

# Formale Semantik

## 07. Getypte höherstufige L-Sprachen

Roland Schäfer

Institut für Germanistische Sprachwissenschaft  
Friedrich-Schiller-Universität Jena

Folien in Überarbeitung. Englische Teile (ab Woche 7) sind noch von 2007!

Stets aktuelle Fassungen: <https://github.com/rsling/VL-Semantik>

- 1 Einfachere Semantik
- 2 Getypte Sprachen

- 3  $\lambda$ -Sprachen
- 4 Ausblick auf Quantifikation bei Montague

Wie unterscheidet sich Montagues System von GB-Semantik?

Welche Rolle spielen Typen?

Was sind  $\lambda$ -Sprachen?

Und woher kennen Sie den  $\lambda$ -Operator eigentlich schon?

Text für heute: Dowty u. a. (1981: Kapitel 4) | Chierchia & McConnell-Ginet 2000: Kapitel 7

Einfachere Semantik

Es geht wie immer auch ohne Bewegung.

- Chierchia | auf Grundlage von GB-Syntax
  - ▶ Syntax und Semantik in Phrasenstrukturen
  - ▶ Sprache wird zu Logik durch unsichtbare Bewegung.
  - ▶ Semantik als eigene Repräsentationsebene
- Montague | Sprache ist Logik!
  - ▶ Direkte Interpretation von Zeichen als logische Symbole
  - ▶ Logische Form (lf) als Sichtbarmachen logischer Eigenschaften
  - ▶ Keine Übersetzung

## Mengen, über Funktionen definiert

- Große Bedeutung von Mengen in formaler Semantik
- Charakteristische Funktion von Mengen  
 $\mathcal{S}(a) = 1 \text{ iff } a \in S, \text{ else } 0$
- CF als Einsortierung in ihre Menge
- CF in Mengendefinitionen  
 $S = \{x : x \bmod 2 = 0\}$   
 $\mathcal{S} = f(x)[x \bmod 2 = 0]$
- Äquivalenz von Mengendenotation und CF-Dontation

## Funktionsapplikation als allgemeiner Kompositionsmechanismus

- Etwas umständliche Interpretation mit T-Sätzen

$$\llbracket [s \text{ NP VP}] \rrbracket^{\mathcal{M},g} = 1 \text{ iff } \llbracket \text{NP} \rrbracket^{\mathcal{M},g} \in \llbracket \text{VP} \rrbracket^{\mathcal{M},g}$$

- CF statt Mengen | Funktion appliziert direkt!

- ▶  $\llbracket \text{Mary} \rrbracket^{\mathcal{M},g} = \text{Mary in } \mathcal{M}$
- ▶  $\llbracket \text{sleeps} \rrbracket^{\mathcal{M},g}$  *be the CF of the set of sleepers in  $\mathcal{M}$*
- ▶  $\llbracket \text{S} \rrbracket^{\mathcal{M},g} = \llbracket \text{sleeps} \rrbracket^{\mathcal{M},g}(\llbracket \text{Mary} \rrbracket^{\mathcal{M},g})$
- ▶ Kein Bedarf an T-Sätzen

Funktionen von Mengen von (Tupeln von) Individuen zu Ausdrücken usw.

- Funktionen **Definitionsbereich**  $S_1 \rightarrow$  **Wertebereich**  $S_2 \mid S_2^{S_1}$   
 $S_2^{S_1}$  | Die Menge aller Funktionen von  $S_1$  zu  $S_2$
- Beispiel | Einstellige und Zweistellige Prädikate
  - ▶  $T = \{0, 1\}$  | Wahrheitswerte
  - ▶  $D$  | Diskursuniversum (Menge aller Individuen)
  - ▶  $D \times D$  | Menge aller 2-Tupel von Individuen
  - ▶  $T^D$  | Menge aller Funktionen von Individuen zu Wahrheitswerten  
Menge der CFs aller einstelligen Prädikate
  - ▶  $T^{D \times D}$  | Menge aller Funktionen von 2-Tupeln von Individuen zu Wahrheitswerten  
Menge der CFs aller zweistelligen Prädikate



# Getypte Sprachen

Logik hat bereits **Typensysteme**, um Syntax zu strukturieren!

