



Semantics & Pragmatics SoSe 2021

Lecture 5: Formal Semantics II (Predicate Logic)

11/05/2021, Christian Bentz



Q&A

Tutorial 1: Information Theory

- ▶ *What about punctuation marks (i.e. full stop here) and upper case letters (e.g. A versus a)? Should these really constitute separate characters?*

– This depends on what the exact goal of an analysis is. For example, if you are interested to compare strings of characters in written language with other (non-)symbolic strings then, arguably, punctuation marks and upper case letters are part of the inventory of written language. On the other hand, if you take written language as an indirect representation of spoken language, then punctuation marks and upper case letters should be neutralized, since there is no correspondence in spoken language.

Having said this, in the tasks I give you it will always be clearly stated which characters to include or not.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Overview

Section 1: Recap of Lecture 4

Section 2: Historical Background

Section 3: Basic Definitions

- The Vocabulary

- Predicates and Functions

Section 4: Translation to Predicate Logic

- Translation Key

Section 5: The Syntax of Predicate Logic

- The Syntax: Recursive Definition

- Construction Trees

- Quantifier Scope

- Formula vs. Sentence

Section 6: The Semantics of Predicate Logic

- Domain of Discourse

- Interpretation Functions

- The Model M

- Valuation

Summary

References



Section 1: Recap of Lecture 4



Historical Perspective

“In the Hellenistic period, and apparently independent of Aristotle’s achievements, the logician Diodorus Cronus and his pupil Philo (see the entry Dialectical school) worked out the beginnings of a logic that took **propositions, rather than terms**,¹ as its basic elements. They influenced the second major theorist of logic in antiquity, the **Stoic Chrysippus (mid-3rd c.)**, whose main achievement is the **development of a propositional logic [...]**”

<https://plato.stanford.edu/archives/spr2016/entries/logic-ancient/>
(accessed 10/02/2021)

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

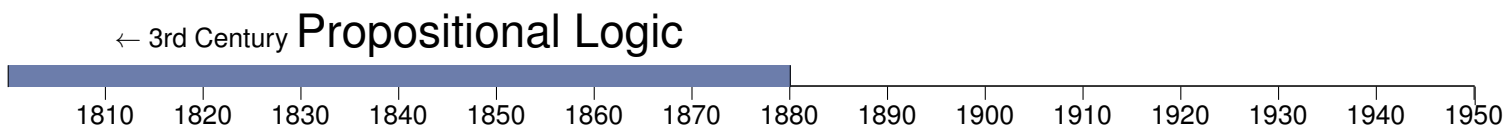
Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



¹A *term* here represents an object, a property, or an action like “Socrates” or “fall”, which cannot by itself be true or false. A *proposition* is then a combination of terms which can be assigned a truth value, e.g. “Socrates falls”.



Propositional Variables

“[...] as logical variables there are symbols which stand for statements (that is ‘propositions’). These symbols are called **propositional letters**, or **propositional variables**. In general we shall designate them by the letters p , q , and r , where necessary with subscripts as in p_1 , q_2 , r_3 , etc.”

Gamut, L.T.F (1991). Volume 1, p. 29.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Propositional Operators

We will here use the following operators (aka connectives):

Operator	Alternative Symbols	Name	English Translation
\neg	$\sim, !$	negation	<i>not</i>
\wedge	$., \&$	conjunction	<i>and</i>
\vee	$+, $	disjunction (inclusive <i>or</i>)	<i>or</i>
XOR	EOR, EXOR, $\oplus, \underline{\vee}$	exclusive <i>or</i>	<i>either ... or</i>
\rightarrow	\Rightarrow, \supset	material implication ²	<i>if ..., then</i>
\leftrightarrow	\Leftrightarrow, \equiv	material equivalence ³	<i>if, and only if ..., then</i>

Note: We will here assume that the English translations and the operators themselves are indeed equivalent in their meanings. However, in language usage, this might not actually be the case.

²aka *conditional*.

³aka *biconditional*.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Propositional Formulas

“The propositional letters and the **composite expressions** which are formed from them by means of connectives are grouped together as *sentences* or **formulas**. We designate these by means of the letters ϕ and ψ , etc. For these **metavariables**, unlike the variables p , q , and r , there is no convention that different letters must designate different formulas.”

Gamut, L.T.F (1991). Volume 1, p. 29.

