Semantics 1

April 26, 2012

Gerhard Jäger

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Sentence semantics

Explanatory goal

- truth conditions of declarative sentences
- meaning relations between declarative sentences
- compositional computation of sentence meanings

Sentence semantics

Truth conditions

- Wittgenstein (1922; Tractatus logico philosophicus):
  Einen Satz verstehen, heißt, wissen, was der Fall ist, wenn er wahr ist. (Man kann ihn also verstehen, ohne zu wissen, ob er wahr ist.)

Sentence semantics

Sense relations

- Entailment (If A is true, B must also be true.)
- Contradiction (A and B cannot be true at the same time.)
- Synonymy (A and B are true under exactly the same conditions.)
- (In-)Consistency (A can (not) be true.)
- Tautology (A is always true.)

Sentence semantics

Compositionality

- The meaning of a complex expression is completely determined by the meanings of its parts and the way they are combined.

Set theory and semantics

Set theory and word meanings

- simplifying assumption for the purposes of sentence semantics: meaning of a predicate is identified with the set of objects to which the predicate applies
  1. $\|$*horse*$\| = \{x|x \text{ is a horse}\}$
  2. $\|$*red*$\| = \{x|x \text{ is red}\}$
  3. $\|$*speaks*$\| = \{x|x \text{ speaks}\}$
- Hyperonymy $\approx$ subset relation

$$A \text{ is a hyperonym of } B \text{ iff } \|B\| \subseteq \|A\|$$

  - z.B. $\|$*horse*$\| \subseteq \|$*animal*$\|$

Set theory and semantics

Boolean operators

- combination of predicates via *and*, *or*, and *not* can be modeled via set theoretic operations
  - $\|\textit{round and red}\| = \|\textit{round}\| \cap \|\textit{red}\|$
  - $\|\textit{round or red}\| = \|\textit{round}\| \cup \|\textit{red}\|$
  - $\|\textit{not red}\| = \overline{\|\textit{red}\|}$
- generally:
  - $\|\alpha \textit{ and } \beta\| = \|\alpha\| \cap \|\beta\|$
  - $\|\alpha \textit{ or } \beta\| = \|\alpha\| \cup \|\beta\|$
  - $\|\textit{not } \alpha\| = \overline{\|\alpha\|}$

Set theory and semantics

Boolsche Operatoren

- set theoretic laws predict semantic equivalences (synonymies):
  - *red and round* ⇔ *round and red (commutativity)*
  - *red or round* ⇔ *round oder red (commutativity)*
  - *red and [round and soft]* ⇔ *[red and round] and soft (associativity)*
  - *red or [round or soft]* ⇔ *[red oder round] oder soft (associativity)*
  - *not [red and round]* ⇔ *[nicht red] and [nicht round] (de Morgan)*
  - *...*

Set theory and semantics

Set theory and sentence semantics

- truth condition of a sentence are **situation dependent**:
  The blackboard is clean. may be true or false, dependening on which blackborad in which room a what time is being refered to
- relativization of truth value to **situation**:
  The blackboard is clean is true in the situation s iff (if and only if) the object that is the blackboard in s is clean in s.
- Meaning of the sentencs (= truth conditions):

$$\|\textit{The blackboard is clean}\| = \{s | \textit{the blackboard in } s \textit{ is clean in } s\}$$

- generelly:

$$\|\phi\| = \{s | \phi \textit{ is true in } s\}$$

Sentence meanings are sets of situations!

Set theory and semantics

What are situations?

- Situations can be spatially and locally bounded:

  the blackboard is clean is true in s.

- Situationens can be temporally bounded and spatially unbounded

  The universe is expanding is true in s.

- some situations are both spatially and temporally unbounded

  $2 + 2 = 4$ is true in s.

Set theory and semantics

What are situations?

