EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Philosophy
General Linguistics

Semantics & Pragmatics SoSe 2021

Lecture 6: Formal Semantics III
(Second-Order Logic)

18/05/2021, Christian Bentz

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Overview

Q&A

Tutorial 2

- *In Exercise 1d) is the difference between the future tense "will" and the present tense relevant?*

  – No. Tense can not be captured in propositional logic. The proposition p could either be "Mary will come" or "Mary comes".

- *In Exercise 1d) and 1g): why do you choose to negate the propositions separately in 1d) such that "Peter and John do not come" translates as $\neg q \wedge \neg r$, while "Nobody was willing to help or support her" gives $\neg(p \vee q)$. Follow up question: Is $\neg q \wedge \neg r$ equivalent to $\neg(p \wedge q)$?*

  These are very good questions! See the answers on the next slides.

Q&A

In order to decide whether two propositional logic formulas are equivalent, we can look at their respective truth tables:

p	q	¬p	¬q	**¬p ∧ ¬q**
1	1	0	0	**0**
1	0	0	1	**0**
0	1	1	0	**0**
0	0	1	1	**1**

p	q	¬p	¬q	**¬p ∨ ¬q**
1	1	0	0	**0**
1	0	0	1	**1**
0	1	1	0	**1**
0	0	1	1	**1**

p	q	p ∧ q	¬ **(p ∧ q)**
1	1	1	**0**
1	0	0	**1**
0	1	0	**1**
0	0	0	**1**

p	q	p ∨ q	¬ **(p ∨ q)**
1	1	1	**0**
1	0	1	**0**
0	1	1	**0**
0	0	0	**1**

Q&A

According to the truth tables above, we have that:

$\neg p \wedge \neg q \not\equiv \neg(p \wedge q)$,
$\neg p \vee \neg q \not\equiv \neg(p \vee q)$.

However:

$\neg p \wedge \neg q \equiv \neg(p \vee q)$,
$\neg p \vee \neg q \equiv \neg(p \wedge q)$.

Why do we choose $\neg p \wedge \neg q$ to represent “Peter and John do not come”?

Would you consider this statement correct if Peter comes and John does not? – Probably not. In this case it refers to both of them not comming, rather than just one of them not comming.

Why do we choose $\neg(p \vee q)$ to represent “Nobody was willing to help or support her”?

Would you consider this statement correct if somebody was willing to help her but not support her? – Probably not. Hence, this statement should be wrong, unless both help and support were consistently denied.

Q&A

Tutorial 2

- *In Exercise 1f) "might" is used in the original sentence, but not in the solutions.*

  – Yes. It is not possible to translate "might" into propositional logic. It is possible in Modal Logic (which we will briefly discuss later in the lecture series). I removed it from the formulation in the exercises to avoid confusion.

- *Construction trees: will we have to use bracket notation (in parallel to the syntax lecture series) to encode construction trees in a (potential) task in the exam?*

  – No. In this lecture series round brackets are used the way they are defined in the syntactic clauses of the respective formal semantic framework. Brackets are not used for any further tree building purpose.

Section 1: Recap of Lecture 5

The Vocabulary

Similar as for propositional logic, we can define a **language** *L* **for predicate logic**. In this case, the "vocabulary" of *L* consits of

- a (potentially infinite) supply of **constant symbols** (e.g. a, b, c, etc.),
- a (potentially infinite) supply of **variable symbols** representing the constants (e.g. x, y, z, etc.),
- a (potentially infinite) supply of **predicate symbols** (e.g. A, B, C, etc.),
- the **connectives** (e.g. $\neg$, $\wedge$, $\vee$, $\rightarrow$, etc.),
- the **quantifiers** $\forall$ and $\exists$,
- as well as the round brackets '(' and ')'.
- (The equal sign '='.)

English sentences:

(1) Socrates admires someone.

(2) Socrates is admired by someone.

(3) All teachers are friendly.

(4) Some teachers are friendly.

(5) Some friendly people are teachers.

(6) All teachers are unfriendly.