Examples:

$\phi \equiv p, q, r, \text{ etc.}$

$\phi \equiv \neg p, \neg q, \neg r, \text{ etc.}$

$\phi \equiv p \wedge q, p \vee q, \text{ etc.}$

$\phi \equiv \neg(\neg p_1 \vee q_5) \rightarrow q, \text{ etc.}$

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



The Vocabulary

We can now define a **language L for propositional logic**. The “vocabulary” A of L consists of the propositional letters (e.g. p, q, r , etc.), the operators (e.g. $\neg, \wedge, \vee, \rightarrow$, etc.), as well as the round brackets ‘(’ and ‘)’. The latter are important to group certain letters and operators together. We thus have:

$$A = \{p, q, r, \dots, \neg, \wedge, \vee, \rightarrow, \dots, (,)\} \quad (1)$$

Section 1: Recap of Lecture 4

Section 2: Historical Background

Section 3: Basic Definitions

Section 4: Translation to Predicate Logic

Section 5: The Syntax of Predicate Logic

Section 6: The Semantics of Predicate Logic

Summary

References



The Syntax: Recursive Definition

Reminiscent of formal grammars of natural languages (see last years lecture on Phrase Structure Grammar), we now also need to define **syntactic rules** which allow for the symbols of the vocabulary to be combined yielding **well-formed expressions**. These rules are:

- (i) Propositional letters in the vocabulary of L are formulas in L .
- (ii) If ψ is a formula in L , then $\neg\psi$ 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.⁴
- (iv) Only that which can be generated by the clauses (i)-(iii) in a finite number of steps is a formula in L .

Gamut, L.T.F (1991). Volume 1, p. 35.

⁴We could also add the *exclusive or* here as a connective.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



The Semantics of Propositional Logic

“The valuations we have spoken of [i.e. truth valuations of formulas] can now, in the terms just introduced [i.e. functions], be described as (unary)⁵ **functions mapping formulas onto truth values**. But not every function with formulas as its domain and truth values as its range will do. A valuation must agree with the **interpretations of the connectives** which are given in their truth tables.”

Gamut, L.T.F (1991). Volume 1, p. 35.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References

⁵An *unary* function is a function with a single argument, e.g. $f(x)$. A *binary* functions could be $f(x,y)$, a *ternary* function $f(x,y,z)$, etc.



Valuation Function: Negation

Given the truth table for *negation* on the left, we get to the definition of the valuation function V on the right.⁶

ϕ	$\neg\phi$
1	0
0	1

For every valuation V and for all formulas ϕ :

$$(i) \quad V(\neg\phi) = 1 \text{ iff } V(\phi) = 0,$$

which is equivalent to

$$(i') \quad V(\neg\phi) = 0 \text{ iff } V(\phi) = 1.$$

Gamut (1991). Volume I, p. 44.

Section 1: Recap of Lecture 4

Section 2: Historical Background

Section 3: Basic Definitions

Section 4: Translation to Predicate Logic

Section 5: The Syntax of Predicate Logic

Section 6: The Semantics of Predicate Logic

Summary

References

⁶Not to be confused with the Vocabulary V defined before.



Valuation Exercise

Assume the formula for which we created a construction tree above:

$$\phi \equiv (\neg(p \vee q) \rightarrow \neg\neg\neg q) \leftrightarrow r. \quad (2)$$

What is the value assigned by $V(\phi)$ given $V(p) = 1$, $V(q) = 0$, and $V(r) = 1$?

Solution

To answer this question, the construction tree comes in handy, namely, we might want to start with valuation at the lowest level of embedding and then work our way up:

- ▶ $V(p \vee q) = 1$
- ▶ $V(\neg(p \vee q)) = 0$
- ▶ $V(\neg\neg\neg q) = 1$
- ▶ $V(\neg(p \vee q) \rightarrow \neg\neg\neg q) = 1$
- ▶ $V((\neg(p \vee q) \rightarrow \neg\neg\neg q) \leftrightarrow r) = 1$

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Beyond Propositional Logic

“The propositional logic outlined in this section is an important part of the logical metalanguage for semantic analysis, but it is not sufficient on its own because it is concerned only with **truth values** [of whole sentences]. We need a way to go beyond p and q , to represent the actual meanings of **the basic propositions** we are dealing with.”

