Semantik und Pragmatik

SS 2005

Universität Bielefeld

Teil 10, 24. Juni 2005

Gerhard Jäger

Existenzquantor: Interpretation

- Intuition:

  $$\exists v(\varphi)$$

  ist wahr, genau dann wenn für irgendeine Konstante c gilt

  $$[c/v]\varphi$$

  ist wahr

- aber:

  $$\exists x(\text{SQUEAK'}(x))$$

  ist wahr in unserem Modell, obwohl es keine Konstante c gibt, so dass folgendes wahr wäre:

  $$\text{SQUEAK'}(c)$$

Existenzquantor: Interpretation

- Problem wird ebenfalls durch Umweg über Belegungsfunktion umgangen:

$$[\exists v(\varphi)]_g^M = 1$$

genau dann wenn es ein Objekte $a \in E$ gibt, so dass

$$[\varphi]_{g[a/v]}^M = 1$$

- in dem Beispiel wäre

$$[\text{SQUEAK'}(x)]_{g[\mathbf{MOUSE}/x]}^M = 1$$

und damit die quantifizierte Formel auch wahr

Semantik der Typentheorie

Definition 1 (Interpretation der Typentheorie (dritte Version))
Sei $M = \langle E, F \rangle$ *ein Modell für die Typentheorie, und* g *eine Belegungsfunktion für* M.

- $[\alpha]_g^M = F(\alpha)$, *wenn* α *eine Konstante ist*
- $[v]_g^M = g(v)$, *wenn* v *eine Variable ist*
- $[\neg\varphi]_g^M = 1 - [\varphi]_g^M$
- $[\varphi \wedge \psi]_g^M = \min([\varphi]_g^M, [\psi]_g^M)$
- $[\varphi \vee \psi]_g^M = \max([\varphi]_g^M, [\psi]_g^M)$
- $[\varphi \to \psi]_g^M = \max(1 - [\varphi]_g^M, [\psi]_g^M)$
- $[\varphi \leftrightarrow \psi]_g^M = 1 - ([\varphi]_g^M - [\psi]_g^M)^2$
- $[\alpha(\beta)]_g^M = [\alpha]_g^M([\beta]_g^M)$
- $[\forall v_a(\varphi)]_g^M = \min(\{[\varphi]_{g[a/v]}^M | a \in Dom(a)\})$
- $[\exists v_a(\varphi)]_g^M = \max(\{[\varphi]_{g[a/v]}^M | a \in Dom(a)\})$

Beispiele

Anmerkung: wenn der Wahrheitswert einer Formel in einem Modell nicht von der Belegungsfunktion abhängt, kann die Belegungsfunktion weggelassen werden. Statt $[\varphi]_g^M$ schreibt man also einfach $[\varphi]^M$.

- $[\exists x \textsc{Animal'}(x)]^M$
- $[\exists x(\textsc{Animal'}(x) \wedge \textsc{Run'}(x))]^M$
- $[\exists x(\textsc{Animal'}(x) \rightarrow \textsc{Run'}(x))]^M$
- $[\forall x(\textsc{Animal'}(x) \rightarrow \textsc{Run'}(x))]^M$
- $[\exists x \textsc{Scream'}(x)]^M$

Beispiele

Anmerkung: wenn der Wahrheitswert einer Formel in einem Modell nicht von der Belegungsfunktion abhängt, kann die Belegungsfunktion weggelassen werden. Statt $[\varphi]_g^M$ schreibt man also einfach $[\varphi]^M$.

- $[\exists x \textsc{Animal'}(x)]^M = 1$
- $[\exists x(\textsc{Animal'}(x) \wedge \textsc{Run'}(x))]^M = 1$
- $[\exists x(\textsc{Animal'}(x) \rightarrow \textsc{Run'}(x))]^M = 1$
- $[\forall x(\textsc{Animal'}(x) \rightarrow \textsc{Run'}(x))]^M = 0$
- $[\exists x \textsc{Scream'}(x)]^M = 0$

