



Semantics & Pragmatics SoSe 2022

Lecture 4: Formal Semantics I (Propositional Logic)



Overview

Section 1: Introduction

Historical Notes

Inference

Section 2: Basic Terminology

Propositions

Variables

Operators

Truth Tables

Section 3: The Syntax of Propositional Logic

Formulas

The Vocabulary

The Syntax: Recursive Definition

Section 4: The Semantics of Propositional Logic

Relations and Functions

Valuation

Section 5: Semantic Validity

Section 6: Beyond Propositional Logic

Summary

References



Section 1: Introduction



Historical Notes



Historical Perspective

“In the Hellenistic period, and apparently independent of Aristotle’s achievements, the logician Diodorus Cronus [died around 284 BCE at Alexandria in Egypt] 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:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

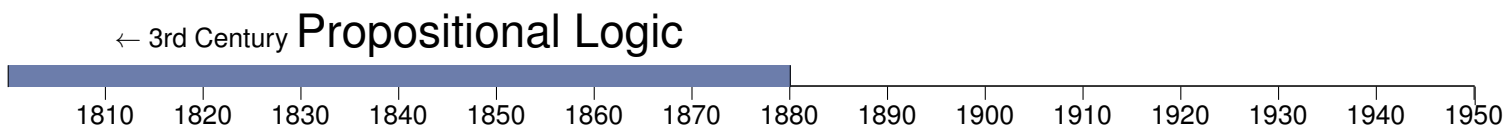
Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
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”.



Inference



The Origin of Logic in Ancient Times: Inference

“[...] knowing that one fact or set of facts is true gives us an adequate basis for concluding that some other fact is also true. **Logic** is the **science of inference**.”

Premisses: The facts which form the basis of the inference.

Conclusions: The fact which is inferred.

Kroeger (2019). *Analyzing meaning*, p. 55.

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

Conclusion: *Therefore, Socrates is mortal.*

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Syllogism

“An important variety of deductive argument in which a conclusion follows **from two or more premises**; especially the categorical syllogism.”

<http://www.philosophypages.com/dy/s9.htm#syl>

Categorical Syllogism

“A logical argument consisting of **exactly three categorical propositions, two premises and the conclusion**, with a total of exactly three categorical terms, each used in only two of the propositions.”

<http://www.philosophypages.com/dy/c.htm#casyl>

Note: The distinction between *syllogism* and *categorical syllogism* is typically dropped by logicians, and inferences drawn from premises are called syllogisms in general.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Types of Inference

There are (at least) **three types of inferences** that are relevant for analyzing sentence meanings:

- ▶ Inferences based on **content words**
- ▶ Inferences based on **logical words** (rather than content words)
- ▶ Inferences based on **quantifiers** (and logical words)

Kroeger (2019). Analyzing meaning, p. 56.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Content Word Inference

If inferences are drawn based purely on **content words**, then we are strictly speaking outside the domain of logic, since logic deals with generalizable patterns of inference, rather than ideosyncrasies of individual words and their meanings.

(2) Premise: *John **killed** the wasp.*

Conclusion: *Therefore, the wasp **died**.*

Note: The validity of the inference here depends on our understanding and definition of the words *killed* and *died*. *Kill* is typically defined as “to cause sb. or sth. to die”. Hence, the inference is valid.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Logical Word Inference

If inferences are drawn based purely on the **meaning of logical words** (operators), then the inference is generalizable to a potentially infinite number of premisses and conclusions. Note that we can replace the propositions by placeholders. Here, we are in the domain of **propositional logic**.

- (3) Premise 1: ***Either Joe is crazy or he is lying.***
Premise 2: ***Joe is not crazy.***

Conclusion: ***Therefore, Joe is lying.***

- (4) Premise 1: ***Either x or y.***
Premise 2: ***not x.***

Conclusion: ***Therefore, y.***

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Quantifier Inference

If quantifiers are used (on top of other logical operators), pure propositional logic is not sufficient anymore. We are then in the domain of **predicate logic**.

