Einführung in die Semantik, 8. Sitzung
Typentheorie, $F_{deutsch}$

Götz Keydana

Göttingen
1. Dezember 2006

Semantik von $F_{deutsch}$
- Satzoperatoren
  - Negation
  - Konjunktion und Disjunktion
- Individuenkonzepte

Typentheorie
- Extension und Intension
- Typentheorie revisited

Die Negation

Die Negation bildet das Komplement zu einer Proposition.

- Wir führen die Domäne der Propositionen D_p ein:
  - $D_p = [W \longrightarrow \{0, 1\}]$
- Wir führen Variablen für Propositionen p, p' ein.
- Wir verwenden den Operator $\neg$ der Aussagenlogik:
  - $\neg$ ist die Funktion $[1 \rightarrow 0, 0 \rightarrow 1]$
- Es folgt:
  $[\![\text{nicht}]\!] = \lambda p \in D_p \lambda w \in W[\neg p(w)]$
- semantische Regel für S $\longrightarrow$ S Mod:
  $[\![[_S[_S\alpha][_{Mod}\beta]]]\!]^M = [\![[_{Mod}\beta]]\!]^M([\![[_S\alpha]]\!]^M)$

Ein Beispiel

(1) $[\![[_S[_S\text{Pynchon schnarcht}][_{Mod}\text{ nicht}]]]\!]^M$

(a.) $[\![[_{Mod}\text{nicht}]]\!]^M([\![[_S\text{Pynchon schnarcht}]]\!]^M)$

(b.) $[\![\text{nicht}]\!]^M([\![\text{schnarcht}]\!]^M([\![\text{Pynchon}]\!]^M))$

(c.) $\lambda p\lambda w[\neg p(w)](\lambda x\lambda w'[x\text{ schnarcht in }w'](\text{Pynchon}))$

(d.) $\lambda p\lambda w[\neg p(w)](\lambda w'[\text{Pynchon schnarcht in }w'])$

(e.) $\lambda w[\neg[\lambda w'[\text{Pynchon schnarcht in }w'](w)]$

(f.) $\lambda w\neg[\text{Pynchon schnarcht in }w]$

Probleme

(2) Pynchon schnarcht und Ellison gähnt nicht

Unsere syntaktische Regel S $\rightarrow$ S Mod erlaubt folgende Strukturen:

(2a.) $[_S\ [_S$ Pynchon schnarcht] und $[_S\ [_S$ Ellison gähnt] $[_{Mod}$ nicht]]]

(2b.) $[_S\ [_S\ [_S$ Pynchon schnarcht] und $[_S$ Ellison gähnt]] $[_{Mod}$ nicht]]

(2b.) ergibt eine falsche Lesart: Die Negation kann nicht Skopus über den gesamten Satz haben.

Lösung

nicht ist eine VP-Negation:

(2c.) $[_S$ $[_S$ Pynchon schnarcht] und $[_S$ $[_{NP}$ Ellison] $[_{VP}$ $[_{VP}$ gähnt] $[_{Mod}$ nicht]]]]

Wir führen für diese Analyse folgende neue Phrasenstrukturregel ein:

- VP $\rightarrow$ VP Mod

Die Semantik der VP-Negation

- Die syntaktische Regel für VP-Negation ist kategoriekonstant: Eine negierte VP bleibt eine VP.
- Für den semantischen Typ der Negation bedeutet das, daß die Bedeutung der Negation die Bedeutung einer VP in die Bedeutung einer VP überführt.
- Der semantische Typ einer VP ist eine Funktion von Individuen in Funktionen von möglichen Welten in Wahrheitswerte.
- Folglich ist der Typ der Negation eine Funktion von Funktionen von Individuen in Funktionen von möglichen Welten in Wahrheitswerte in Funktionen von Individuen in Funktionen von möglichen Welten in Wahrheitswerte.