(7) Some teachers are unfriendly.

Translations:

(1) $\exists$yAsy

(2) $\exists$xAxs

(3) $\forall$x(Tx $\rightarrow$ Fx)

(4) $\exists$x(Tx $\wedge$ Fx)

(5) $\exists$x(Fx $\wedge$ Tx)

(6) $\forall$x(Tx $\rightarrow$ $\neg$Fx)

(7) $\exists$x(Tx $\wedge\neg$Fx)

Notes:
We have to add *Tx: x is a teacher* to the key.

Due to the so-called commutatitivity of $\wedge$, i.e. $\phi \wedge \psi \equiv \psi \wedge \phi$, we have that the predicate logic expressions in (4) and (5) are seen as equivalent too. However, the asymmetry might be seen as actually relevant in the natural language examples.

Generally, we have that $\forall$x$\neg\phi \equiv \neg\exists$x$\phi$.

The Syntax: Recursive Definition

Given the vocabulary of L we then define the following clauses to create formulas of L.

(i) If A is an n-ary predicate letter in the vocabulary of L, and each of $t_1, \ldots, t_n$ is a constant or a variable in the vocabulary of L, then $At_1, \ldots, t_n$ is a formula in L.

(ii) If ϕ is a formula in L, then $\neg\phi$ is too.

(iii) If ϕ and ψ are formulas in L, then $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$ are too.

(iv) If ϕ is a formula in L and x is a variable, then $\forall x\phi$ and $\exists x\phi$ are formulas in L.

(v) Only that which can be generated by the clauses (i)-(iv) in a finite number of steps is a formula in L.

Gamut, L.T.F (1991). Volume 1, p. 75.

Examples of Valid and Invalid Formulas

Formula	Rule Applied
Aa ✓	(i)
Ax ✓	(i)
Aab ✓	(i)
Axy ✓	(i)
$\neg Axy$ ✓	(i) and (ii)
$Aa \rightarrow Axy$ ✓	(i) and (iii)
$\forall x(Aa \rightarrow Axy)$ ✓	(i),(iii), and (iv)
$\forall xAa \rightarrow Axy$ ✓	(i),(iii), and (iv)
a x	–
A x	–
$\forall$ x	–
$\forall(Axy)$ x	–

Definition: *Formula* vs. *Sentence*

There is a further distinction between **formulas** and **sentences** in predicate logic. Namely, sentences are a subset of formulas for which it holds that: “A sentence is a formula in *L* which **lacks free variables**.”[1]

Gamut, L.T.F (1991). Volume 1, p. 77.

Sentence	**Not a Sentence (but Formula)**
Aa	Ax
$\forall$x(Fx)	Fx
$\forall$x(Ax $\rightarrow$ $\exists$yBy)	Ax $\rightarrow$ $\exists$yBy

[1]Free variables, in turn, are precisely defined by Gamut (1991), p.77 in their Definition 3. We will simply state here that a variable is free if it is not within the scope of a quantifier.

Interpretation Functions

"The interpretation of the constants in L will therefore be an attribution of some entity in D to each of them, that is, a function with the set of constants in L as its domain and D as its range. Such functions are called **interpretation functions**."

$$I(c) = e. \tag{1}$$

"$I(c)$ is called the *interpretation of a constant* c, or its *reference* or its *denotation*, and if e is the entity in D such that $I(c) = e$, then c is said to be one of e's names (e may have several different names)."

Gamut, L.T.F (1991). Volume 1, p. 88.

Example

$I = \{\langle m, e_1\rangle, \langle s, e_1\rangle, \langle v, e_1\rangle\}$
$I(m) = e_1$
$I(s) = e_1$
$I(v) = e_1$
Translation key: m: morning star; s: evening star; v: venus.

Definition: A model **M** for language L^2

"A model **M** for a language L of predicate logic consists of a domain D (this being a nonempty set) and an interpretation function I which [...] conforms to the following requirements:

(i) if c is a constant in L, then $I(c) \in D$;

(ii) if B is an n-ary predicate letter in L, then $I(B) \subseteq D^n$."