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

Conclusion: *Therefore, Socrates is mortal.*

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Why use Formal Logic?

- ▶ We might (to some degree) **overcome** *ambiguity, vagueness, indeterminacy* inherent to language (if we want to).
- ▶ Logic provides precise rules and methods to determine the **relationships between meanings of sentences** (entailments, contradictions, paraphrase, etc.).
- ▶ Systematically testing mismatches between logical inferences and speaker intuitions might help **determining the meanings of sentences**.
- ▶ Formal logic helps **modelling compositionality**.
- ▶ Formal logic is a **recursive system**, and might hence correctly model recursiveness in language.

Kroeger (2019). Analyzing meaning, p. 54.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Examples: Non-sequitur

Typical ill-formed logical arguments, for which the conclusion does not necessarily follow from the premisses (non-sequitur).

- (6) Premise 1: *We (should) ride bikes or use solar panels.*
Premise 2: *We do not ride bikes.*

Conclusion: *Therefore, we do not (need to) use solar panels.*

- (7) Premise 1: *Global warming can be caused by fluctuations in the earth's orbit or volcanic eruptions.*

Conclusion: *Therefore, global warming cannot be caused by humans.*

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Section 2: Propositional Logic



Proposition

“The meaning of a simple declarative sentence is called a **proposition**. A proposition is a claim about the world which may (in general) be true in some situations and false in others.”

Kroeger (2019), p. 35.

“To know the meaning of a [declarative] sentence is to know what the world would have to be like for the sentence to be true.”

Kroeger (2019), p. 35, citing Dowty et al. (1981: 4).

- (8) *Mary snores.*
- (9) *King Henry VIII snores.*
- (10) *The unicorn in the garden snores.*

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Formal Definition: Extension

Remember that within **denotational semantics** meaning is construed as the mapping between a given word and the real-world object it refers to (reference theory of meaning). More generally, words, phrases or sentences are said to have **extensions**, i.e. real-world situations they refer to.

Zimmermann & Sternefeld (2013), p. 71.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References

Type of expression	Type of extension	Example	Extension of example
proper name	individual	<i>Paul</i>	Paul McCartney
definite description	individual	<i>the biggest German city</i>	Berlin
noun	set of individuals	<i>table</i>	the set of tables
intransitive verb	set of individuals	<i>sleep</i>	the set of sleepers
transitive verb	set of pairs of individuals	<i>eat</i>	the set of pairs $\langle \textit{eater}, \textit{eaten} \rangle$
ditransitive verbs	set of triples of individuals	<i>give</i>	the set of triples $\langle \textit{donator}, \textit{recipient}, \textit{donation} \rangle$



Formal Definition: Extensions

“Let us denote the **extension** of an expression A by putting double brackets ‘ $\llbracket \ \rrbracket$ ’ around A , as is standard in semantics. The extension of an expression depends on the **situation s** talked about when uttering A ; so we add the index s to the closing bracket.”

Zimmermann & Sternefeld (2013), p. 85.

$\llbracket \text{Paul} \rrbracket_s = \text{Paul McCartney}^2$

$\llbracket \text{the biggest German city} \rrbracket_s = \text{Berlin}$

$\llbracket \text{table} \rrbracket_s = \{ \text{table}_1, \text{table}_2, \text{table}_3, \dots, \text{table}_n \}^3$

$\llbracket \text{sleep} \rrbracket_s = \{ \text{sleeper}_1, \text{sleeper}_2, \text{sleeper}_3, \dots, \text{sleeper}_n \}$

$\llbracket \text{eat} \rrbracket_s = \{ \langle \text{eater}_1, \text{eaten}_1 \rangle, \langle \text{eater}_2, \text{eaten}_2 \rangle, \dots, \langle \text{eater}_n, \text{eaten}_n \rangle \}$

²Zimmermann & Sternefeld just put the full proper name in brackets here, Kroeger follows another convention and just puts the first letter in lower case, e.g. $\llbracket p \rrbracket_s$.

