Semantic Theory

Type-Thory

Manfred Pinkal
Stefan Thater

2008-04-24

Overview

- A Reminder: First-Order Predicate Logic (FOL)
- Limits of Predicate Logic
- Type Theory
- Semantics Construction

Predicate Logic – Vocabulary

- Non-logical expressions:
  - Individual constants: CON
  - n-place relation constants: $PRED^n$, for all $n \geq 0$
- Individual variables: VAR

Predicate Logic – Syntax

- Terms: TERM = VAR ∪ CON
- Atomic formulas:
  - $R(t_1, \ldots, t_n)$ for $R \in PRED^n$ and $t_1, \ldots, t_n \in$ TERM
  - $s = t$ for $s, t \in$ TERM
- Well-formed formulas: The smallest set FORM such that
  - All atomic formulas are in FORM
  - If φ, ψ are in FORM, then $\neg\varphi$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, $(\varphi \leftrightarrow \psi)$ are in FORM
  - If x is individual variable, and φ is in FORM, then $\forall x\varphi$ and $\exists x\varphi$ are in FORM

Scope

- If $\forall x\varphi$ ($\exists x\varphi$) is a subformula of a formula ψ, then we call φ the scope of this occurrence of $\forall x$ ($\exists x$) in ψ.
- We distinguish distinct occurrences of quantifiers as there are formulae like $\forall x A(x) \wedge \forall x B(x)$.
- Example:
  - $\exists x(\forall y(T(y) \leftrightarrow x=y) \wedge F(x))$

Free and Bound Variables

- An occurrence of a variable x in a formula φ is said to be free in φ if this occurrence of x does not fall within the scope of a quantifier $\forall x$ or $\exists x$ in φ.
- If $\forall x\psi$ (or $\exists x\psi$) is a subformula of φ and x is free in ψ, then this occurrence of x is said to be bound by this occurrence of the quantifier $\forall x$ (or $\exists x$).
- Examples:
  - $\forall x(A(x) \wedge B(x))$ – x occurs bound in B(x)
  - $\forall x\, A(x) \wedge B(x)$ – x occurs free in B(x)
- A sentence is a formula without free variables.

Predicate Logic – Semantics

- Expressions of Predicate Logic are interpreted relative to model structures and variable assignments.
- Model structure: $M = \langle U_M, V_M \rangle$
  - U_M is non-empty universe (individual domain)
  - V_M is an interpretation function assigning individuals ($\in U_M$) to individual constants and n-ary relations over U_M to n-place predicate symbols.
- Assignment function for variables $g: VAR \rightarrow U_M$

Predicate Logic – Semantics

- Interpretation of terms with respect to a model structure M and a variable assignment g:
  - $[\![\alpha]\!]^{M,g} = V_M(\alpha)$, if α is an individual constant
  - $[\![\alpha]\!]^{M,g} = g(\alpha)$, if α is a variable

Predicate Logic – Semantics

- Interpretation of formulas with respect to a model structure M and variable assignment g:

$$\llbracket R(t_1, \ldots, t_n) \rrbracket^{M,g} = 1 \quad \text{iff} \quad (\llbracket t_1 \rrbracket^{M,g}, \ldots, \llbracket t_n \rrbracket^{M,g}) \in V_M(R)$$
$$\llbracket s = t \rrbracket^{M,g} = 1 \quad \text{iff} \quad \llbracket s \rrbracket^{M,g} = \llbracket t \rrbracket^{M,g}$$
$$\llbracket \neg\varphi \rrbracket^{M,g} = 1 \quad \text{iff} \quad \llbracket \varphi \rrbracket^{M,g} = 0$$
$$\llbracket \varphi \wedge \psi \rrbracket^{M,g} = 1 \quad \text{iff} \quad \llbracket \varphi \rrbracket^{M,g} = 1 \text{ and } \llbracket \psi \rrbracket^{M,g} = 1$$
$$\llbracket \varphi \vee \psi \rrbracket^{M,g} = 1 \quad \text{iff} \quad \llbracket \varphi \rrbracket^{M,g} = 1 \text{ or } \llbracket \psi \rrbracket^{M,g} = 1$$
$$\llbracket \varphi \rightarrow \psi \rrbracket^{M,g} = 1 \quad \text{iff} \quad \llbracket \varphi \rrbracket^{M,g} = 0 \text{ or } \llbracket \psi \rrbracket^{M,g} = 1$$
$$\llbracket \varphi \leftrightarrow \psi \rrbracket^{M,g} = 1 \quad \text{iff} \quad \llbracket \varphi \rrbracket^{M,g} = \llbracket \psi \rrbracket^{M,g}$$
$$\llbracket \exists x\varphi \rrbracket^{M,g} = 1 \quad \text{iff} \quad \text{there is a } d \in U_M \text{ such that } \llbracket \varphi \rrbracket^{M,g[x/d]} = 1$$
$$\llbracket \forall x\varphi \rrbracket^{M,g} = 1 \quad \text{iff} \quad \text{for all } d \in U_M, \llbracket \varphi \rrbracket^{M,g[x/d]} = 1$$

- g[x/d] is the variable assignment which is identical to g except that it assigns the individual d to variable x.