Gamut, L.T.F (1991). Volume 1, p. 91.

Example

$D = \{e_1, e_2, e_3\}$
$I(m) = e_1$
$I(j) = e_2$
$I(p) = e_3$
$I(S) \subset D^2$
Translation key: j: John; p: Peter; m: morning star; Sxy: x sees y.

[2] The approach we follow here is called *Approach A* or *the interpretation of quantifiers by subsitution* in Gamut (1991), p. 89. There is also another alternative Approach B, which we do not consider here.

Definition: The valuation function $\mathbf{V}_M$

"If **M** is a model for L whose interpretation function I is a function of the constants in L onto the domain D, then $\mathbf{V}_M$, the valuation V based on M, is defined as follows:"

(i) If $Aa_1, \ldots, a_n$ is an atomic sentence in L, then $\mathbf{V}_M(Aa_1, \ldots, a_n) = 1$ if and only if $\langle I(a_1), \ldots, I(a_n) \rangle \in I(A)$.

(ii) $V_M(\neg\phi) = 1$ iff $V_M(\phi) = 0$.

(iii) $V_M(\phi \wedge \psi) = 1$ iff $V_M(\phi) = 1$ and $V_M(\psi) = 1$.

(iv) $V_M(\phi \vee \psi) = 1$ iff $V_M(\phi) = 1$ or $V_M(\psi) = 1$.

(v) $V_M(\phi \to \psi) = 0$ iff $V_M(\phi) = 1$ and $V_M(\psi) = 0$.

(vi) $V_M(\phi \leftrightarrow \psi) = 1$ iff $V_M(\phi) = V_M(\psi)$.

Gamut, L.T.F (1991). Volume 1, p. 91.

Definition: The valuation function $\mathbf{V}_M$

(vii) $V_M(\forall x\phi) = 1$ iff $V_M([c/x]\phi) = 1$ for all constants c in L.

(viiii) $V_M(\exists x\phi) = 1$ iff $V_M([c/x]\phi) = 1$ for at least one constant c in L.

If $V_M(\phi) = 1$, then ϕ is said to be true in model **M**.

Gamut, L.T.F (1991). Volume 1, p. 91.

Note: The notation [c/x] means "replacing x by c". Note that this valuation works only for formulas that consist of atomic sentences and/or formulas with variables that are bound, for formulas with free variables, it does not work.

Valuation Example

Given a Model of the world **M**, consisting of D and I, and some formula ϕ which adheres to predicate logic syntax (and which consists of atomic sentences and or quantifications with bound variables), we can then evaluate the truth of ϕ as follows.

Model **M**

$D = \{e_1, e_2, e_3\}$
$I = \{\langle j, e_1\rangle, \langle p, e_2\rangle, \langle m, e_3\rangle, \langle S, \{\langle I(j), I(m)\rangle, \langle I(p), I(m)\rangle\}\rangle\}$
Translation key: j: John; p: Peter; m: morning star; Sxy: x sees y.

Valuation

"John sees the morning star": V_M(Sjm) = 1 (according to (i))
"Everybody sees the morning star": V_M(∀xSxm) = 0 (according to (vii))[3]

[3]This valuation gives 0 since the morning star (m) is a constant c in L, but it does not see itself, i.e. $\langle I(m), I(m)\rangle \notin S$.

Section 2: Beyond First-Order Logic

Beyond First-Order Logic

We have seen that predicate logic is an extension of propositional logic, by introducing predicates and quantifiers. **First-Order Predicate Logic** might itself be superseded by another logical system, called **Second-Order Predicate Logic**.

Gamut, L.T.F (1991). Volume 1, p. 168.

Take the following English sentences:

(1) Mars is red.
(2) Red is a color.
(3) Mars has a color.
(4) John has at least one thing in common with Peter.

How can we translate these into logical expressions?