³Kroeger (2019) uses upper case notation for both nouns and predicates, e.g. TABLE and SLEEP respectively.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Formal Definition: Frege's Generalization

“The **extension of a sentence S** is its **truth value**, i.e., 1 if S is true and 0 if S is false.”

Zimmermann & Sternefeld (2013), p. 74.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional

S_1 : The African elephant is the biggest land mamal.

$\llbracket S_1 \rrbracket_s = 1$, with s being 21st century planet earth.

$\llbracket S_1 \rrbracket_s = 0$, with s being planet earth.

S_2 : The African elephant is the biggest mamal.

$\llbracket S_2 \rrbracket_s = 0$, with s being 21st century planet earth.

$\llbracket S_2 \rrbracket_s = 0$, with s being planet earth.





Formal Definition: Proposition

“The **proposition expressed by a sentence** is the **set of possible cases [situations]** of which that sentence is true.”

Zimmermann & Sternefeld (2013), p. 141.

Coin-flip example:

situation	flip1	flip2
1	heads	heads
2	tails	tails
3	heads	tails
4	tails	heads

Sentence

S_1 : only one flip landed heads up

S_2 : all flips landed heads up

S_3 : flips landed at least once tails up

etc.

Proposition

$\llbracket S_1 \rrbracket = \{3, 4\}$

$\llbracket S_2 \rrbracket = \{1\}$

$\llbracket S_3 \rrbracket = \{2, 3, 4\}$

etc.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



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:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
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 ⁴	<i>if ..., then</i>
\leftrightarrow	\Leftrightarrow, \equiv	material equivalence ⁵	<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.

⁴aka *conditional*.

⁵aka *biconditional*.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Truth Tables

In a **truth table** we identify the extensions of (declarative) sentences as truth values. In the notation typically used, the variables p and q represent such **truth values of sentences**.⁶ The left table below gives the notation according to Zimmermann & Sternefeld, the right table according to Kroeger. We will use the latter for simplicity.

$[[S_1]]_s$	$[[S_2]]_s$	$[[S_1]]_s \wedge [[S_2]]_s$	p	q	$p \wedge q$
1	1	1	T	T	T
1	0	0	T	F	F
0	1	0	F	T	F
0	0	0	F	F	F

⁶Kroeger (2019), p. 58 and Gamut (1991), p.29 (cited above) write that p and q are variables that represent propositions. However, according to the definitions in Zimmermann & Sternefeld (given above) this is strictly speaking not correct, rather, the variables stand for extensions of sentences.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Negation

“When we have said that p and $\neg p$ must have opposite truth values in any possible situation, we have provided a definition of the **negation operator**; nothing needs to be known about the specific meaning of p .”

Kroeger (2019). *Analyzing meaning*, p. 59.

p	$\neg p$
T	F
F	T

(11) S_1 : *Peter is your child.*

$p \equiv \llbracket S_1 \rrbracket_s \in \{T, F\}$

$\neg p \equiv \neg \llbracket S_1 \rrbracket_s \in \{T, F\}$

Example: if the situation s is such that Peter is *not* the child of the person referred to as *you*, then $p \equiv \llbracket S_1 \rrbracket_s = F$, and $\neg p \equiv \neg \llbracket S_1 \rrbracket_s = T$, otherwise the other way around.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Conjunction

“In the same way, the operator \wedge ‘and’ can be defined by the truth table [below]. This table says that $p \wedge q$ (which is also sometimes written $p \& q$) is true just in case both p and q are true, and false in all other situations.”

Kroeger (2019). *Analyzing meaning*, p. 59.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

(12) S_1 : *Peter is your child.*

$p \equiv \llbracket S_1 \rrbracket_s \in \{T, F\}$

(13) S_2 : *The moon is blue.*

$q \equiv \llbracket S_2 \rrbracket_s \in \{T, F\}$

$p \wedge q \equiv (\llbracket S_1 \rrbracket_s \wedge \llbracket S_2 \rrbracket_s) \in \{T, F\}$

Example: if the situation s is such that Peter *is* the child of the person referred to as *you*, but the moon is *not* blue, then

$p \wedge q \equiv \llbracket S_1 \rrbracket_s \wedge \llbracket S_2 \rrbracket_s = F$.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Disjunction (Inclusive *or*)

“The operator \vee ‘or’ is defined by the truth table [below]. This table says that $p \vee q$ is true whenever either p is true or q is true; it is only false when both p and q are false. Notice that this *or* of standard logic is the *inclusive or*, corresponding to the English phrase *and/or*, because it includes the case where both p and q are true.”