Die semantische Regel

- Wir führen D_{VP} als die Domäne der VP-Bedeutungen ein.
- Wir führen P, P′ als Variablen für VP-Bedeutungen ein.
- Dann folgt:
  $[\![\text{nicht}]\!]^M = \lambda P \in D_{VP} \lambda x \in D_e \lambda w \in W[\neg P(x)(w)]$
- semantische Regel für VP → VP Mod:
  $[\![[_{\text{VP}}[_{\text{VP}}\alpha][_{\text{Mod}}\beta]]]\!]^M = [\![[_{\text{Mod}}\beta]]\!]^M([\![[_{\text{VP}}\alpha]]\!]^M)$

Ein Beispiel

(2) $[\![[_S [_{NP} \text{Ellison}][_{VP} [_{VP} \text{gähnt}][_{Mod} \text{nicht}]]]]\!]^M$

(a.) $[\![[_{VP} [_{VP} \text{gähnt}][_{Mod} \text{nicht}]]]\!]^M([\![[_{NP} \text{Ellision}]]\!]^M)$

(b. $[\![[_{Mod} \text{nicht}]]\!]^M([\![[_{VP} \text{gähnt}]]\!]^M)([\![[_{NP} \text{Ellision}]]\!]^M)$

(c.) $[\![\text{nicht}]\!]^M([\![\text{gähnt}]\!]^M)([\![\text{Ellision}]\!]^M)$

(d.) $\lambda P \lambda x \lambda w[\neg P(x)(w)](\lambda x' \lambda w'[x' \text{ gähnt in } w'])(\text{Ellison})$

(e.) $\lambda x \lambda w[\neg \lambda x' \lambda w'[x' \text{ gähnt in } w'](x)(w)](\text{Ellison})$

(f.) $\lambda x \lambda w[\neg[x \text{ gähnt in } w]](\text{Ellison})$

(g.) $\lambda w[\neg[\text{Ellison gähnt in } w]]$

Die Koordinatoren

Die syntaktische Regel für Koordination erzeugt eine ternäre Struktur: S → S Coor S.

- Gegeben sei die Domäne der Propositionen D_p,
- Variablen für Propositionen p, p'.
- Wir verwenden die Operatoren $\wedge$ und $\vee$ der Aussagenlogik:
  - $\wedge$ ist die Funktion $[\langle 1,1\rangle \rightarrow 1, \langle 1,0\rangle \rightarrow 0, \langle 0,1\rangle \rightarrow 0, \langle 0,0\rangle \rightarrow 0]$
  - $\vee$ ist die Funktion $[\langle 1,1\rangle \rightarrow 1, \langle 1,0\rangle \rightarrow 1, \langle 0,1\rangle \rightarrow 1, \langle 0,0\rangle \rightarrow 0]$
- Es folgt:
  $[\![\text{und}]\!] = \lambda p \in D_p \lambda p' \in D_p \lambda w \in W[p(w) \wedge p'(w)]$
  $[\![\text{oder}]\!] = \lambda p \in D_p \lambda p' \in D_p \lambda w \in W[p(w) \vee p'(w)]$
- semantische Regel für S → S Coor S:
  $[\![[_S[_S\alpha][_{Coor}\beta][_S\gamma]]]\!]^M = [\![[_{Coor}\beta]]\!]^M([\![[_S\alpha]]\!]^M)([\![[_S\gamma]]\!]^M)$

Ein Beispiel

(3) $[\![[_S [_S \text{Pynchon gähnt}][_{Coor}\, und][_S \text{Ellison schnarcht}]]]\!]^M$

(a.)

$[\![[_{Coor}\text{und}]\!]^M([\![[_S \text{Pynchon gähnt}]]\!]^M)([\![[_S \text{Ellison schnarcht}]]\!]^M)$

(b.) $\lambda p \lambda p' \lambda w[p(w) \wedge$
$p'(w)]([\![[_S \text{Pynchon gähnt}]]\!]^M)([\![[_S \text{Ellison schnarcht}]]\!]^M)$

(c.) $\lambda w[[\![[_S \text{Pynchon gähnt}]]\!]^M(w) \wedge$
$[\![[_S \text{Ellison schnarcht}]]\!]^M(w)]$

(d.) $\lambda w[\lambda w'[\text{Pynchon gähnt in } w'](w) \wedge$
$\lambda w'[\text{Ellison schnarcht in } w'](w)]$

(e.) $\lambda w[\text{Pynchon gähnt in } w \wedge \text{Ellison schnarcht in } w]$

Probleme mit Namen

Wir haben die Bedeutung von NPs bisher unabhängig von möglichen Welten definiert.