- $L_{Type}$  | Prädikatenlogik  $L_1$  plus Typen
- Typen | Semantisch fundierte Klassen von Ausdrücken
- Einfache Typen
  - ▶ Terme |  $\langle e \rangle$
  - ▶ Wffs/Formeln |  $\langle t \rangle$  | Ersetzt Startsymbol S der PSG!
- Komplexe/funktionale Typen
  - ▶ Einstellige Prädikate |  $\langle e, t \rangle$
  - ▶ Zweistellige Prädikate |  $\langle e, \langle e, t \rangle \rangle$
- Allgemein |  $\langle \sigma, \tau \rangle$ -Ausdrücke denotieren Funktionen von Denotaten von  $\langle \sigma \rangle$ -Ausdrücken zu Denotaten von  $\langle \tau \rangle$ -Ausdrücken.

Homogenes Diskursuniversum  $D$  (auch  $U$  und bei Dowty u. a. 1981  $A$ )

- Allgemein  $D_\alpha$  | Menge von Denotaten von Ausdrücken des Typs  $\alpha$
- Einfache Typen
  - ▶  $D_{\langle e \rangle} = U$
  - ▶  $D_{\langle t \rangle} = \{0, 1\}$
- Komplexe Typen | Rekursiv definierte Denotate
  - ▶ Allgemein |  $D_{\langle \alpha, \beta \rangle} = D_{\langle \beta \rangle}^{D_{\langle \alpha \rangle}}$
  - ▶ Einstellige Prädikate |  $D_{\langle e, t \rangle} = D_{\langle t \rangle}^{D_{\langle e \rangle}}$
  - ▶ Zweistellige Prädikate |  $D_{\langle e, \langle e, t \rangle \rangle} = (D_{\langle t \rangle}^{D_{\langle e \rangle}})^{D_{\langle e \rangle}}$
- Interpretation weiterhin durch  $V, g$

$\langle\sigma\rangle$ -Ausdrücke saturieren  $\langle\sigma, \tau\rangle$ -Ausdrücke zu  $\langle\tau\rangle$ -Ausdrücken.

- Beispiel für Saturierung durch Funktionsapplikation (FA)
  - ▶ Wenn  $P$  vom Typ  $\langle e, \langle e, t \rangle \rangle$ ,  $Q$  vom Typ  $\langle e, t \rangle$  und  $x, y$  vom Typ  $\langle e \rangle$
  - ▶ dann ist  $Q(x)$  vom Typ  $\langle t \rangle$
  - ▶ und  $P(x)$  vom Typ  $\langle e, t \rangle$  sowie  $P(x)(y)$  vom Typ  $\langle t \rangle$
- Funktionale Typen von Funktoren
  - ▶ Negation  $\neg$  | Typ  $\langle t, t \rangle$
  - ▶ Andere Funktoren  $\wedge, \vee, \rightarrow, \leftrightarrow$  | Typ  $\langle t, \langle t, t \rangle \rangle$

Wirklich keine T-Sätze mehr!

- Semantik für  $\langle e \rangle$ -Typen (Terme)

$$\llbracket a_n \rrbracket^{\mathcal{M},g} = V(a_n)$$

$$\llbracket x_n \rrbracket^{\mathcal{M},g} = g(x_n)$$

- Ansonsten nur FA

$$\llbracket \delta(\alpha) \rrbracket^{\mathcal{M},g} = \llbracket \delta \rrbracket^{\mathcal{M},g}(\llbracket \alpha \rrbracket^{\mathcal{M},g})$$

Sprache höherer Ordnung = Sprache mit Variablen über höhere Typen  $\langle \sigma, \tau \rangle$