Kroeger (2019). *Analyzing meaning*, p. 60.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

(14) S_1 : *Peter is your child.*

$p \equiv \llbracket S_1 \rrbracket_s \in \{T, F\}$

(15) S_2 : *The moon is blue.*

$q \equiv \llbracket S_2 \rrbracket_s \in \{T, F\}$

$p \vee q \equiv (\llbracket S_1 \rrbracket_s \vee \llbracket S_2 \rrbracket_s) \in \{T, F\}$

Example: if the situation s is such that *Peter is not* the child of the person referred to as *you*, but the moon *is* indeed blue, then $p \vee q \equiv \llbracket S_1 \rrbracket_s \vee \llbracket S_2 \rrbracket_s = T$.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Exclusive or

“[The table below] shows how we would define this exclusive “sense” of *or*, abbreviated here as XOR. The table says that p XOR q will be true whenever either p or q is true, but not both; it is false whenever p and q have the same truth value.”

Kroeger (2019). *Analyzing meaning*, p. 60.

p	q	p XOR q
T	T	F
T	F	T
F	T	T
F	F	F

(16) S_1 : *Peter is your child.*

$p \equiv \llbracket S_1 \rrbracket_s \in \{T, F\}$

(17) S_2 : *The moon is blue.*

$q \equiv \llbracket S_2 \rrbracket_s \in \{T, F\}$

p XOR $q \equiv (\llbracket S_1 \rrbracket_s \text{ XOR } \llbracket S_2 \rrbracket_s) \in \{T, F\}$

Example: if the situation s is such that Peter *is* the child of the person referred to as *you*, and the moon *is* indeed blue, then

p XOR $q \equiv \llbracket S_1 \rrbracket_s \text{ XOR } \llbracket S_2 \rrbracket_s = F$.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Material Implication (Conditional)

“The material implication operator \rightarrow is defined by the truth table [below]. (The formula $p \rightarrow q$ can be read as *if p (then) q*, *p only if q*, or *q if p*.) The truth table says that $p \rightarrow q$ is defined to be false just in case p is true but q is false; it is true in all other situations.”

Note: p is called the *antecedent* here, and q the *consequent*.

Kroeger (2019). *Analyzing meaning*, p. 61.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

(18) S_1 : *Peter is your child.*

$p \equiv \llbracket S_1 \rrbracket_s \in \{T, F\}$

(19) S_2 : *The moon is blue.*

$q \equiv \llbracket S_2 \rrbracket_s \in \{T, F\}$

$p \rightarrow q \equiv (\llbracket S_1 \rrbracket_s \rightarrow \llbracket S_2 \rrbracket_s) \in \{T, F\}$

Example: if the situation s is such that Peter *is* the child of the person referred to as *you*, but the moon *is not* blue, then $p \rightarrow q \equiv \llbracket S_1 \rrbracket_s \rightarrow \llbracket S_2 \rrbracket_s = F$. In all other situations, it is T.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Material Equivalence (Biconditional)

“The formula $p \leftrightarrow q$ (read as *p if and only if q*) is a short-hand or abbreviation for: $(p \rightarrow q) \wedge (q \rightarrow p)$. The **biconditional** operator is defined by the truth table [below].”