The adjective “red” is a **property of individuals**. Hence, the first sentence can be straightforwardly translated into predicate logic notation as

(5) Rm (**Rx**: x is red, m: Mars)

What about the second sentence? We could stick with standard predicate notation and translate it into

(6) Cr (Cx: x is a color, **r**: red)

Note however, that now we have treated “red” once as a property of individuals in (5), and once as an individual itself in (6). In predicate logic terms it is once represented as a **predicate constant**, and once as a **constant**.

Second-Order Predicates

To circumvent this discrepancy, we can construe the predicate *x is a color* not as a property, but as a **property of properties**. $\mathcal{C}$ then represents a so-called **second-order property**, i.e. a **second-order predicate** over the first-order predicate *x is red*.

Instead of

(7) Cr (Cx: x is a color, r: red),

we then get

(8) $\mathcal{C}$R ($\mathcal{C}$X: X is a predicate with the property of being a color, Rx: x is red)

Note: We introduce **two** new sets of symbols here compared to standard predicate logic, a) the set of *second-order predicates*, and b) the set of *first-order predicate variables*. See details below.

First-Order and Second-Order Logic

A **second-order logic** language L' is then an extension to a standard predicate logic language L by adding second-order predicates to L. The original language L is then sometimes referred to as **first-order logic** language.

Further Examples:

(9) $\exists X(\mathcal{C}X \wedge Xm)$ (English sentence: “Mars has a color.”)

(10) $\exists X(Xj \wedge Xp)$ (English sentence: “John has at least one thing in common with Peter.”)

(11) $\exists \mathcal{X}(\mathcal{X}R \wedge \mathcal{X}G)$ (English sentence: “Red has something (a property) in common with green.”)

Historical Side Note

It is disputed whether second-order predicates are necessarily needed in logical systems generally, and for natural language logic in particular. Some of the reasons for this include:

- There is **no completeness theorem** for second-order logic (see Gamut 1991, Volume 1, p. 171), while for first-order logic there is.
- W. V. Quine rejected the idea that **quantification over predicates** makes sense. He conceptualized predicates as an abbreviation for an incomplete sentence, e.g. F standing for "...is friendly", and such *incomplete sentences* are not to be seen as *objects* to quantify over.

See also discussion on *https://en.wikipedia.org/wiki/Second-order_logic* under *History and disputed value*.

Section 3: The Vocabulary

Vocabulary (shared with First-Order Logic)

The vocabulary of a second-order logic language *L* consists of symbols which are *shared with first-order logic languages*, and some which need to be introduced especially to fit the *second-order properties*. The once **shared with first-order logic** languages are:

- A (potentially infinite) supply of constant symbols (e.g. a, b, c, etc.),
- a (potentially infinite) supply of variable symbols representing the constants (e.g. x, y, z, etc.),
- a (potentially infinite) supply of **first-order predicate constants** (e.g. A, B, C, etc.),
- the connectives (e.g. $\neg$, $\wedge$, $\vee$, $\rightarrow$, etc.),
- the quantifiers $\forall$ and $\exists$,
- as well as the round brackets '(' and ')'.

Vocabulary (special to Second-Order Logic)

The vocabulary extensions to fit **second-order logic requirements** are:

- A (potentially infinite) supply of **first-order predicate variables** (e.g. X, Y, Z, etc.), which are necessary to quantify over first-order predicates,
- a (potentially infinite) supply of **second-order predicate constants** (e.g. $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, etc.).

If we wanted to take it even at a higher-order level we could also have:

- a (potentially infinite) supply of **second-order predicate variables** (e.g. $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, etc.) to stand in for second-order predicates.