Tautologien

Definition 2 (Tautologie) *Eine Formel φ ist eine typentheoretische **Tautologie**, formal notiert als*

$$\Rightarrow \varphi$$

genau dann wenn für alle Modelle M und alle Belegungsfunktionen g gilt:

$$[\varphi]_g^M = 1$$

Kontradiktionen

Definition 3 (Kontradiktion) *Eine Formel φ ist eine typentheoretische **Kontradiktion** genau dann wenn für alle Modelle M und alle Belegungsfunktionen g gilt:*

$$[\varphi]_g^M = 0$$

Logische Äquivalenz

Definition 4 (Logische Äquivalenz) *Zwei Formeln* φ *und* ψ *sind* ***logisch äquivalent****, formal notiert als*

$$\varphi \Leftrightarrow \psi$$

genau dann wenn für alle Modelle M *und alle Belegungsfunktionen* g *gilt:*

$$[\varphi]^M_g = [\psi]^M_g$$

Logische Folgerung

Definition 5 (Logische Folgerung) *Aus den Prämissen $\varphi_1, \ldots, \varphi_n$ folgt die Konklusion ψ logisch – formal notiert als*

$$\varphi_1 \ldots, \varphi_n \Rightarrow \psi$$

genau dann wenn für alle Modelle M und alle Belegungsfunktionen g gilt: wenn $[\varphi_i]_g^M = 1$ für alle $1 \leq i \leq n$, dann gilt auch $[\psi]_g^M = 1$.

Zeit und Tempus

- logische Quantoren werden auch, aber nicht nur zur Übersetzung von nominalen Quantoren in der natürlichen Sprache gebraucht
- weiteres linguistisches Phänomen, das als Quantifikation analysiert werden kann: **Tempus**
- Grundidee:
  - es gibt Variable und Konstante für *Zeiten*
  - n-stellige Verben werden als $n + 1$-stellige Prädikate übersetzt
  - das erste Argument eines Verbs (bzw. seiner Übersetzung) ist eine Zeit-Variable
  - Tempusmorpheme (*Präsens, Präteritum* und Zeit-Adverbien (*immer, manchmal*) drücken Quantifikation über Zeiten aus

Tempus: Beispiele

(1) Peter schlief.

- intuitive Bedeutung des Präteritum: Peters Schlaf fand zu **einem** Zeitpunkt in der Vergangenheit statt

$$\exists t(t < \textsf{NOW'} \wedge \textsf{SLEEP'}(t)(\textsf{PETER'}))$$

Tempus: Beispiele

- Bemerkungen dazu:
  - „$<$“ ist eine zweistellige Relation zwischen Zeiten
  - korrekte Notation wäre also: $<(\textsc{now}')(t)$, aber „Infix-Notation“ (Prädikationssymbol zwischen den Argumenten) ist allgemein üblich
  - intendierte Bedeutung von „$<$“ ist „liegt vor“
  - NOW’ ist eine Konstante für Zeiten
  - intendierte Bedeutung: Sprech-Zeitpunkt

Tempus: Beispiele

(1) Peter schlief immer.

$$\forall t(t < \text{NOW'} \rightarrow \text{SLEEP'}(t)(\text{PETER'}))$$

- Zeitadverb „immer“ hat ähnliche Funktion wie Quantor „alle“ $\leadsto$ beide führen Allqantor ein

Tempus: Beispiele

(1) Peter schlief gestern.

$$\exists t(t < \text{NOW'} \wedge \text{YESTERDAY'}(t) \wedge \text{SLEEP'}(t)(\text{PETER'}))$$

- Adverbien wie „gestern“ werden als einstellige Prädikate über Zeiten übersetzt

Tempus: Beispiele

$$\forall t(t < \text{NOW'} \rightarrow \text{SLEEP'}(t)(\text{PETER'}))$$

$$\Rightarrow$$

$$\exists t(t < \text{NOW'} \wedge \text{YESTERDAY'}(t) \wedge \text{SLEEP'}(t)(\text{PETER'}))$$

Also Voraussage: Aus Peter schlief immer folgt logisch Peter schlief gestern.

Tempus: Beispiele

(1) Peter wird schlafen.

$$\exists t(\text{NOW'} < t \wedge \text{SLEEP'}(t)(\text{PETER'}))$$

Tempus: Beispiele

(1) *Peter wird gestern schlafen.