- Wir fügen unserem Vokabular einen weiteren Namen hinzu: *der Papst*.
- Auf wen dieser Name referiert, hängt von dem Kontext ab, in dem wir den Namen benutzen. Die Referenz muß daher über mögliche Welten definiert werden.
- Namen wie *der Papst* bezeichnen Individuenkonzepte, d.h. Funktionen von möglichen Welten in Individuen.
- Wir führen die Domäne der Individuenkonzepte D_I ein:
  - D_I ist die Menge der Funktionen von W in D_e.
- Für die Bedeutung von *Papst* gilt:
  $[\![\text{der Papst}]\!]^M = \lambda w \in W[\text{der Papst in } w]$

Typenuniformität

Wir haben nun 2 verschiedene Bedeutungen für Namen:

- $[\![\text{Ellison}]\!]^M = \text{Ellison}$
- $[\![\text{der Papst}]\!]^M = \lambda w \in W[\text{der Papst in } w]$

Um Einheitlichkeit zu erzielen, können wir den Typ von Namen wie *Ellison* auf den von Namen wie *der Papst* anheben:

- $[\![\text{Ellison}]\!]^M = \lambda w \in W[\text{Ellison in } w]$
- Als Konsequenz dieser Anhebung müssen wir nun allerdings die Bedeutung von VPs und Vs modifizieren:
  $[\![\text{schnarcht}]\!]^M = \lambda x \in D_I \lambda w \in W[x(w) \text{ schnarcht in } w]$
  $[\![\text{liest}]\!]^M = \lambda y \in D_I \lambda x \in D_I \lambda w \in W[x(w) \text{ liebt } y(w) \text{ in } w]$

Wir werden im folgenden auf diese Komplikation verzichten.

binär verzweigende Strukturen

Wir kennen verschiedene syntaktische Regeln für binär verzweigende Strukturen wie:

- $[\![[_S[_{NP}\alpha][_{VP}\beta]]]\!]^M = [\![[_{VP}\beta]]\!]^M([\![[_{NP}\alpha]]\!]^M)$
- $[\![[_{VP}[_V\alpha][_{NP}\beta]]]\!]^M = [\![[_V\alpha]]\!]^M([\![[_{NP}\beta]]\!]^M)$
- etc.

In allen Fällen gibt die Regel explizit an, welcher Teilausdruck auf welchen appliziert wird.

Aber: Der semantische Typ eines Teilausdrucks bestimmt, ob der Teilausdruck Funktion oder Argument ist. Wir können daher unsere Regeln auf folgende Regel reduzieren:

- $[\![[\alpha\beta]]\!]^M = [\![\alpha]\!]^M([\![\beta]\!]^M) \vee [\![\beta]\!]^M([\![\alpha]\!]^M)$
  in Abhängigkeit davon, welche Kombination semantisch wohlgeformt ist.

Typengetriebene Interpretation

Da die Typen der Teilausdrücke die Interpretation des Gesamtausdrucks bestimmen, nennt man eine solche semantische Regel *typengetriebene Interpretation*.

Die neue Regel gilt für jede Art binär verzweigter Phrasenstrukturen.

Typengetriebene Interpretation erlaubt es, die Syntax von $F_{deutsch}$ zu vereinfachen:

- Wir ersetzen die beiden VP-Regeln durch eine:
  VP $\rightarrow$ V (NP), wobei () Optionalität anzeigt.
- Alle Verben führen wir nun durch eine Regel ein:
  V $\rightarrow$ {schnarcht, gähnt, liest, kennt}

Übergenerierung

Die Syntax erzeugt nun Sätze wie

(8) Pynchon schnarcht

(9) Ellison kennt Pynchon

(10) *Ellison kennt

(11) *Pynchon schnarcht Ellison

Modularität

Übergenerierung ist trotzdem ausgeschlossen, weil Sätze wie (10) und (11) keine Satzbedeutung haben.