Kroeger (2019). *Analyzing meaning*, p. 61.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

(20) S_1 : *Peter is your child.*

$p \equiv \llbracket S_1 \rrbracket_s \in \{T, F\}$

(21) S_2 : *The moon is blue.*

$q \equiv \llbracket S_2 \rrbracket_s \in \{T, F\}$

$p \leftrightarrow q \equiv (\llbracket S_1 \rrbracket_s \leftrightarrow \llbracket S_2 \rrbracket_s) \in \{T, F\}$

Example: if the situation s is such that *Peter is* the child of the person referred to as *you*, and the moon *is* blue, or if both is *not* the case, then $p \leftrightarrow q \equiv \llbracket S_1 \rrbracket_s \leftrightarrow \llbracket S_2 \rrbracket_s = T$. In all other situations, it is F.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Building Truth Tables

We will follow the following four steps to analyze the sentence below:

1. Identify the **logical words** and translate them into **logical operators**
2. **Decompose the sentence** into its component declarative parts and assign **variables** to them (i.e. p and q).
3. Translate the whole sentence into **propositional logic notation**
4. Start the truth table with the variables (i.e. p and q) **to the left**, and then add operators step by step (from the most embedded to the outer layers).

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References

Example Sentence: *If the president is either crazy or he is lying, and it turns out he is lying, then he is not crazy.*



Section 3: The Syntax of Propositional Logic



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:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
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:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
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 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.⁷
- (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.

⁷We could also add the *exclusive or* here as a connective.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



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

Formula	Rule Applied
p ✓	(i)
$\neg\neg\neg q$ ✓	(i) and (ii)
$((\neg p \wedge q) \vee r)$ ✓	(i), (ii), and (iii)
$((\neg(p \vee q) \rightarrow \neg\neg\neg q) \leftrightarrow r)$ ✓	(i), (ii), and (iii)
pq ✗	—
$\neg(\neg\neg p)$ ✗	—
$\wedge p\neg q$ ✗	—
$\neg((p \wedge q \rightarrow r))$ ✗	—

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Building Unique Construction Trees

Similar to Phrase Structure Grammars (PSG), we can build **complex expressions** in a propositional logic language L . Here are some parallels and differences:

- ▶ L has a **vocabulary** A . The propositional letters would correspond to the terminal symbols in a PSG.
- ▶ The **operators** in the vocabulary A are associated with branchings in the tree. In a PSG, the re-write operator ‘ \rightarrow ’ also creates branchings. The brackets in A represent branchings, and are the same as for the bracket notation of PSGs.
- ▶ The **clauses** (i)-(iv) are similar to a set of rewrite rules.
- ▶ The **metavariables** ϕ and ψ are akin to non-terminal symbols, but we will leave them out here, as this would further complicate the tree building.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Example

Assume we want to check whether the formula⁸

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

is a valid expression in L . We therefore have to check whether rewrite steps down to the propositional letters adhere to clauses (i)-(iii). It is useful to follow the following steps:

- ▶ Determine the **depth of embedding** of the formula. This corresponds to the **number of operators** in the formula.⁹
- ▶ Check the **number of negations**. This number corresponds to the number of **unary branches**, since negation applies recursively to the same formula.
- ▶ Start with the **highest level of embedding** as the first split, and go from there.

⁸By convention, we leave away the outermost brackets of such formulas.

⁹Alternatively, the number of opening/closing brackets -1 , since we drop the outer brackets. This number corresponds to the number of binary branchings in the tree.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