- *Type* ist die Menge aller Typen
  - ▶  $\langle e \rangle, \langle t \rangle \in \text{Type}$
  - ▶ Wenn  $\langle \sigma \rangle, \langle \tau \rangle \in \text{Type}$ , dann  $\langle \sigma, \tau \rangle \in \text{Type}$
  - ▶ Nichts sonst ist in *Type*.
- *ME* ist die Menge aller bedeutungsvollen Ausdrücke
  - ▶  $ME_\sigma$  ist die Menge der Ausdrücke vom Typ  $\sigma$  |  $ME = \bigcup ME_\sigma$  mit  $\sigma \in \text{Type}$
  - ▶ *Ty* ist eine Funktion von Ausdrücken zu ihren Typen |  $Ty(a) = \sigma$  iff  $a \in ME_\sigma$
- Höhere Ordnung | Variablen über Ausdrücke von funktionalen Typen
  - ▶  $P_{\langle e, t \rangle}$  und  $Q_{\langle e, \langle e, t \rangle \rangle}$  | Bekannte Konstanten höherer (=funktionaler) Typen
  - ▶ Parallel  $v_{n_{\langle e, t \rangle}}$  | Die  $n$ -te Variable über einstellige Prädikate
  - ▶ Damit möglich  $M = \{v_{1_{\langle e, t \rangle}} : \llbracket v_{1_{\langle e, t \rangle}}(m) \rrbracket = 1\}$   
Wenn  $\llbracket m \rrbracket = \text{Maria}$ , dann ist  $M$  die Menge von Marias Eigenschaften!

Zusammenfassung | Die Semantik reduziert sich auf FA und Variablenauswertung.

- Interpretation von Termen und Funktionsausdrücken
  - ▶ Nicht-logische Konstanten |  $\alpha: \llbracket \alpha \rrbracket^{\mathcal{M},g} = V(\alpha)$
  - ▶ Variablen |  $\alpha: \llbracket \alpha \rrbracket^{\mathcal{M},g} = V(\alpha)$
  - ▶ Wenn  $\alpha \in \langle a, b \rangle$  und  $\beta \in a$  dann  $\llbracket \alpha(\beta) \rrbracket^{\mathcal{M},g} = \llbracket \alpha \rrbracket^{\mathcal{M},g}(\llbracket \beta \rrbracket^{\mathcal{M},g})$
- Logische Konstanten (Typen  $\langle t, t \rangle$  und  $\langle t, \langle t, t \rangle \rangle$ ) denotieren Funktionen in  $\{0, 1\}$ .
- Quantoren
  - ▶ Für Variable  $v_{1\langle \alpha \rangle}$  und Wff  $\phi \in ME_t$  ist  $\llbracket (\forall v_1)\phi \rrbracket^{\mathcal{M},g} = 1$  gdw für alle  $a \in D_\alpha$   $\llbracket \phi \rrbracket^{\mathcal{M},g[a/v_1]} = 1$
  - ▶ Für Variable  $v_{1\langle \alpha \rangle}$  und Wff  $\phi \in ME_t$  ist  $\llbracket (\exists v_1)\phi \rrbracket^{\mathcal{M},g} = 1$  gdw für mindestens ein  $a \in D_\alpha$   $\llbracket \phi \rrbracket^{\mathcal{M},g[a/v_1]} = 1$

$$\forall v_{\langle e, t \rangle} \left[ v_{\langle e, t \rangle}(j) \rightarrow v_{\langle e, t \rangle}(d) \right]$$

- Eine quantifizierbare Variable vom Typ  $\langle e, t \rangle$  |  $v_{\langle e, t \rangle}$
- Zwei Individuenkonstanten |  $j, d \in ME_{\langle e \rangle}$  z. B. John und Dorothy
- *Für alle einstelligen Prädikate gilt: Wenn  $j$  die vom Prädikat beschriebene Eigenschaft hat, hat  $d$  auch diese Eigenschaft.*
- Wann ist diese Wff wahr?
  - ▶ Wenn  $j$  und  $d$  **alle benennbaren Eigenschaften teilen**?
  - ▶ Eine Eigenschaft jedes Objekts |  
CF der Menge  $\{x : x \text{ is the sole member of this set}\}$  (union set)
  - ▶ Einzige Möglichkeit für Wahrheit der Wff also  $j=d$



*non* in Sätzen wie *This function is non-continuous*.