- situations need not be real:
  If Kennedy had not been shot, the Vietnam war would have ended in 1964 refers to a hypothetical situation where the sentence *Kennedy was shot* is false in 1964.
- Semantics deals with *possible situations*
- many authors ignore the possible boundedness of situations and use the term *possible world* (= maximal situations)
- situations in natural language semantics play a role comparable to models in propositional logic and predicate logic

Set theory and semantics

sense relations

- ϕ entails ψ (notation: $\phi \Rightarrow \psi$) iff

$$\|\phi\| \subseteq \|\psi\|$$

- ϕ and ψ are contradictory

$$\|\phi\| \cap \|\psi\| = \emptyset$$

- ϕ and ψ are equivalent (synonymous) uff

$$\|\phi\| = \|\psi\|$$

- ϕ is inconsistent: $\|\phi\| = \emptyset$
- ϕ is consistent: $\|\phi\| \neq \emptyset$
- ϕ is a tautology: $\|\phi\| = S$ (S: set of all situations)

Set theory and semantics

Boolean operations on clauses

- $\|\phi$ *and* $\psi\| = \|\phi\| \cap \|\psi\|$
- $\|\phi$ *or* $\psi\| = \|\phi\| \cup \|\psi\|$
- $\|$ *It is not the case that* $\phi\| = \overline{\|\phi\|}$

This leads to general semantic laws, such as

$$\phi \text{ and } \psi \Rightarrow \phi$$

because

$$\|\phi \text{ and } \psi\| = \|\phi\| \cap \|\psi\| \subseteq \|\phi\|$$

Set theory and semantics

functions

various ways to describe functions:

$$
\begin{array}{rl}
\|\textit{mother}\| & m : \text{persons} \to \text{persons} \\
& x \mapsto \text{the mother of } x \\
\|\textit{age}\| & a : \text{persons} \to \text{natural numbers} \\
& x \mapsto \text{the age of } x\text{, in years} \\
\|\textit{successor}\| & s : \text{natural numbers} \to \text{natural numbers} \\
& x \mapsto x + 1 \\
\|\textit{square}\| & q : \text{natural numbers} \to \text{natural numbers} \\
& x \mapsto x^2
\end{array}
$$

Set theory and semantics

functions

- algebraic notation:

$$f(x) = x^2$$

- set theoretic notation:

$$f = \{\langle x, x^2 \rangle | x \in N\}$$

Set theory and semantics

λ-notation for functions

- originates in logic and theoretica computer science
- very convenient for the purposes of linguistic semantics
- examples:
  - m : $\lambda x.$(the mother of x)
  - a : $\lambda x.$(the age of x, in years)
  - s : $\lambda x.(x+1)$
  - q : $\lambda x.(x^2)$
- such expressions are called **lambda terms**
- general format:

  λ variable.(description of the value of the variable)
- variable is place holder for argument of the function
- expression in parantheses gives recipe for computing the value of the variable
- formation of a lambda term from a description is called *lambda abstraction*

Lambda notation

computing with lambda terms

$[\lambda x.(\text{mother of } x)](\text{Isaac})$ $= \text{mother of Isaac}$ $= \text{Sarah}$	$[\lambda x.x^2](3)$ $= 3^2$ $= 9$

- General procedure:
  1. delete the λ, the variable, and the period
  2. replace all free occurrences of the variable inside the expression after the period by the argument
  3. if possible, simplify the resulting expression
- This operation is called **lambda conversion**.

Lambda notation

lambda notation with domain specification

- functions have a domain:

$$\{\langle x, x^2\rangle | x \in N\} \neq \{\langle x, x^2\rangle | x \in R\}$$

- notation $\lambda x.x^2$ is therefore incomplete
- complete notation: specification of the domain in the lambda prefix:
  - $\lambda x \in N.(x^2)$
  - $\lambda x \in R.(x^2)$
- general format:
  λ variable $\in$ domain.(description of function value)

Lambda notation

lambda notation with domain specification

- example
  - $(\lambda x \in R.(x^2 + 3x + 2))(-10) = 72$
  - $(\lambda x \in N.(x^2 + 3x + 2))(-10)$ is undefined
- domain specification and parantheses around value description are frequently omitted when no ambiguity arises

Lambda notation

variable conventions

- notation with explicit domain specification is cumbersome
- simplification via variable conventions:
  - each variable name is, by convention, associated with a certain domain:
    - $x, y, z, \ldots$: E (individuals/entities)
    - $s, s', s_1, s_2, \ldots$: S (situations)
    - $P, Q, P', \ldots : S \times E$ (relations between situations and individuals)
    - $R, S, \ldots$: $S \times E \times E$ (relations between situations and pairs of individuals)
    - $p, q, \ldots$: $POW(S)$ (sets of sets of individuals)

Lambda Notation

variable conventions

- as long as not indicated differently, it is tacitly assumed that the value of a variable falls into the corresponding domain
- for example:

$\lambda x.\phi$	abbreviates	$\lambda x \in E.\phi$
$\lambda s'.\phi$	abbreviates	$\lambda s' \in S.\phi$
$\lambda P.\phi$	abbreviates	$\lambda P \in S \times E.\phi$
$\lambda p.\phi$	abbreviates	$\lambda p \in POW(S).\phi$

etc.

