Prädikate, Mengen, Funktionen

Prädikate als Mengen:

$$M_P^{\mathcal{A},g} = \{[\![x]\!]^{\mathcal{A},g} \mid [\![P(x)]\!]^{\mathcal{A},g} = 1\}$$

Mengen als Funktionen:

charakteristische Funktion f_M

$$f_M : U \to \{0,1\}, f(x) = \begin{cases} 1, \text{ wenn } x \in M \\ 0, \text{ wenn } x \notin M \end{cases}$$

Typtheorie, Motivation

Informal: Unterscheidung zwischen verschiedenen Objekten

z.B.

- Terme und Formeln, Prädikate und Funktionen mit einer, zwei oder mehr Stellen
- 'einfache' Elemente, Mengen
- Relationen in bestimmten Mengen

Sprache: Valenzrahmen von Verben:

schenken, verschenken

glauben, vermuten

Typtheorie, formal

Die Menge der *Typen* ist definiert als:

1. e ist ein Typ (Objekte)
2. t ist ein Typ (Wahrheitswerte)
3. wenn a und b Typen sind, dann auch $\langle a, b\rangle$

Die Definition kann auf n-Tupel ausgedehnt werden, dann ist es aber sinnvoller, für Funktionen einen zusätzlichen zweistelligen Typoperator einzusetzen: $(a{\rightarrow}b)$. Damit können Funktionen als spezielle Paare unterschieden werden.

Eine typisierte (formale) Sprache ist eine Sprache, bei der jedem wohlgeformten Ausdruck ein Typ zugewiesen werden kann.

D_τ bezeichnet die Menge der Denotate der Ausdrücke vom Typ τ.

Sei U eine Grundmenge, dann:

$D_e = U, D_t = \{0, 1\}$ und $D_{\langle a,b\rangle} = \{f \mid f : D_a{\rightarrow}D_b\}$

Oder, bei n-Tupeln:

$D_e = U, D_t = \{0, 1\}$, $D_{\langle a,b\rangle} = \{p \mid p \in D_a \times D_b\}$ und $D_{a{\rightarrow}b} = \{f \mid f : D_a{\rightarrow}D_b\}$

Beispiele: prädikatenlogische Formalisierung

D_e = { *Peter, Maria, ein Buch* },
Prädikate *Schenken* und *Verschenken*.

Peter schnarcht
(0) *Schnarchen*(*Peter*)

Peter verschenkt ein Buch ...
(1) *Verschenken*(*Peter*, *ein Buch*)

Maria schenkt Peter ein Buch...
(2) *Schenken*(*Maria*, *Peter*, *ein Buch*)

Typ von *Schnarchen*: $\langle e, t \rangle$

Typ von *Verschenken*: $\langle e, \langle e, t \rangle \rangle$

von *Schenken*?

Maria vermutet, dass Peter ihr ein Buch schenkt...

Maria vermutet stark, dass Peter ihr ein Buch schenkt...

Maria vermutet sehr stark, dass Peter ihr ein Buch schenkt...

Beispiele

Mengentheorie:

D_e: Menge aller Elemente und Teilmengen einer Menge M

- $x : e$
- $x \subset y : t$
- $\{x \mid x \in y\} : e$
- $\{x \mid x \subset y\} : e$
- $x \times y : \langle e, e \rangle$
- $x \in y : t$
- $f : x \rightarrow y : \langle e, e \rangle$

Typregel

Funktionsapplikation:

Sei f vom Typ $(\alpha \rightarrow \beta)$ und a vom Typ α.

Dann ist $f(a)$ vom Typ β.

Beispiele: $D_e = \mathcal{N}$ (die Menge der natürlichen Zahlen)

Sei $f, f(x) = x^2$ ... Typ von f : $\langle e, e \rangle$

Sei $f, f(x) = (x^3 \leq 0)$ ... Typ von f : $\langle e, t \rangle$

Lambda-Operator

Church (1941): Notation für Definition und Anwendung von Funktionen.

Statt “sei $f : A \to B$ mit $f(x) = x^2 + 3$ ”:

(Abstraktion) $\lambda x.x^2 + 3$

Alternative Schreibweisen: $\lambda x(x^2 + 3), \lambda x\ x^2 + 3$

Anwendung: statt f(4) (und Def. wie oben) Auswertung, Ersetzung der Variable durch das Argument

(λ-Konversion) $(\lambda x.x^2 + 3)(4) \equiv 4^2 + 3 = 19$

Der λ-Operator bindet eine Variable wie ein Quantor!

Lambda-Abstraktion, formal

Syntax

Sei x eine Variable vom Typ α,

u ein Ausdruck vom Typ β, in dem x nicht gebunden vorkommt.

Dann ist $\lambda x\ (u)$ ein Ausdruck vom Typ $(\alpha \rightarrow \beta)$.