- intuitiv: konfligierende Informationen
- „gestern" impliziert Vergangenheit, und Futur Zukunft
- „gestern" sollte also die Information $t <$ NOW' in die Übersetzung einführen, genau wie das Präteritums-Morphem

$$\exists t(t < \text{NOW'} \wedge \text{YESTERDAY'}(t) \wedge \text{NOW'} < t \wedge \text{SLEEP'}(t)(\text{PETER'}))$$

Tempus: Beispiele

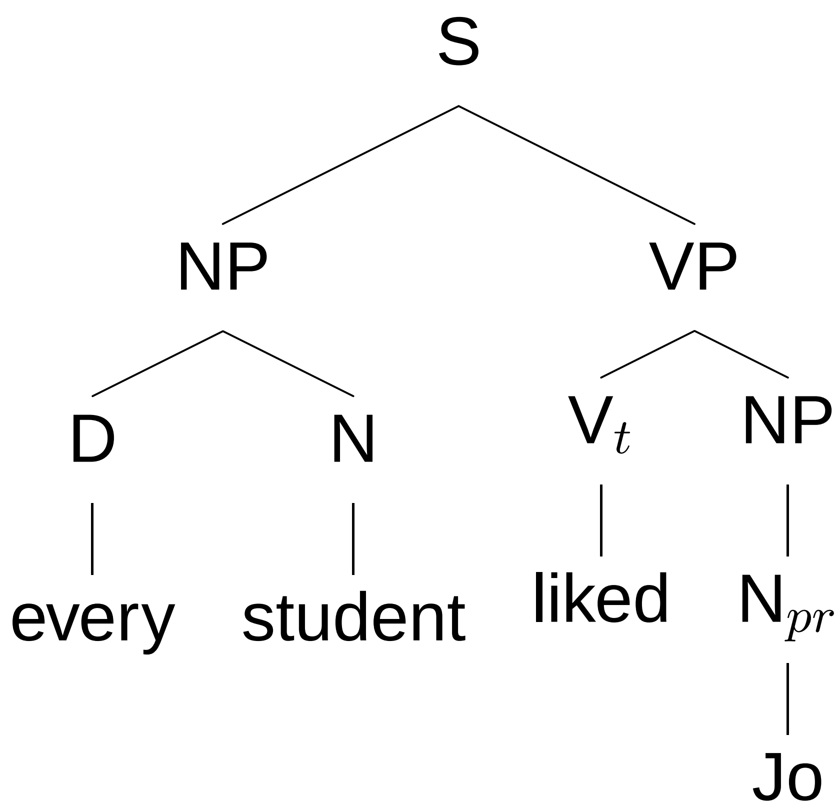
- Formel ist konsistent
- steht aber im Widerspruch zu unserem Allgemeinwissen über die Anordnung von Zeiten
- Grundannahmen über Struktur der Zeit können als **Axiome** formuliert werden, z.B.

$$\forall t \neg (t < t)$$
$$\forall t, t', t'' (t < t' \wedge t' < t'' \rightarrow t < t'')$$
$$\forall t, t' \neg (t < t' \wedge t' < t)$$

- Übersetzung von (5) steht im Widerspruch zum dritten Axiom; daher ist (5) semantisch abweichend