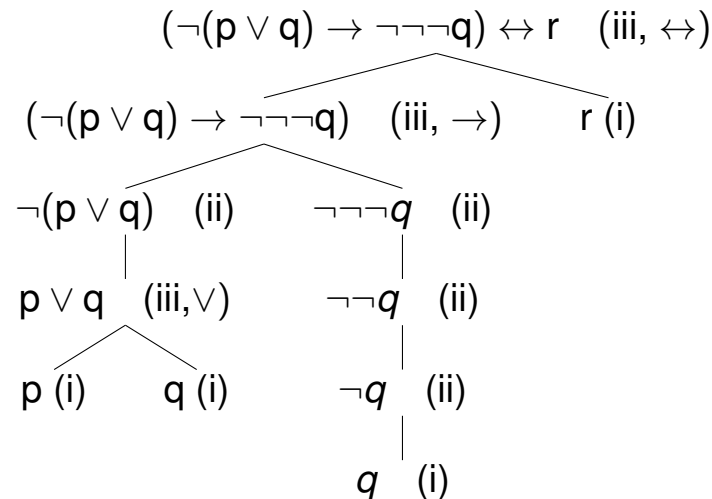
Section 6:
Beyond
Propositional
Logic

Summary

References



Example



Note: The level of embedding is 3 here. The *biconditional* (\leftrightarrow) constitutes the highest level of embedding, the *conditional* (\rightarrow) the middle level, the *or-statement* (\vee) the lowest level. Importantly, on the right of each formula in the tree, we note in parentheses which clause licenses the formula. In the case of operator application, we also give the operator for completeness, e.g. (iii, \leftrightarrow).

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Section 4: The Semantics of Propositional Logic



Meaning as the Valuation of Truth

The **semantics of a propositional language L** consists of the **valuation of the truthfulness** of simple and complex expressions derived via the syntax of L . In practice, this is typically done by means of using a truth table (see also last years lecture on propositional logic.) However, to further understand the formal underpinnings of truth-table evaluation, we first need to introduce further concepts, such as **relations** and **functions**.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Relation

“A **set of ordered pairs** is called a **relation**. The **domain** of the relation is the set of all the first elements of each pair and its **range** is the set of all the second elements.”

Kroeger (2019), p. 234.

Examples:

$$A = \{\langle a, 3 \rangle, \langle f, 4 \rangle, \langle c, 6 \rangle, \langle a, 7 \rangle\} \quad (3)$$

$$B = \{\langle 2, 3 \rangle, \langle 3, 2 \rangle, \langle 4, 7 \rangle, \langle 5, 2 \rangle, \langle 6, 7 \rangle, \langle 7, 4 \rangle\} \quad (4)$$

Both sets A and B are relations.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Function

“A set of ordered pairs defines a **mapping**, or correspondence, from the domain onto the range [...] A **function** is a relation (= a set of ordered pairs) in which each element of the **domain is mapped to a single, unique value in the range.**”

Kroeger (2019), p. 235.

Invalid

$$A(a) = 3$$

$$A(a) = 7$$

$$A(c) = 6$$

$$A(f) = 4$$

Valid

$$B(2) = 3$$

$$B(3) = 2$$

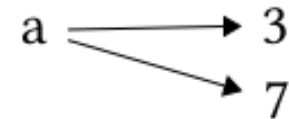
$$B(4) = 7$$

$$B(5) = 2$$

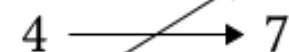
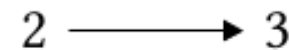
$$B(6) = 7$$

$$B(7) = 4$$

a. Set A



b. Set B



Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Examples of Functions

Notation ¹⁰	Function	Domain	Range
$D(x)$ or $d(x)$	Date of birth of x	People	Dates
$M(x)$ or $m(x)$	Mother of x	People	People
$\neg x$	Negation of x	Formulas	Formulas
$S(x, y)$ or $s(x, y)$	Sum of x and y	Numbers	Numbers
$T(x, y, z)$ or $t(x, y, z)$	Time at which the last train from x via y to z departs	Stations	Time

Note: “Mother of x ” or “father of x ” are valid functions, since there is only one mother and one father that can be assigned to an individual x . However, “brother of x ” and “sister of x ” are not valid functions, since the same individual x might have different brothers and sisters.

¹⁰The letters are arbitrarily chosen here to reflect the first letter of the function explanation. Otherwise, f , g , h , etc. are typically used. Upper and lower case is also a matter of convention.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
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)¹¹ **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:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References

¹¹An *unary* function is a function with a single argument, e.g. $f(x)$. A *binary* function 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.¹²

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:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References

¹²Not to be confused with the Vocabulary V defined before.