Semantik

Sei x eine Variable vom Typ α, u ein Ausdruck vom Typ β, in dem x nicht gebunden vorkommt,

dann ist $[\![\lambda x\ (u)]\!]^{\mathcal{M},h}$ eine Funktion von D_α nach D_β mit:

für alle k in D_α: $[\![\lambda x\ (u)]\!]^{\mathcal{M},h}\ (k) = [\![u]\!]^{\mathcal{M},h'}$

und h' = h bis auf $h'(x)$, $h'(x) = k$

Lambda-Abstraktion, Beispiel

Denotation von Verben:

Bruno raucht. ... *Rauchen(Bruno)*

$\forall k : [\![\lambda x(\textit{Rauchen}(x))]\!]^{\mathcal{M},h}(k) =$
$[\![\textit{Rauchen}(x)]\!]^{\mathcal{M},h'}, h'(x) = k$

Sei $\mathcal{M} = \langle U, I \rangle$ mit:
$U = \{$*Anna, Bruno, Clara*$\}$,
$I($*Rauchen*$) = \{$Anna$\}$

$[\![\lambda x\ (\textit{Rauchen}(x))]\!]^{\mathcal{M},h}\ ([\![Bruno]\!]^{\mathcal{M},h}) = 0$

Die Sprache TL, Syntax

Typisierte Logik: versammelt Prädikatenlogik, Typen, λ-Operator und Identität:

Symbole:

- Operatoren $\wedge, \vee, \neg$
- Quantoren $\forall, \exists$
- zu jedem Typ τ eine Menge von Variablen V_τ
- zu jedem Typ τ eine Menge von Konstanten K_τ
- der Lambda-Operator λ
- das Gleichheitssymbol $=$
-

Die Sprache TL, Syntax, Ausdrücke

Die Menge der Ausdrücke vom Typ τ, E_τ:

Für alle Typen a, b

- Alle Konstanten und Variablen vom Typ τ sind Elemente von E_τ.
- Wenn $x \in E_a$ und $u \in E_b$, dann ist $\lambda x\ u$ in $E_{a \to b}$.
- Wenn $u \in E_{a \to b}$ und $x \in E_a$, dann ist $u(x)$ in E_b.
- Wenn ϕ und ψ in E_t sind (also Formeln), dann auch: $\neg\phi, \phi \vee \psi, \phi \wedge \psi$.
- Wenn $\phi \in E_t$ und x in V_a ist, dann sind $\forall x\phi$ und $\exists x\phi$ in E_t.
- Wenn u und v in E_a sind, dann ist $u = v$ in E_t.

Viel aussagekräftiger als Prädikatenlogik: Prädikate und Funktionen höherer Ordnung, Quantifikation über beliebige Ausdrücke...

Die Sprache TL, Semantik

Typen

Sei U eine Grundmenge, die ”Diskursdomäne”. Die möglichen Denotate von Ausdrücken der Typen sind:

- $D_e = U$
- $D_t = \{0, 1\}$
- $D_{\langle a,b\rangle} = \{f \mid f : D_a \rightarrow D_b\}$

Ausdrücke

Tatsächliche Denotate von Ausdrücken in einem Model $\mathcal{M} = \langle U, F\rangle$ mit einer Variablenbelegung h.

F ersetzt hier I und ist ”nur” eine Belegung der Konstanten, keine Interpretation von Prädikaten- und Funktionssymbolen mehr (die werden mit Hilfe des Lambda-Operators ”anonym” definiert)

TL, Semantik von Ausdrücken

Sei exp ein Ausdruck in TL, dann ist das Denotat von exp bez. $\mathcal{M}$ und h, $[\![exp]\!]^{\mathcal{M},h}$ wie folgt definiert:

- Sei c eine nicht-logische Konstante, dann:
  $[\![c]\!]^{\mathcal{M},h} = F(c)$.
- Sei x eine Variable, dann:
  $[\![x]\!]^{\mathcal{M},h} = g(x)$.
- (Abstraktion)
  Sei $x \in E_a$, $u \in E_b$, x nicht gebunden in u,
  dann ist $[\![\lambda x\ u]\!]^{\mathcal{M},h}$ eine Funktion von D_a nach D_b mit:
  für alle k in D_a: $[\![\lambda x\ u]\!]^{\mathcal{M},h}\ (k) = [\![u]\!]^{\mathcal{M},h'}$
  und h' = h bis auf $h'(x)$, $h'(x) = k$.
- (Konversion)
  Sei $u \in E_{a \to b}$, $x \in E_a$, dann:
  $[\![u(x)]\!]^{\mathcal{M},h} = [\![u]\!]^{\mathcal{M},h}([\![x]\!]^{\mathcal{M},h})$.