Kroeger (2019). *Analyzing meaning*, p. 66.

Example Sentences (Set 1):

p : John is hungry.

q : John is smart.

r : John is my brother.

Example Sentences (Set 2):

p : John snores.

q : Mary sees John.

r : Mary gives George a cake.

Note: Propositional logic assigns variables (p , q , r) to whole declarative sentences, and hence is “blind” to the fact that the first set of sentences shares both the same subject, and the copula construction, whereas the second set of sentences uses predicates of different valencies and different subjects and objects.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Beyond Propositional Logic

A second major limitation of propositional logic is that it cannot take into account **quantifications**, and hence cannot decide on the truth values of the classical syllogisms below.

- (1) Premise 1: **All** men are mortal.
Premise 2: Socrates is a man.

Conclusion: Therefore, Socrates is mortal.

- (2) Premise 1: Arthur is a lawyer.
Premise 2: Arthur is honest.

Conclusion: Therefore, **some (= at least one)** lawyer is honest.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



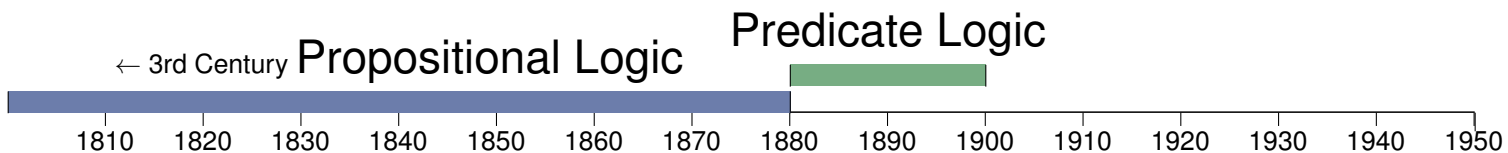
Section 2: Historical Background



Historical Background

“The first formulation of **predicate logic** can be found in Frege (1879); a similar system was developed independently by Peirce (1885). Modern versions radically differ from these ancestors in notation but not in their expressive means.”

Zimmermann & Sternefeld (2013), p. 244.



Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



“[...] fand ich ein Hindernis in der **Unzulänglichkeit der Sprache**, die bei aller entstehenden Schwerfälligkeit des Ausdruckes doch, je verwickelter die Beziehungen wurden, desto weniger die Genauigkeit erreichen liess, welche mein Zweck verlangte. Aus diesem Bedürfnisse ging der Gedanke der vorliegenden **Begriffsschrift** hervor.”

Frege (1879). Begriffsschrift: Eine der arithmetischen nachgebildete Formelsprache des reinen Denkens, p. X.

Translation: [...] I found the **inadequacy of language** to be an obstacle; no matter how unwieldy the expressions I was ready to accept, I was less and less able, as the relations became more and more complex, to attain the precision that my purpose required. This **deficiency** led me to the idea of the present **ideography**.



Section 1: Recap of Lecture 4

Section 2: Historical Background

Section 3: Basic Definitions

Section 4: Translation to Predicate Logic

Section 5: The Syntax of Predicate Logic

Section 6: The Semantics of Predicate Logic

Summary

References



Section 3: Predicate Logic (Basic Definitions)



The Vocabulary

Similar as for propositional logic, we can define a **language L for predicate logic**. In this case, the “vocabulary” of L consists 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 ‘=’).

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Quantifiers

“Standard predicate logic makes use of two quantifier symbols: the **Universal Quantifier** \forall , and the **Existential Quantifier** \exists . As the mathematical examples [below] illustrate, these quantifier symbols **must introduce a variable**, and this variable is said to be bound by the quantifier.”

Kroeger (2019) Analyzing meaning, p. 69.

Examples:

*For all x it is the case that x plus x equals x times two.
There is some y for which y plus four equals y divided by three.*

Quantifier notation:

$\forall x(x+x = 2x)$
 $\exists y(y+4 = y/3)$

Note: The **parentheses** are used here to delimit the expression that the quantifier scopes over. This follows the notation by Gamut (1991), while Kroeger (2019) would use *square brackets* here.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Predicates and Functions

The following types of symbols (sometimes referred to as “non-logical”) are relevant for our analyses:

- ▶ **Predicate symbols:** these are typically given as upper case letters, and reflect relations between n elements, where $n \geq 0$, and $n \in \mathbb{N}$ (i.e. natural numbers).⁷
- ▶ **Function symbols:** these are typically given with lower case letters (f, g , etc.), and take n variables as their arguments (similar to predicates), e.g. $f(x)$, $f(x, y)$, etc. However, Gamut (1991) use upper case letters here, remember the valuation function $V(\phi)$ from the lecture on propositional logic.