Predicate Logic – Semantics

- A formula Φ is true in the model structure M iff $\llbracket \Phi \rrbracket^{M,g} = 1$ for every variable assignment g.
- A model structure M satisfies (or: is a model for) a set of formulas Γ iff every formula A ∈ Γ is true in M.
- A formula Φ is satisfiable iff there is at least one model structure M such that φ is true in M.
- A formula Φ is valid iff Φ is true in all model structures.
- A formula Φ is a contradiction iff there is no model structure M such that φ is true in M.

Entailment and Deduction

- A set of formulas Γ entails a formula Φ (notation: Γ ⊨ A) iff Φ is true in every model of Γ.
- A (sound and complete) calculus for FOL allows to prove φ from Γ iff Γ ⊨ φ by manipulating the formulas syntactically: resolution, tableaux, natural deduction, …
- Calculi can be implemented to obtain:
  - theorem provers: check entailment, validity, and unsatisfiability
  - model builders: check satisfiability, compute models
  - model checkers: determine whether model satisfies a formula

Logic and Language

- The meaning of a natural language sentence S can be approximated by the truth-conditions of S.
- We usually use logical expressions to represent the truth-conditions of natural language sentences.

Talking about students

(1) *Max is a student*

- student(max)

(2) *Not all students passed [the exam]*

- ¬∀x(student(x) → pass(x))

(3) *Only Mary flunked.*

- ¬pass(mary) ∧ ∀x(¬pass(x) → x = mary)

Limits of First-Order Logic

- Expressivity
  - non-intersective adjectives (e.g., “good”, “alleged”, …)
  - higher-order predicates (e.g., “healthy”)
  - higher-order quantification
  - …
- Semantics construction
  - We want to be able to assign semantic representations to arbitrary syntactic constituents.
  - [S [NP Peter] [VP likes Mary]] ⇒ like(p*, m*)
  - [VP likes Mary] ⇒ ???

Limits of First-Order Logic

(1) *Max is a blond thief*

- thief(max) ∧ blond(max)

(2) *Max is a good thief*

- thief(max) ∧ good(max) ?

(3) *Max is an alleged student*

- thief(max) ∧ alleged(max) ???

(4) *Max is a student*

- student(max)

- (1) + (4) entail that Max is a blond student,
- but (2) + (4) do not entail that Max is a good student.
- (3) does not even entail that Max is a thief.

Limits of First-Order Logic

(1) *Max and Mary have something in common*

(2) *Mary has all properties of a good student*

Limits of First-Order Logic

(1) *Max is always late*

(2) *Possibly, Max is a student*.

Type Theory

- The types of non-logical expressions provided by FOL are not sufficient to describe the semantic function of all natural language expressions.
- Type theory provides a much richer inventory of types: higher-order relations and functions of different kinds.

Type Theory – Types

- Basic types:
  - e (“entities”)
  - t (“truth-values”)
- Complex types:
  - If σ, τ are types, then $\langle\sigma, \tau\rangle$ is a type.
- Complex types are the type of functions mapping arguments of type σ to values of type τ.

Some useful Types

- Individual: e
- Sentence: t
- One-place predicates: $\langle e,t\rangle$
- Two-place relation: $\langle e,\langle e,t\rangle\rangle$
- Sentence adverbial: $\langle t,t\rangle$
- Attributive adjective: $\langle\langle e,t\rangle,\langle e,t\rangle\rangle$
- Degree modifier: $\langle\langle\langle e,t\rangle,\langle e,t\rangle\rangle,\langle\langle e,t\rangle,\langle e,t\rangle\rangle\rangle$

Type Theory – Vocabulary

- For every type τ a possibly empty, pairwise disjoint sets of non-logical constants CON_τ.
- For every type τ an infinite and pairwise disjoint sets of variables VAR_τ.
- The usual logical operators: $\forall$, $\exists$, $\wedge$, $\vee$, …

Type Theory – Syntax

- The sets of well-formed expressions WE_τ for every type τ are given by:

(1) $CON_\tau \subseteq WE_\tau$, for every type τ

(2) If α is in $WE_{\langle\sigma, \tau\rangle}$, β in WE_σ, then $\alpha(\beta) \in WE_\tau$.