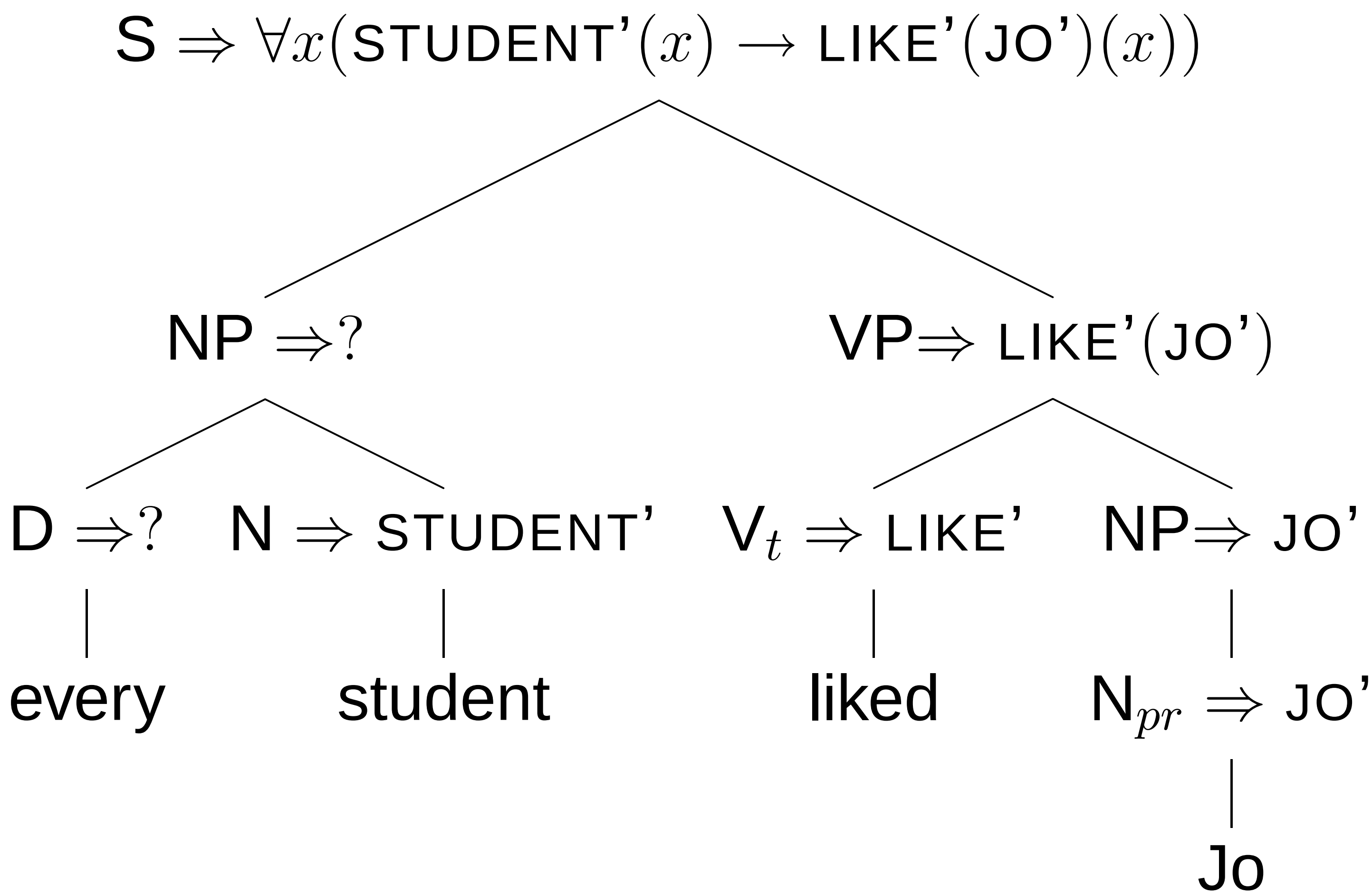
Quantifikation und Kompositionalität

(1) Every student liked Jo.



- bei kompositionaler Übersetzung sollte jeder Knoten übersetzbar sein

Quantifikation und Kompositionalität



Quantifikation und Kompositionalität

- erster Versuch: höherstufige Variablen
  - *every* $\Rightarrow \forall x(P_{\langle e,t\rangle}(x) \rightarrow Q_{\langle e,t\rangle}(x))$
  - *student* $\Rightarrow$ STUDENT'
  - *every student* $\Rightarrow \forall x(\text{STUDENT'}_{\langle e,t\rangle}(x) \rightarrow Q_{\langle e,t\rangle}(x))$
  - *liked Jo* $\Rightarrow$ LIKE'(JO')
  - *every student liked Jo*
    $\Rightarrow \forall x(\text{STUDENT'}(x) \rightarrow \text{LIKE'}(\text{JO'})(x))$

Quantifikation und Kompositionalität

- entsprechende Grammatik müsste folgendermaßen aussehen:
  - $$\begin{array}{lll} S & \rightarrow & NP, VP \\ S & \Rightarrow & [VP'/Q]NP' \end{array}$$
  - $$\begin{array}{lll} NP & \rightarrow & D, N \\ NP & \Rightarrow & [N'/P]D' \end{array}$$
- würde funktionieren, ist aber unplausibel, weil
  - die Übersetzung von *every* und *every student* dann Formeln vom Typ t wären
  - diese Ausdrücke also wahrheitswert-fähig sein müssten.

Quantifikation und Kompositionalität

- eleganter ist es, wenn Funktionsanwendung die normale semantische Operation ist
- intuitiv einleuchtende Grammatik-Regeln
  - $$\begin{array}{lcl} S & \rightarrow & NP, VP \\ S & \Rightarrow & NP'(VP') \end{array}$$
  - $$\begin{array}{lcl} NP & \rightarrow & D, N \\ NP & \Rightarrow & D'(N') \end{array}$$
- was man also bräuchte:
  - $VP' = \text{LIKE'}(\text{JO'})$
  - $NP'(\text{LIKE'}(\text{JO'})) = \forall x(\text{STUDENT'}(x) \rightarrow \text{LIKE'}(\text{JO'})(x))$