Example of a Translation Key

Constants	First-Order Pred.	Second-Order Pred.
j: Jumbo	B_1x: x is a bee	$\mathcal{A}X$: X is a property with the property of being an animal
s: Simba	Ex: x is an elephant	$\mathcal{C}X$: X is a property with the property of being a color
b: Bambi	Lx: x is a lion	
m: Maya	Dx: x is a deer	
	B_2: x has big ears	
	Fx: x is fast	
	Gx: x is gray	
	Yx: x is yellow	
	B_3: x is brown	
	Cxy: x chases y	

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Philosophy
General Linguistics

Section 4: The Syntax of Second-Order Logic

The Syntax: Recursive Definition

Given the vocabulary of L we then define the following clauses to create formulas of L:

(i) If A is an n-ary **first-order** predicate letter/constant in L, and $t_1, \ldots, t_n$ are individual terms in L, then $At_1, \ldots, t_n$ is an (atomic) formula in L;

(ii) If X is a [**first-order**] predicate variable and t is an individual term in L, then Xt is an atomic formula in L;

(iii) If $\mathcal{A}$ is an n-ary **second-order** predicate letter/constant in L, and $T_1, \ldots, T_n$ are **first-order unary** predicate constants, or predicate variables, in L, then $\mathcal{A}T_1, \ldots, T_n$ is an (atomic) formula in L;

(iv) If ϕ is a formula in L, then $\neg\phi$ is too;

(v) If ϕ and ψ are formulas in L, then $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, and $(\phi \leftrightarrow \psi)$ are too.

The Syntax: Recursive Definition

(vi) If x is an individual variable ϕ is a formula in L, then $\forall x \phi$ and $\exists x \phi$ are also formulas in L;

(vii) If X is a [first-order] predicate variable, and ϕ is a formula in L, then $\forall X \phi$ and $\exists X \phi$ are also formulas in L;

(viii) Only that which can be generated by the clauses (i)-(vii) in a finite number of steps is a formula in L.

Gamut, L.T.F (1991). Volume 1, p. 170.

Note: In the above clauses (i) and (ii), the word "term" is used, which has not been defined by us before. In the context here, suffices to say that it includes both constants and variables (of constants), i.e. a, b, c, etc. and x, y, z, etc.

Examples of Valid and Invalid Formulas

Formula	Rule Applied
Aa ✓	(i)
Ax ✓	(i)
Axy ✓	(i)
Xa ✓	(ii)
Xx ✓	(ii)
$\mathcal{A}$A ✓	(iii)
Xa$\rightarrow\neg$Xb ✓	(ii), (iv) and (v)
$\forall$X$\forall$x(Xa$\rightarrow$Axy) ✓	(i),(ii), (v), (vi), and (vii)
x x	–
X x	–
Xab x	–
$\forall$(Xa) x	–

Section 5: The Semantics of Second-Order Logic

The Semantics of Second-Order Logic

Similar as for the syntax of second-order logic, its semantics can also be defined based on what has been defined for first-order logic before.

For instance, just as a **first-order predicate** denotes a **set of entities**, a **second-order predicate** denotes a **set of a set of entities**.

However, since the formal definitions of valuation functions get increasingly more complex, and the interpretation with regards to natural language examples more abstract, we will not further delve into the issue here.

Gamut, L.T.F (1991). Volume 1, p. 173-174.

Summary

Summary

- **Second-order predicate logic** goes beyond first-order predicate logic by, firstly, introducing **predicate variables**, which allow to quantify over first-order predicates, and secondly, by introducing **second order predicates**, which are to be seen as properties of properties, i.e. predicates over predicates.
- These changes lead to **adjustments in the formal definitions** of the syntax and semantics of the logical language L.
- These adjustments enable the translation of a wider array of natural language sentences, although there are still natural language phenomena not captured appropriately.

References

References

Gamut, L.T.F (1991). *Logic, Language, and Meaning. Volume 1: Introduction to Logic*. Chicago: University of Chicago Press.

Kroeger, Paul R. (2019). *Analyzing meaning. An introduction to semantics and pragmatics.* Second corrected and slightly revised version. Berlin: Language Science Press.

Zimmermann, Thomas E. & Sternefeld, Wolfgang (2013). *Introduction to semantics. An essential guide to the composition of meaning.* Mouton de Gruyter.

Thank You.

Contact:

Faculty of Philosophy
General Linguistics
Dr. Christian Bentz
SFS Wihlemstraße 19-23, Room 1.24
chris@christianbentz.de
Office hours:
During term: Wednesdays 10-11am
Out of term: arrange via e-mail