- Produktives Suffix im Englischen, wie *nicht-* im Deutschen
- Bedeutungsbeitrag | **Invertiert die CF** eines Adjektivs
- **Komplementbildung** der Ursprungsmenge in  $D_{\langle e, t \rangle}$
- Syntax und Semantik von *non*
  - ▶ Adjektiv *continuous* | Typ  $\langle e, t \rangle$
  - ▶ Typ von *non* | In: Adjektiv / Out: Adjektiv |  $\langle \langle e, t \rangle, \langle e, t \rangle \rangle$
  - ▶  $\llbracket non \rrbracket^{\mathcal{M}, g} = h$  s. t.  $h \in D_{\langle \langle e, t \rangle, \langle e, t \rangle \rangle}$  and for every  $k \in D_{\langle e, t \rangle}$  and every  $d \in D_{\langle e \rangle}$   
 $(h(k))(d) = 1$  iff  $k(d) = 0$  and  $(h(k))(d) = 0$  iff  $k(d) = 1$

Optionale Argumente wie in *I eat.* oder *Vanity kills.*

- Zweistellige Verben wie *eat* in  $ME_{\langle e, \langle e, t \rangle \rangle}$
- Aus einem zweistelligen Verb ein einstelliges machen
  - ▶ Phonologisch leere lexikalische Konstante |  $R_0 \in ME_{\langle \langle e, \langle e, t \rangle \rangle, \langle e, t \rangle \rangle}$   
Ähnlich wie lexikalische Regeln in HPSG
  - ▶ Semantik |  $\llbracket R_0 \rrbracket^{\mathcal{M}, g} = h$  s. t.  $h \in D_{\langle \langle e, \langle e, t \rangle \rangle, \langle e, t \rangle \rangle}$  and for all  $k \in D_{\langle e, \langle e, t \rangle \rangle}$  and all  $d \in D_{\langle e \rangle}$   
 $(h(k))(d) = 1$  iff there is some  $d' \in D_{\langle e \rangle}$  s. t.  $k(d')(d) = 1$

# $\lambda$ -Sprachen

Was bedeutet  $f(x) = 3x^2 + 5x + 8$  ?

- $3x^2 + 5x + 8$  ist eine Wff mit einer ungebundenen Variable.
- Die Variable wird gebunden und die Wff wird damit zur Funktion  
x wird zur Eingabevariable und muss bei Anwendung durch Eingabewert ersetzt werden.
- Außerdem wird die Funktion f genannt.

In  $\lambda$ -Notation:  $f \stackrel{def}{=} \lambda x [3x^2 + 5x + 8]$

Mit  $\lambda$  bildet man ad hoc anonyme Funktionen.

- Abstraktion über Wffs beliebiger Komplexität
- $\lambda$ -Bindung der Variable | Gebundene Variable als Eingabevariable der Funktion
- Sehr ähnlich wie Listendefinition
  - ▶ Menge |  $\{x : x \bmod 2 = 0\}$  | allgemein  $\{x : \phi\}$
  - ▶ CF dieser Menge |  $\lambda x [x \bmod 2 = 0]$  | allgemein  $\lambda x [\phi]$