Der Lambda-Operator

- λ-Operator: Erweiterung der Typentheorie
- erlaubt es, derartige Gleichungen zu lösen:
  - $NP'(\textsf{LIKE'}(\textsf{JO'})) = \forall x(\textsf{STUDENT'}(x) \rightarrow \textsf{LIKE'}(\textsf{JO'})(x))$
  - $NP' = \lambda P \forall x(\textsf{STUDENT'}(x) \rightarrow P(x))$
- allgemeine Definition der **Lambda-Konversion:**

$$(\lambda v \varphi)(\alpha) = [\alpha/v]\varphi$$

- Seitenbedingung (spielt für unsere Anwendungen keine Rolle): alle Variablen, die auf der linke Seite der Gleichung frei sind, sind auch auf der rechten Seite frei
- also:

$$(\lambda v_t \exists x v)(P(x)) \neq \exists x P(x)$$

Typ von Lambda-Ausdrücken

- wie Quantoren hat λ einen **Skopus** und **bindet eine Variable**
- λ-Ausdruck besteht aus drei Teilen:
  - λ
  - Variable
  - Skopus des λ-Operators
- λ-Operator kreiert Funktoren:
  - $\varphi : a$
  - $v : b$
  - $\lambda v \varphi : \langle b, a \rangle$

Syntax der Typentheorie

Definition 1 (Syntax der Typentheorie, endgültige Version)

1. *Von jedem Typ gibt es unendlich viele Variablen.*
2. *Von jedem Typ gibt es unendlich viele Konstanten.*
3. *Eine Konstante oder Variable eines Typs ist ein Ausdruck dieses Typs.*
4. *Wenn φ und ψ Ausdrücke vom Typ t sind, dann sind $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi, \varphi \leftrightarrow \psi$ auch Ausdrücke vom Typ t.*
5. *Wenn α ein Ausdruck vom Typ $\langle a, b\rangle$ ist und β ein Ausdruck vom Typ a, dann ist ein Ausdruck $\alpha(\beta)$ vom Typ b.*
6. *Wenn v eine Variable ist und φ ein Ausdruck vom Typ t, dann sind $\forall v(\varphi)$ und $\exists v(\varphi)$ ebenfalls Ausdrücke vom Typ t.*
7. *Wenn α ein Ausdruck vom Typ a ist und v eine Variable vom Typ b, dann ist $\lambda v \alpha$ ein Ausdruck vom Typ $\langle b, a\rangle$.*
8. *Nichts sonst ist ein Ausdruck.*

Semantik der Typentheorie

Definition 1 (Interpretation der Typentheorie (endgültige Version))
Sei $M = \langle E, F\rangle$ ein Modell für die Typentheorie, und g eine Belegungsfunktion für M.

- $[\alpha]_g^M = F(\alpha)$, *wenn α eine Konstante ist*
- $[v]_g^M = g(v)$, *wenn v eine Variable ist*
- $[\neg\varphi]_g^M = 1 - [\varphi]_g^M$
- $[\varphi \wedge \psi]_g^M = \min([\varphi]_g^M, [\psi]_g^M)$
- $[\varphi \vee \psi]_g^M = \max([\varphi]_g^M, [\psi]_g^M)$
- $[\varphi \rightarrow \psi]_g^M = \max(1 - [\varphi]_g^M, [\psi]_g^M)$
- $[\varphi \leftrightarrow \psi]_g^M = 1 - ([\varphi]_g^M - [\psi]_g^M)^2$
- $[\alpha(\beta)]_g^M = [\alpha]_g^M([\beta]_g^M)$
- $[\forall v_a(\varphi)]_g^M = \min(\{[\varphi]_{g[l/v]}^M | l \in Dom(a)\})$
- $[\exists v_a(\varphi)]_g^M = \max(\{[\varphi]_{g[l/v]}^M | l \in Dom(a)\})$
- $[\lambda v_a(\alpha)]_g^M = \{\langle l, [\alpha]_{g[l/v]}^M\rangle | l \in Dom(a)\}$

Semantik des Lambda-Operators

Theorem 0 *Wenn durch die Lambda-Konversion keine freien Variablen gebunden werden, gilt für alle Modelle M und Belegungsfunktionen g:*

$$[(\lambda v \alpha)\beta]^M_g = [[\beta/v]\alpha]^M_g$$

Auf der Ebene der typentheoretischen Übersetzung natürlichsprachlicher Ausdrücke darf also jederzeit Lambda-Konversion durchgeführt werden, ohne dass sich dadurch an der eigentlichen Semantik etwas ändert.