Section 1: Recap of Lecture 4

Section 2: Historical Background

Section 3: Basic Definitions

Section 4: Translation to Predicate Logic

Section 5: The Syntax of Predicate Logic

Section 6: The Semantics of Predicate Logic

Summary

References

⁷Zimmermann & Sternefeld (2013), p. 245 denote the set of all n -place predicates of a so-called *predicate logic lexicon* or *language* L as $PRED_{n,L}$.



Predicates (Notational Confusion)

Predicate symbols: these are typically given as upper case letters, and reflect relations between n elements, where $n \geq 0$, and $n \in \mathbb{N}$ (i.e. natural numbers). These are also called **n-ary** or **n-place predicate symbols**.

Examples:

Socrates snores

Peter is honest

Mary sees Peter

Mary gives Paul Lucy

Kroeger:

SNORE(s)

HONEST(p)

SEE(m,p)

GIVE(m,p,l)

Z & S:

S(s)

H(p)

S(m,p)

G(m,p,l)

Gamut:

S₁s

Hp

S₂mp₁

Gmp₂l

Note: In this lecture series, we will work with the Gamut (1991) notation, as most of the concepts and definitions here are developed according to the chapters in their introduction.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Predicates

When there is no **concrete example sentence in English** (or any other language) that a predicate logic formulation refers to, then the notation might use some upper case letter which represents **some particular predicate** which is not further defined. Gamut do not use indices in this case.

Z & S:

$P(x)$

$Q(x)$

$R(x,y)$

$S(x,y,z)$

Gamut:

Ax

Bx

Axy

Bxyz

Note: Importantly, **this is different from a predicate variable**, typically denoted by upper case X , which will become relevant in second-order logic. A variable can stand in for *any* predicate logic constant defined in the language L .

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Section 4: Translation to Predicate Logic



Beyond Propositions

Consider the following (valid) **logical inference** below. If we formulate a propositional logic language to translate these sentences, we simply represent them with propositional letters. Which, in fact, yields an invalid inference.

- (3) Premise 1: Casper is bigger than John.
Premise 2: John is bigger than Peter.

Conclusion: *Therefore, Casper is bigger than Peter.*

- (4) Premise 1: p
Premise 2: q

Conclusion: r

Gamut, L.T.F (1991). Volume 1, p. 66-67.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



The Predicate Logic Alternative

If, on the other hand, we translate these sentences into predicates reflecting the original relations between Casper, Peter, and John, then we get a better reflection of the natural language sentence structure. This can then be used to create a logically valid inference.

- (5) Premise 1: Casper is bigger than John.
Premise 2: John is bigger than Peter.

Conclusion: *Therefore, Casper is bigger than Peter.*

- (6) Premise 1: Bcj
Premise 2: Bjp

Conclusion: Bcp

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Translation Key

In order to translate a set of natural language sentences into predicate logic expressions unambiguously, we need a **translation key** listing the **predicates** and **constant symbols**.

Gamut, L.T.F (1991). Volume 1, p. 68.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References

English sentences:

- (1) John is bigger than Peter or John is bigger than Socrates.
- (2) Alcibiades does not admire himself.
- (3) If Socrates is a man, then he is mortal.
- (4) Ammerbuch lies between Tübingen and Herrenberg.
- (5) Socrates is a mortal man.

Translation key:

a_1 : Alcibiades
 a_2 : Ammerbuch
 j : John
 p : Peter
 s : Socrates
 t : Tübingen
 h : Herrenberg
 Axy : x admires y
 B_1xy : x is bigger than y
 B_2xyz : x lies between y and z
 M_1x : x is a man
 M_2x : x is mortal



Translation Examples

We can then translate the natural language sentences into predicate logic by further identifying the logical operators, i.e. connectives and negation.