## Nur wenige Erweiterungen in $L_{Type}$

- Für jede Wff  $\phi$  mit  $Ty(\phi) = \langle t \rangle$  und jede  $x \in Var$  und jede  $a \in Con$ 
  - ▶ Abstraktion |  $\phi \implies \lambda x [\phi^{a/x}]$   
Definition  $\phi^{a/x}$  | Wff  $\phi$  in der alle  $a$  durch  $x$  getauscht wurden
  - ▶ Anwendung der Funktion ( $\lambda$ -Konversion) |  $\lambda x [\phi^{a/x}] (a) = \phi$
- Es gilt  $\lambda x [\phi^{a/x}] (a) \equiv \phi$  für jede Wff  $\phi$ , jede  $a \in Con$  und jede  $x \in Var$
- $x$  kann von einem beliebigen Typ  $\sigma$  sein.
- Es gilt für  $\lambda x [\phi]$  mit  $x \in ME_{\langle \sigma \rangle}$  stets  $\phi \in ME_{\langle t \rangle}$  sowie  $\lambda x [\phi] \in ME_{\langle \sigma, t \rangle}$

## Abstraktion über Individuenvariable und Prädikatsvariable

- Individuenvariable  $x_{\langle e \rangle}$  alternativ  $v_{1_{\langle e \rangle}}$ 
  - ▶  $\lambda x_{\langle e \rangle} [L(x)]$
  - ▶ Mit  $L$  z. B. für *laughs*
  - ▶ Die CF der Menge von Individuen  $d \in D_{\langle e \rangle}$  mit die Eigenschaft  $L$  (alle Lachenden)
  - ▶ Mengendefinition dazu  $\{x : L(x)\}$
- Prädikatsvariable  $P_{\langle e, t \rangle}$  alternativ  $v_{1_{\langle e, t \rangle}}$ 
  - ▶  $\lambda P_{\langle e, t \rangle} [P(l)]$
  - ▶ Mit  $l$  z. B. für *Horst Lichter*
  - ▶ Die CF aller Eigenschaften  $k \in D_{\langle e, t \rangle}$  von  $l$  (alle Eigenschaften Horst Lichters)
  - ▶ Mengendefinition dazu  $\{P : P(l)\}$

Die vollen Regeln aus Dowty u. a. (1981: 102) (Syn C.10 and Sem 10)

- If  $\alpha \in ME_\alpha$  and  $u \in Var_b$ , then  $\lambda u [\alpha] \in ME_{\langle b, a \rangle}$ .
- If  $\alpha \in ME_a$  and  $u \in Var_b$  then  $\llbracket \lambda u [\alpha] \rrbracket^{\mathcal{M}, g}$  is that function  $h$  from  $D_b$  into  $D_a$  ( $h \in D_a^{D_b}$ ) s. t. for all objects  $k$  in  $D_b$ ,  $h(k)$  is equal to  $\llbracket \alpha \rrbracket^{\mathcal{M}, g[k/u]}$ .



## Konversionen/Reduktionen | Arten, $\lambda$ -Ausdrücke umzuschreiben

- $\alpha$ -Konversion | Umbenennung von Variablen
  - ▶  $\lambda x [\phi] \stackrel{\alpha}{\equiv} \lambda y [\phi^{[x/y]}]$  gdw  $y$  in  $\phi$  nicht vorkommt
- $\beta$ -Reduktion | Funktionsapplikation
  - ▶  $\lambda x [\phi] (a) \stackrel{\beta}{\equiv} \phi^{[x/a]}$
  - ▶ Ausdrücke mit nicht realisierten, aber möglichen  $\beta$ -Reduktionen:  $\beta$ -Redex
- $\eta$ -Reduktion | Entfernen von leeren Abstraktionen
  - ▶  $\lambda x [F(x)] \stackrel{\eta}{\equiv} F$  gdw  $Ty(F) = Ty(\lambda x [F(x)])$  (und  $x$  nicht frei in  $F$  ist)
  - ▶ Ausdruck mit nicht realisierten, aber möglichen  $\eta$ -Reduktionen:  $\eta$ -Redex
  - ▶ Mäßige Semantiker |  $\eta$ -Redex-Fetisch mit  $\lambda x \lambda y \lambda z [gibt'(x, y, z)]$  usw.