(10) *Ellison kennt

(a.) $[\![[_{VP}[_{V}\text{kennt}]]]\!]^M([\![[_{NP}\text{Ellison}]]\!]^M)$ oder $[\![[_{NP}\text{Ellison}]]\!]^M([\![[_{VP}[_{V}\text{kennt}]]]\!]^M)$

(b.) $[\![\text{kennt}]\!]^M([\![\text{Ellison}]\!]^M)$ oder $[\![\text{Ellison}]\!]^M([\![\text{kennt}]\!]^M)$

(c.) $\lambda y \lambda x \lambda w[x \text{ kennt } y \text{ in } w](\text{Ellison})$ oder $\text{Ellison}(\lambda y \lambda x \lambda w[x \text{ kennt } y \text{ in } w])$

(d.) $\lambda y \lambda x \lambda w[x \text{ kennt } y \text{ in } w](\text{Ellison})$

(e.) $\lambda x \lambda w[x \text{ kennt Ellison in } w]$

(10e.) ist ungesättigt, somit kein Satz. Aber: Wir haben keine syntaktische Regel, die es erlaubt, $[_S[_{NP}\text{Ellison}][_{VP}[_V\text{kennt}]]]$ mit einer weiteren NP zu konkatenieren.

Typen bisher

Wir haben in der Ermittlung semantischer Regeln wiederholt mit semantischen Typen gearbeitet:

- Wir haben den semantischen Typ der VP aus dem Wissen abgeleitet, daß
  - der semantische Typ der NP das Individuum ist,
  - der semantische Typ von S eine Funktion von möglichen Welten in Wahrheitswerte ist.
  - Weil S $\rightarrow$ NP VP, muß der semantische Typ der VP eine Funktion von Individuen in Funktionen von möglichen Welten in Wahrheitswerte sein.
  - Wir haben gesehen, daß transitive Verben mit einem Objekt und intransitive Verben, also [kennt Ellison] und [schnarcht], denselben semantischen Typ haben.
  - Wir haben eine semantische Regel zur typengetriebenen Interpretation für binär verzweigende Phrasenstrukturen eingeführt.

Typentheorie

Wir nehmen (vorläufig) 2 Arten gesättigter Bedeutungen an: Individuen (Namenbedeutungen) und Wahrheitswerte (Satzbedeutungen).

Entsprechend führen wir folgende Elementartypen ein:

- e (Individuen oder Entitäten in D_e)
- t (Wahrheitswerte in $D_t = \{1, 0\}$)

Mit folgender rekursiver Regel können komplexe Typen gebildet werden:

- Wenn σ, τ Typen sind, dann ist $\langle \sigma, \tau \rangle$ ein Typ.

Ein einfaches Beispiel

(8) Pynchon schnarcht

- *Pynchon* ist vom Typ e
- *Pynchon schnarcht* ist vom Typ t
- Also ist *schnarcht* vom Typ $\langle e, t \rangle$

$$\dfrac{\dfrac{\text{Pynchon}}{e : \text{Pynchon}} \qquad \dfrac{\text{schnarcht}}{\langle e, t \rangle : \lambda x \lambda w[x \text{ schnarcht in } w]}}{\begin{array}{ll} t: & \lambda x \lambda w[x \text{ schnarcht in } w](\text{Pynchon}) \\ & \lambda w[\text{Pynchon schnarcht in } w] \end{array}}$$

Beispiel eines transitiven Satzes

(9) Ellison kennt Pynchon

$$
\dfrac{\dfrac{\dfrac{\text{kennt}}{\langle e,\langle e,t\rangle\rangle : \lambda y\lambda x\lambda w[x \text{ kennt } y \text{ in } w]} \quad \dfrac{\text{Pynchon}}{e : \text{Pynchon}}}{\begin{array}{ll}\langle e,t\rangle : & \lambda y\lambda x\lambda w[x \text{ kennt } y \text{ in } w](\text{Pynchon}) \\ & \lambda x\lambda w[x \text{ kennt Pynchon in } w]\end{array}} \quad \dfrac{\text{Ellison}}{e : \text{Ellison}}}{\begin{array}{ll}t : & \lambda x\lambda w[x \text{ kennt Pynchon in } w](\text{Ellison}) \\ & \lambda w[\text{Ellison kennt Pynchon in } w]\end{array}}
$$

Argumentreihenfolge

Wir haben die Bedeutung von z.B. *schnarcht* bisher folgendermaßen dargestellt:

- $[\![\text{scharcht}]\!]^M = \lambda x \lambda w[x \text{ schnarcht in } w]$
  Die Bedeutung der VP ist eine Funktion von Individuen in Funktionen von möglichen Welten in Wahrheitswerte.