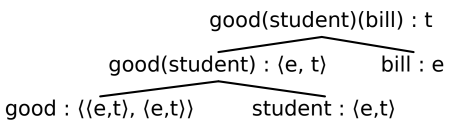
(3) If A, B are in WE_t, then $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$ are in WE_t.

(4) If A is in WE_t, then $\forall vA$ and $\exists vA$ are in WE_t, where v is a variable of arbitrary type.

(5) If α, β are well-formed expressions of the same type, then $\alpha = \beta \in WE_t$.

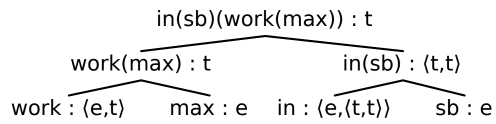
Building Well-Formed Expressions

(1) *Bill is a good student.*



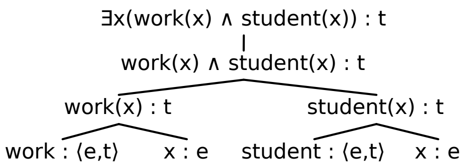
Building Well-Formed Expressions

(1) *Max works in Saarbrücken.*



Building Well-Formed Expressions

(1) *A student works.*



Type Theory – Semantics [1/3]

- Let U be a non-empty set of entities.
- The domain of possible denotations D_τ for every type τ is given by:
  (1) $D_e = U$
  (2) $D_t = \{0,1\}$
  (3) $D_{\langle\sigma,\tau\rangle}$ is the set of all functions from D_σ to D_τ

Type Theory – Semantics [2/3]

- A model structure for a type theoretic language consists of a pair M = (U, V), where
  - U (or U_M) is a non-empty domain of individuals
  - V (or V_M) is an interpretation function, which assigns to every member of CON_τ an element of D_τ.
- Variable assignment g assigns every variable of type τ a member of D_τ.

Type Theory – Semantics [3/3]

- Interpretation with respect to model structure M and variable assignment g:

$$[\![\alpha]\!]^{M,g} = V_M(\alpha), \text{ if } \alpha \text{ constant}$$
$$[\![\alpha]\!]^{M,g} = g(\alpha), \text{ if } \alpha \text{ variable}$$
$$[\![\alpha(\beta)]\!]^{M,g} = [\![\alpha]\!]^{M,g}([\![\beta]\!]^{M,g})$$
$$[\![\neg\phi]\!]^{M,g} = 1 \text{ iff } [\![\phi]\!]^{M,g} = 0$$
$$[\![\phi \wedge \psi]\!]^{M,g} = 1 \text{ iff } [\![\phi]\!]^{M,g} = 1 \text{ and } [\![\psi]\!]^{M,g} = 1, \text{ etc.}$$
$$[\![\alpha = \beta]\!]^{M,g} = 1 \text{ iff } [\![\alpha]\!]^{M,g} = [\![\beta]\!]^{M,g}$$

- if $v \in VAR_\tau$:

$$[\![\exists v\phi]\!]^{M,g} = 1 \text{ iff there is } d \in D_\tau \text{ such that } [\![\phi]\!]^{M,g[v/d]} = 1$$
$$[\![\forall v\phi]\!]^{M,g} = 1 \text{ iff for all } d \in D_\tau : [\![\phi]\!]^{M,g[v/d]} = 1$$

Type Theory

- The definition of the syntax and semantics of type theory is a straightforward extension of FOL.
- Notions like "satisfies," "valid," "satisfiable," "entailment" carry over almost verbatim from FOL.
- Type theory is sometimes called "higher-order logic:"
  - first-order logic allows quantification over individual variables (type e)
  - second-order logic allows quantification over variables of type ⟨σ, τ⟩ where σ and τ are atomic
  - ...

Characteristic Functions

- Many natural language expression have a type (σ, t).
- These types are the type of functions mapping elements of type σ to true or false.
- Such function are also known as characteristic functions, and can be thought of as subsets of D_σ (i.e., sets of σ's).
- Example: "blond" is a constant of type ⟨e,t⟩ and can be seen as characterising the set of blond individuals.

Currying

- All functional types are interpreted as one-place functions.
- How do we deal with functions with multiple arguments?
- Currying (a.k.a. "Schönfinkeln"):
  - simulate term P(a,b) as the term P(a)(b)
  - simulate type ⟨e × e, t⟩ as the type ⟨e, ⟨e,t⟩⟩.

Examples

- "Max is a student"
  - $[\![\text{student}(\text{max})]\!]^{M,g}$ = ...
- "Max is a blond student"
  - $[\![\text{blond}(\text{student})(\text{max})]\!]^{M,g}$ = ...