# The *non* example revised (Dowty et al., 104)

- $\forall x \forall v_{0\langle e,t \rangle} \left[ (\mathbf{non}(v_{0\langle e,t \rangle}))(x) \leftrightarrow \neg(v_{0\langle e,t \rangle}(x)) \right]$
- $\forall v_{0\langle e,t \rangle} \left[ \lambda x \left[ (\mathbf{non}(v_{0\langle e,t \rangle}))(x) \right] = \lambda x \left[ \neg(v_{0\langle e,t \rangle}(x)) \right] \right]$
- $\forall v_{0\langle e,t \rangle} \left[ \mathbf{non}(v_{0\langle e,t \rangle}) = \lambda x \left[ \neg(v_{0\langle e,t \rangle}(x)) \right] \right]$   
(since  $\lambda x [\mathbf{non}(v)(x)]$  is unnecessarily abstract/ $\eta$  reduction)
- $\lambda v_{0\langle e,t \rangle} \left[ \mathbf{non}(v_{0\langle e,t \rangle}) = \lambda v_{0\langle e,t \rangle} \left[ \lambda x \left[ \neg(v_{0\langle e,t \rangle}(x)) \right] \right] \right]$
- and since that is about all assignments for  $\lambda v_{0\langle e,t \rangle}$ :  
 $\mathbf{non} = \lambda v_{0\langle e,t \rangle} \left[ \lambda x \left[ \neg v_{0\langle e,t \rangle}(x) \right] \right]$

*Mary is non-adjacent.*

(translate 'adjacent' as  $c_{0\langle e,t \rangle}$ , 'Mary' as  $c_{0\langle e \rangle}$ , ignore the copula)

Ausblick auf Quantifikation bei Montague

# The behavior of quantified NPs

- syntactically like referential NPs
- semantically like PC quantifiers
- *Every student walks.*:  $\forall v_{0\langle e \rangle} \left[ c_{0\langle e, t \rangle}(v_{0\langle e \rangle}) \rightarrow c_{1\langle e, t \rangle}(v_{0\langle e \rangle}) \right]$
- *Some student walks.*:  $\exists v_{0\langle e \rangle} \left[ c_{0\langle e, t \rangle}(v_{0\langle e \rangle}) \wedge c_{1\langle e, t \rangle}(v_{0\langle e \rangle}) \right]$
- making referential NPs and QNPs **the same type?**

# A higher type

- $\lambda v_{0\langle e,t \rangle} \forall v_{0\langle e \rangle} \left[ c_{0\langle e,t \rangle}(v_{0\langle e \rangle}) \rightarrow v_{0\langle e,t \rangle}(v_{0\langle e \rangle}) \right]$
- a second order function
- characterizes the set of all predicates true of every student
- equally:  $\lambda v_{0\langle e,t \rangle} \exists v_{0\langle e \rangle} \left[ c_{0\langle e,t \rangle}(v_{0\langle e \rangle}) \wedge v_{0\langle e,t \rangle}(v_{0\langle e \rangle}) \right]$

# Combining with some predicate

- Wie kann man Passiv in  $L_{Type}$  modellieren?



- Chierchia, Gennaro & Sally McConnell-Ginet. 2000. *Meaning and grammar: An introduction to semantics*. 2. Aufl. Cambridge, MA: MIT Press.
- Dowty, David R., Robert E. Wall & Stanley Peters. 1981. *Introduction to Montague semantics*. Dordrecht: Kluwer.

## Kontakt

Prof. Dr. Roland Schäfer  
Institut für Germanistische Sprachwissenschaft  
Friedrich-Schiller-Universität Jena  
Fürstengraben 30  
07743 Jena

<https://rolandschaefer.net>  
[roland.schaefer@uni-jena.de](mailto:roland.schaefer@uni-jena.de)

## Creative Commons BY-SA-3.0-DE

Dieses Werk ist unter einer Creative Commons Lizenz vom Typ *Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland* zugänglich. Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie

<http://creativecommons.org/licenses/by-sa/3.0/de/> oder wenden Sie sich brieflich an Creative Commons, Postfach 1866, Mountain View, California, 94042, USA.