Gamut, L.T.F (1991). Volume 1, p. 68.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References

English sentences:

- (1) John is bigger than Peter **or** John is bigger than Socrates.
- (2) Alcibiades does **not** admire himself.
- (3) **If** Socrates is a man, **then** he is mortal.
- (4) Ammerbuch lies between Tübingen and Herrenberg.
- (5) Socrates is a mortal man.

Translations:

- (1) $B_{1jp} \vee B_{1js}$
- (2) $\neg Aa_1a_1$
- (3) $M_1s \rightarrow M_2s$
- (4) B_2a_2th
- (5) $M_1s \wedge M_2s$



Translation with Quantifiers

If ϕ is an expression of predicate logic, then $\forall x\phi$ is called the **universal generalization** of ϕ . Likewise, $\exists x\phi$ is called the **existential generalization**.

Gamut, L.T.F (1991). Volume 1, p. 71.

Section 1: Recap of Lecture 4

Section 2: Historical Background

Section 3: Basic Definitions

Section 4: Translation to Predicate Logic

Section 5: The Syntax of Predicate Logic

Section 6: The Semantics of Predicate Logic

Summary

References

English sentences:

- (1) Everyone is friendly.
- (2) Someone is friendly.
- (3) No one is friendly.
- (4) Everyone is unfriendly.

Translations:

- (1) $\forall xFx$
- (2) $\exists xFx$
- (3) $\neg\exists xFx$
- (4) $\forall x\neg Fx$

Notes: We have to add $Fx: x \text{ is friendly}$ to the key. Someone/somebody and no one/nobody are seen as equivalent. Further, note that while we would clearly consider (3) and (4) two different English sentences, the predicate logic translations are perfectly equivalent, i.e. $\neg\exists xFx \equiv \forall x\neg Fx$.



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) $\exists y A s y$
- (2) $\exists x A x s$
- (3) $\forall x (T x \rightarrow F x)$
- (4) $\exists x (T x \wedge F x)$
- (5) $\exists x (F x \wedge T x)$
- (6) $\forall x (T x \rightarrow \neg F x)$
- (7) $\exists x (T x \wedge \neg F x)$

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References

Notes:

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

Due to the so-called commutativity 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 $\forall x \neg \phi \equiv \neg \exists x \phi$.



Section 5: The Syntax of Predicate Logic



The Syntax: Recursive Definition

Given the vocabulary of L we 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, \dots, t_n is a constant or a variable in the vocabulary of L , then At_1, \dots, 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.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Examples of **Valid** and **Invalid** Formulas

Formula	Rule Applied
Aa ✓	(i)
Ax ✓	(i)
Aab ✓	(i)
Axy ✓	(i)
Axb ✓	(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 ✗	—
A ✗	—
\forall ✗	—
$\forall(Axy)$ ✗	—

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

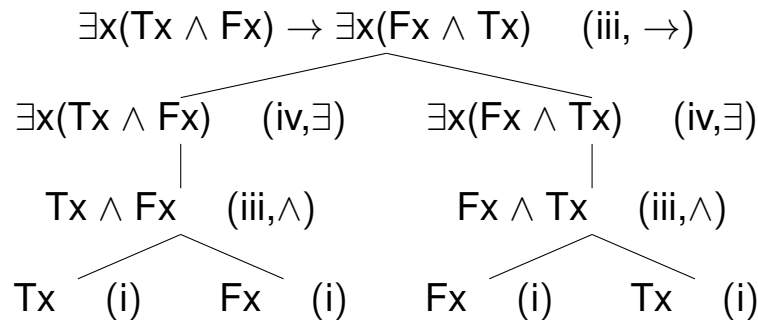
Summary

References



Building Construction Trees

Just as for propositional logic expressions, we can also **build construction trees for predicate logic expressions** (if they are correctly derived).



Note: The **number of branchings** (depth of embedding) is still given by the number of connectives. **Quantifiers behave like negation** here in the sense of creating unary branches. Also, note how we now have predicates as **atomic formulas**, i.e. *terminal symbols*, rather than single letters representing propositions.

Section 1: Recap of Lecture 4

Section 2: Historical Background

Section 3: Basic Definitions

Section 4: Translation to Predicate Logic

Section 5: The Syntax of Predicate Logic

Section 6: The Semantics of Predicate Logic

Summary

References



Definition: Quantifier Scope

“If $\forall x\psi$ is a subformula of ϕ , then ψ is called the scope of this particular occurrence of the quantifier $\forall x$ in ϕ . The same applies to occurrences of the quantifier $\exists x$.”