Lambda Notation

functions can take other functions as arguments

- argument of a function may be complex:
  - argument is a set:
    - $\lambda X \in POW(N).(X \cap \{1,2,3\})$
    - $(\lambda X \in POW(N).(X \cap \{1,2,3\}))(\{2,3,4\}) = \{2,3,4\} \cap \{1,2,3\} = \{2,3\}$
    - $(\lambda X \in POW(N).(X \cap \{1,2,3\}))(\{4,5,6\}) = \{4,5,6\} \cap \{1,2,3\} = \emptyset$
    - $(\lambda X \in POW(N).(X \cap \{1,2,3\}))$(*Isaak*) ist nicht definiert
  - argument is also a function:
    - $\lambda f \in N \mapsto N.(f(3))$
    - $(\lambda f \in N \mapsto N.(f(3)))(\lambda x \in N.(x^2)) = (\lambda x \in N.x^2)(3) = 3^2 = 9$

Lambda Notation

functions can take other functions as arguments

further examples:

$$\begin{aligned}(\lambda f.(f(3)+f(4)))(\lambda x.x^2+x+1) &= (\lambda x.x^2+x+1)(3)+(\lambda x.x^2+x+1)(4)\\ &= 3^2+3+1+4^2+4+1\\ &= 34\end{aligned}$$

$$\begin{aligned}(\lambda f.f(f(3)-9))(\lambda x.x^2+x+1) &= (\lambda x.x^2+x+1)((\lambda x.x^2+x+1)(3)-9)\\ &= (\lambda x.x^2+x+1)((3^2+3+1)-9)\\ &= (\lambda x.x^2+x+1)(4)\\ &= 4^2+4+1\\ &= 21\end{aligned}$$

Lambda Notation

functions can have other functions as values

Likewise, the value of a function can be a function again, e.g.:

- $\lambda x \lambda y.x + y$
  - $((\lambda x(\lambda y.x + y))(2))(3) =$
  - $= (\lambda y.2 + y)(3)$
  - $= 2 + 3 = 5$
- such functions have a prefix of several lambda operators in a row
- we follow the convetions
  - lambda operators associate to the right
  - arguments associate to the left
  - lambda operators bind stronger than arguments
  - hence: first lambda belongs to first argument, second lambda to second argument etc.

Lambda Notation

functions can have other functions as values

$$(\lambda x_1. \cdots .\lambda x_n.\alpha)(a_1)\cdots(c_n)$$

abbreviates

$$(((\lambda x_1.(\cdots .(\lambda x_n.(\alpha)(a_1))))\cdots)(c_n))$$

Lambda Notation

scope, variable binding, renaming of variables

- λ operator is similar to quantifier in predicate logic in several respects
- as in predicate logic, the name of a variable is inessential:

$$\begin{aligned} \forall x(P(x) \rightarrow Q(x)) &= \forall y(P(y) \rightarrow Q(y)) \\ \lambda x.x^2 + 3x + 4 &= \lambda w.w^2 + 3w + 4 \end{aligned}$$

- it is only important which variable occurrences have the same name, and which ones have different names

Lambda Notation

characteristic functions in lambda notation

- characteristic function χ_M of a set M:
  - range: $\{0,1\}$
  - definition: $\chi_M(x) = 1$ iff $x \in M$, 0 otherwise
- meaning of sentences of the meta-language is always "true" (i.e,, 1) or "false" (i.e., 0)
- therefore the characteristic function of a set can be expressed as a λ-term:

$$\lambda x.x \in M$$

- examples:
  - suppose $M = \{x|x \textit{ is a man}\}$
  - then: $\chi_M = \lambda x.x \textit{ is a man}$

All sets can be expressed as lambda terms.

Lambda Notation

representing meanings in lambda notation

- It depends on the situation whether or not a given individual has a certain property.
- Situation dependence must be anchored in lexical meaning:
    - $\|$*horse*$\| = \lambda x \lambda s.x$ *is a horse in* s
    - $\|$*red*$\| = \lambda x \lambda s.x$ *is red in* s
    - $\|$*talks*$\| = \lambda x \lambda s.x$ *talks in* s
    - $\|$*Peter talks*$\| = \lambda s.$*Peter talks in* s