *jede Frau, einige Frauen, die meisten Frauen, drei Frauen, eine Frau, viele Frauen, wenige Frauen, die drei Frauen*
- diese Art NPn heißen **generalisierte Quantoren** (oder einfach **Quantoren**, wenn Verwechslung mit Quantoren i.S.d. Logik ausgeschlossen ist)

Generalisierte Quantoren Bestimmte Inferenzmuster, die für Eigennamen gelten, gelten nicht für GQn:

(1)
a. Hans liest Anna Karenina $\Rightarrow$ Anna Karenina wird gelesen.
b. Niemand liest Anna Karenina $\not\Rightarrow$ Anna Karenina wird gelesen.

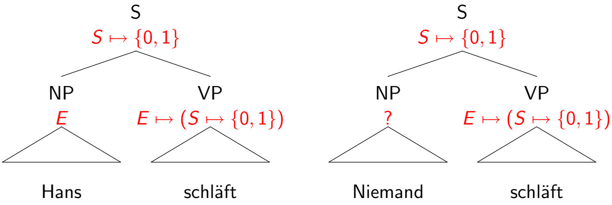
(2)
a. Hans kennt Anna und Hans mag Maria $\Leftrightarrow$ Hans kennt Anna und mag Maria.
b. Ein Mann kennt Anna und ein Mann mag Maria $\not\Leftrightarrow$ Ein Mann kennt Anna und mag Maria.

(3)
a. Hans kennt Anna oder Hans mag Maria $\Leftrightarrow$ Hans kennt Anna oder mag Maria.
b. Jeder Mann kennt Anna oder jeder Mann mag Maria $\not\Leftrightarrow$ Jeder Mann kennt Anna oder mag Maria.

Generalisierte Quantoren

- Wenn Bedeutung von GQn Individuum wäre, müssten diese Inferenzmuster aber gelten!

$\leadsto$ Bedeutung eines GQ ist nicht ein Individuum



Quantoren

Generalisierte Quantoren

- Wenn Bedeutungskomposition weiterhin über Funktionsanwendung geschehen soll, müssen GQn Bedeutungen des folgenden Typs sein:

  $$(E \mapsto (S \mapsto \{0,1\})) \mapsto (S \mapsto \{0,1\})$$

  also Funktionen von VP-Bedeutungen in Satz-Bedeutungen
- einfacher ausgedrückt (wobei Bildung der charakteristischen Funktion und Schönfinkelisierung implizit bleibt):

  $$POW(S \times POW(S \times E))$$

Quantoren

Generalisierte Quantoren

- Bedeutung einiger GQn:
  - jeder, alle: $\lambda P \lambda s. \forall x(\text{PERSON'}(x) \rightarrow P(s, x))$
  - niemand, keiner: $\lambda P \lambda s. \neg \exists x(\text{PERSON'}(x) \wedge P(s, x))$
  - jemand: $\lambda P \lambda s. \exists x(\text{PERSON'}(x) \wedge P(s, x))$
- Allgemeines Muster: Bedeutung des Quantors ist Satzbedeutung, bei der die Bedeutung der VP durch eine Variable ersetzt wird, über die Lambda-abstrahiert wird

Generalisierte Quantoren

S

$\lambda s \neg \exists x (\text{PERSON'}(x) \wedge \text{SLEEP'}(s, x))$

NP

$\lambda P \lambda s . \neg \exists x (\text{PERSON'}(x) \wedge P(s, x))$

Niemand

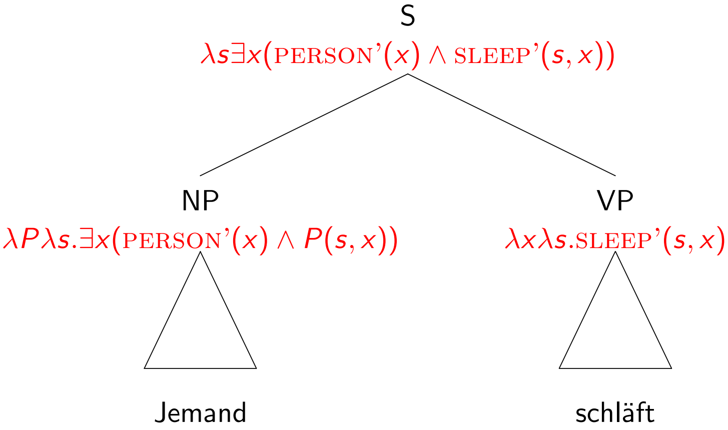
VP

$\lambda x \lambda s . \text{SLEEP'}(s, x)$

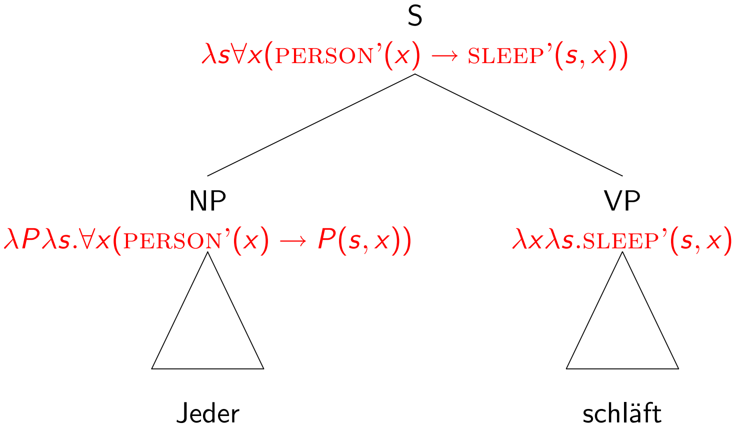
schläft

Quantoren

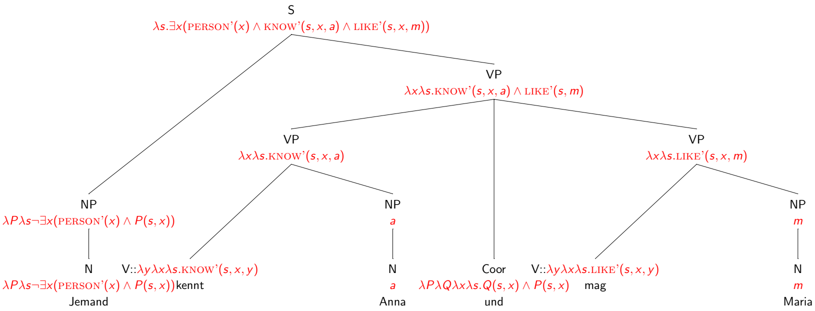
Generalisierte Quantoren



Generalisierte Quantoren



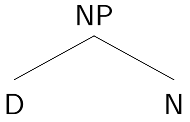
Generalisierte Quantoren



Quantoren

Determinierer

- Wie berechnet sich Bedeutung von syntaktisch komplexen GQn?



- Bedeutung des Nomens: Eigenschaft $\rightsquigarrow$ Teilmenge von E
- Bedeutung des Determinierers: Funktion von Bedeutung eines Nomens in Bedeutung eines GQ

$$POW(E) \mapsto (E \mapsto (S \mapsto \{0,1\})) \mapsto (S \mapsto \{0,1\})$$

- *de facto* äquivalent zu

$$POW(S \times POW(S \times E) \times POW(E))$$