Gamut, L.T.F (1991). Volume 1, p. 76.

Assume $\phi \equiv \neg\exists x\exists y(\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy)$, we then have the following quantifiers and scopes for subformulas of ϕ :

Quantifier

$\exists w$

$\forall z$

$\exists y$

$\exists x$

Scope

Azw

$\exists wAzw \rightarrow Ayz$

$\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy$

$\exists y(\forall z(\exists wAzw \rightarrow Ayz) \wedge Axy)$

Note: The opening and closing brackets generally indicate the quantifier scope when connectives are involved, except for outer brackets, which can be dropped. If no connective is involved, then we don't need the brackets, e.g. $\exists wAzw$ rather than $\exists w(Azw)$.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



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**.”⁸

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

Sentence

Aa

$\forall xFx$

$\forall x(Ax \rightarrow \exists yBy)$

Not a Sentence (but Formula)

Ax

Fx

$Ax \rightarrow \exists yBy$

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References

⁸Free 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.



Section 6: The Semantics of Predicate Logic



Domain of Discourse

In predicate logic, we use quantifications as in *everybody is friendly*. This means we have to define the **domain of discourse** (D) as a set of entities (e), since statements of this type might be true or false in one domain (e.g. Hawaii), but not in another (e.g. Germany).

$$D = \{e_1, e_2, \dots, e_i\} \text{ with } D \neq \{\}. \quad (3)$$

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

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



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**.”

$$I(c_j) = e_j. \quad (4)$$

“ $I(c)$ 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.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Definition: A model \mathbf{M} for language L^9

“A model \mathbf{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

$$D = \{e_1, e_2, e_3\}$$

$$I(m) = e_1$$

$$I(j) = e_2$$

$$I(p) = e_3$$

$$I(S) \subseteq D^2$$

Translation key: j : John; p : Peter; m : morning star; Sxy : x sees y .

⁹The approach we follow here is called *Approach A* or *the interpretation of quantifiers by substitution* in Gamut (1991), p. 89. There is also another alternative Approach B, which we do not consider here.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Definition: The valuation function V_M

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

- (i) If Aa_1, \dots, a_n is an atomic sentence in L , then $V_M(Aa_1, \dots, a_n) = 1$ if and only if $\langle I(a_1), \dots, I(a_n) \rangle \in I(A)$.
- (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) $V_M(\phi \rightarrow \psi) = 0$ 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.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Definition: The valuation function V_M

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

(viii) $V_M(\exists x\phi) = 1$ 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 \mathbf{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 **sentences** of predicate logic as defined above. That is, it works for formulas that consist of atomic sentences and/or formulas with variables that are bound. For *formulas with free variables*, it does not work.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



Valuation Example

Given a Model of the world \mathbf{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 \mathbf{M}

$$D = \{e_1, e_2, e_3\}$$

$$I = \{\langle j, e_1 \rangle, \langle p, e_2 \rangle, \langle m, e_3 \rangle, \langle S, \{\langle I(j), I(m) \rangle, \langle I(p), I(m) \rangle\} \rangle\}$$

$$I(S) = \{\langle I(j), I(m) \rangle, \langle I(p), I(m) \rangle\}$$

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”: $V_M(\forall x Sxm) = 0$ (according to (vii))¹⁰

¹⁰This 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 1: Recap of Lecture 4

Section 2: Historical Background

Section 3: Basic Definitions

Section 4: Translation to Predicate Logic

Section 5: The Syntax of Predicate Logic

Section 6: The Semantics of Predicate Logic

Summary

References



Summary



Summary

- ▶ **Predicate logic** goes beyond propositional logic by, firstly, teasing apart predicates and their arguments, and secondly, introducing quantifiers.
- ▶ These changes lead to **adjustments in the formal definitions** of the syntax and semantics of the logical language L.
- ▶ While these adjustments enable a more precise translation of natural language sentences, there are also still plenty of **disagreements** with the predicate logic language L.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



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.

Section 1: Recap
of Lecture 4

Section 2:
Historical
Background

Section 3: Basic
Definitions

Section 4:
Translation to
Predicate Logic

Section 5: The
Syntax of
Predicate Logic

Section 6: The
Semantics of
Predicate Logic

Summary

References



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