TL, Semantik von Ausdrücken II

- Seien ϕ und ψ in E_t, dann:
  $[\![\neg\phi]\!]^{\mathcal{M},h} = \neg[\![\phi]\!]^{\mathcal{M},h}$
  $[\![\phi \wedge \psi]\!]^{\mathcal{M},h} = [\![\phi]\!]^{\mathcal{M},h} \wedge [\![\psi]\!]^{\mathcal{M},h}$
  $[\![\phi \vee \psi]\!]^{\mathcal{M},h} = [\![\phi]\!]^{\mathcal{M},h} \vee [\![\psi]\!]^{\mathcal{M},h}$.
- Sei $\phi \in E_t$ und $x \in E_a$, dann:
  $[\![\forall x\phi]\!]^{\mathcal{M},h} = 1$ gdw. für alle Variablenbelegungen g, die ausser auf x mit h identisch sind, gilt:
  $[\![\phi]\!]^{\mathcal{M},g} = 1$.
  und:
  $[\![\exists x\phi]\!]^{\mathcal{M},h} = 1$ gdw. für mindestens eine Variablenbelegungen g, die ausser auf x mit h identisch sind, gilt:
  $[\![\phi]\!]^{\mathcal{M},g} = 1$.
- Seien u und v in E_a, dann:
  $[\![u = v]\!]^{\mathcal{M},g} = 1$ gdw. $[\![u]\!]^{\mathcal{M},g} = [\![v]\!]^{\mathcal{M},g}$.

Lambda-Kalkül, Λ

Weitere Anwendung des Lambda-Operators: das Lambda-Kalkül. Unterschied zu TL: keine logischen Operatoren, nicht typisiert; dafür Axiomatisierung.

Syntax von Λ:

Vokabular: Variablen x, y, λ, Klammern;

syntaktische Regeln:

- $x \in \Lambda$
- Wenn $M \in \Lambda$, dann auch $(\lambda x M)$.
- Wenn M, N in Λ, dann auch (MN).
- Wenn M, N in Λ, dann ist $M = N$ eine Formel.

Axiomatisierung

- $(\lambda x M)N = M'$, M' ist M, worin jedes Auftreten von x durch N ersetzt ist.
- $=$ ist reflexiv, symmetrisch und transitiv.
- Wenn $M = N$, dann $MZ = NZ$, $ZM = ZN$ und $(\lambda x M) = (\lambda x N)$

Linguistische Anwendungen

Kompositionale Analyse von formalen Ausdrücken höherer Ordnung
$\sim$ kompositionale Semantik natürlicher Sprache.

Beispiele:

Denotation von Verben:

intransitiv, Abb. des Subjekts auf einen Wahrheitswert:

$$\lambda P[\lambda x[P(x)]]$$

transitiv, Abb. von Subjekt und Objekt auf Ww.:

$$\lambda y \lambda x[\lambda P[P(x)]](y), \text{ oder:}$$
$$\lambda y \lambda x[\lambda P[P(x, y)]]$$

Adverben - gesuchte Funktion muss ein Verb-Denotat auf ein Verb-Denotat abbilden:

Bsp:

$$\lambda P[\lambda x[\text{schnell}(P(x))]]$$

allgemein:

$$\lambda Q[\lambda P[\lambda x[Q(P(x))]]]$$

Weitere Beispiele: Lohnstein, [PMW]

Montague: Intensionale Logik

Kombination von allen bisher eingeführten Konzepten:

- Aussagenlogik: logische Operatoren, Wahrheitswerte
- Prädikatenlogik: Terme, Quantoren
- Modallogik: mögliche Welten, Informationszustände; Operatoren $\Box$ (notwendig), $\Diamond$ (möglich)
- Temporallogik: zeitlicher Verlauf, Zeitpunkte; Operatoren **P** (*Past*), **F** (*Future*)
- Lambda-Operator: Abstraktion über Variablen beliebigen Typs
- Unterscheidung zwischen Intension und Extension

(Formalisierung siehe Lohnstein oder [PMW])

Modelle dazu: Bäume von ”einfachen Modellen”, die mit Indizes $\langle w, t \rangle$ für die ”mögliche Welt” w und den Zeitpunkt t versehen sind.

Intension und Extension

Frege: Sinn und Bedeutung

Der Abendstern, der Morgenstern vs. die Venus ...

Die Königin von England vs. Elisabeth, Victoria, ...

Intension als Funktion von Indizes
($\langle w, t \rangle$, = mögliche Welt und Zeitpunkt)
auf Menge von Extensionen (Individuen).

Neue Operatoren: Intensor $^\wedge$:

Sei α ein Ausdruck mit dem Denotat $[\![\alpha]\!]^{\mathcal{M},w,t,g}$.

Dann ist am Index $\langle w, t \rangle$ $[\![^\wedge\alpha]\!]^{\mathcal{M},w,t,g}$ die Intension von α

Extensor $^\vee$: Umkehrfunktion von $^\wedge$:

$[\![^\vee{}^\wedge\alpha]\!]^{\mathcal{M},w,t,g} = [\![\alpha]\!]^{\mathcal{M},w,t,g}$

Nicht umgekehrt: Extension gibt es nur zu bestimmtem Index, Intension ermöglicht, an jedem Index die Extension zu bestimmen.

zusätzlicher Typ s:

Sei X vom Typ a, dann ist $^\wedge X$ vom Typ $\langle s, a \rangle$.

Sei X vom Typ $\langle s, a \rangle$, dann ist $^\vee X$ vom Typ a.