Wir können die Argumente aber auch in umgekehrter Reihenfolge abbinden:

- $[\![\text{schnarcht}]\!]^M = \lambda w \lambda x[x \text{ schnarcht in } w]$
  Die Bedeutung der VP ist nunmehr eine Funktion von möglichen Welten in Funktionen von Individuen in Wahrheitswerte.

Beide Darstellungen sind äquivalent.

Wir können die Bedeutung sämtlicher Ausdrücke auf diese Weise als Funktion von möglichen Welten in andere Entitäten darstellen.

Bedeutungsregel für binäre Phrasenstrukturen

Wir müssen unsere Bedeutungsregel für komplexe Strukturen an das neue Format anpassen:

- $[\![[\alpha\beta]]\!]^M = \lambda w[[\![\alpha]\!]^M(w)([\![\beta]\!]^M(w))]$ oder $\lambda w[[\![\beta]\!]^M(w)([\![\beta]\!]^M(w))]$
  je nach semantischer Wohlgeformtheit

Beispielableitung

(10.) Ellison schnarcht

(a.) $\lambda w[[\![\text{schnarcht}]\!]^M(w)([\![\text{Ellison}]\!]^M(w))]$

(b.) $\lambda w[\lambda w'\lambda x[x \text{ schnarcht in } w'](w)(\lambda w''[\mathit{Ellison}](w))]$

(c.) $\lambda w[\lambda x[x \text{ schnarcht in } w](\mathit{Ellison})]$

(d.) $\lambda w[\text{Ellison schnarcht in } w]$

Intension und Extension

- Bedeutungen von Ausdrücken sind Funktionen von möglichen Welten.
- Die Bedeutung eines Ausdrucks in einer bestimmten Welt wird durch Anwendung der Funktion auf eben diese Welt gewonnen.
- Wir haben somit zwei Ebenen der Bedeutung:
  - Die Intension eines Ausdrucks ist die Funktion von möglichen Welten. Ihre Kenntnis erlaubt es einem Hörer/Sprecher zu bestimmen, ob ein Ausdruck in einer gegebenen Welt zutrifft.
  - Die Extension eines Ausdrucks ist die Intension appliziert auf eine bestimmte Welt.

$$[\text{Intension von } \alpha](w) = \text{Extension von } \alpha \text{ in } w$$

Ein neuer elementarer Typ

Um unsere Typentheorie der intensionalen Semantik anzupassen, führen wir eine dritte gesättigte Bedeutung ein, die mögliche Welt.

- e (Individuen oder Entitäten in D_e)
- t (Wahrheitswerte in $D_t = \{1, 0\}$)
- s (mögliche Welten in D_s)
- Wenn σ, τ Typen sind, dann ist $\langle \sigma, \tau \rangle$ ein Typ.

Beispielanwendungen

(8) Pynchon schnarcht

$$\dfrac{\dfrac{\text{Pynchon}}{\langle s, e\rangle} \qquad \dfrac{\text{schnarcht}}{\langle s, \langle e, t\rangle\rangle}}{\langle s, t\rangle}$$

(9) Ellison kennt Pynchon

$$\dfrac{\dfrac{\dfrac{\text{kennt}}{\langle s, \langle e, \langle e, t\rangle\rangle\rangle} \qquad \dfrac{\text{Pynchon}}{\langle s, e\rangle}}{\langle s, \langle e, t\rangle\rangle} \qquad \dfrac{\text{Ellison}}{\langle s, e\rangle}}{\langle s, t\rangle}$$

Zurück zum Anfang

Da jeder sprachliche Ausdruck vom Typ $\langle s, \sigma \rangle$ ist, kann in der Darstellung auf den Typ s verzichtet werden:

(8) Pynchon schnarcht

$$\frac{\dfrac{\text{Pynchon}}{e} \qquad \dfrac{\text{schnarcht}}{\langle e, t \rangle}}{s}$$

(9) Ellison kennt Pynchon

$$\frac{\dfrac{\dfrac{\text{kennt}}{\langle e, \langle e, t \rangle \rangle} \qquad \dfrac{\text{Pynchon}}{e}}{\langle e, t \rangle} \qquad \dfrac{\text{Ellison}}{e}}{t}$$