Valuation Function: Conjunction

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

ϕ	ψ	$\phi \wedge \psi$
1	1	1
1	0	0
0	1	0
0	0	0

For every valuation V and
for all formulas ϕ :

$$(ii) \quad V(\phi \wedge \psi) = 1 \text{ iff } V(\phi) = 1 \text{ and } V(\psi) = 1.$$

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

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Valuation Function: Disjunction (inclusive *or*)

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

ϕ	ψ	$\phi \vee \psi$
1	1	1
1	0	1
0	1	1
0	0	0

For every valuation V and
for all formulas ϕ :

$$(iii) \quad V(\phi \vee \psi) = 1 \text{ iff } V(\phi) = 1 \text{ or } V(\psi) = 1.$$

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

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Valuation Function: Material Implication (Conditional)

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

ϕ	ψ	$\phi \rightarrow \psi$
1	1	1
1	0	0
0	1	1
0	0	1

For every valuation V and
for all formulas ϕ :

$$(iv) \quad V(\phi \rightarrow \psi) = 0 \text{ iff } V(\phi) = 1 \text{ and } V(\psi) = 0.$$

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

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Valuation Function: Material Equivalence (Biconditional)

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

ϕ	ψ	$\phi \leftrightarrow \psi$
1	1	1
1	0	0
0	1	0
0	0	1

For every valuation V and
for all formulas ϕ :

$$(v) \quad V(\phi \leftrightarrow \psi) = 1 \text{ iff } V(\phi) = V(\psi).$$

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

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Valuation Exercise

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

$$\phi \equiv (\neg(p \vee q) \rightarrow \neg\neg\neg q) \leftrightarrow r. \quad (5)$$

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:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Valuation Functions and Truth Tables

Note that **valuation functions** and **truth tables** are intimately related. Namely, application of valuation functions is just a more formalized way of determining truth values of complex propositional logic formulas. The arguments of evaluation functions correspond to the formulas given in truth table columns.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Section 5: Semantic Validity



Examples: Non-sequitur

Typical ill-formed logical arguments, for which the conclusion does not necessarily follow from the premisses (non-sequitur).

- (22) Premise 1: *We (should) ride bikes or use solar panels.*
Premise 2: *We do not ride bikes.*

Conclusion: *Therefore, we do not (need to) use solar panels.*

- (23) Premise 1: *Global warming can be caused by fluctuations in the earth's orbit or volcanic eruptions.*

Conclusion: *Therefore, global warming cannot be caused by humans.*

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Semantic Validity

For formulas $\phi_1, \dots, \phi_n, \psi$ in propositional logic
 $\phi_1, \dots, \phi_n \models \psi$ ¹³ holds just in case for all valuations V such
that $V(\phi_1) = \dots = V(\phi_n) = 1, V(\psi) = 1$.¹⁴

Gamut (1991). Volume I, p. 117.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References

¹³The symbol \models in propositional and predicate logic means “models” or “semantically entails”.

¹⁴The reference to a model world \mathbf{M} is skipped here, since we haven’t defined it yet.



Example: Checking Semantic Validity

- (24) Premise 1: *We (should) ride bikes or use solar panels.*
 Premise 2: *We do not ride bikes.*

Conclusion: *Therefore, we do not (need to) use solar panels.*

p	q	$p \vee q$	$\neg p$	/	$\neg q$
1	1	1	0		0
1	0	1	0		1
0	1	1	1	*	0
0	0	0	1		1

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Section 6: Beyond Propositional Logic



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:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
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.

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

Conclusion: Therefore, Socrates is mortal.

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

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

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Summary

- ▶ In the **formal definition** of a propositional logic language L we have a “**syntax**” and a “**semantics**” part.
- ▶ The syntax consists of a set of **propositional letters**, **operators** (connectives), and **brackets**. These constitute the **vocabulary** of L . It further includes **clauses**, i.e. “rewrite rules” on how to combine symbols in an acceptable way to yield **formulas**, which are represented by metavariables.
- ▶ The **semantics** consists of the definition of a **valuation function** V , which takes formulas as its domain, and the truth values $[0,1]$ as its range. The valuation function hence maps formulas to truth values.

Section 1:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
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:
Introduction

Section 2: Basic
Terminology

Section 3: The
Syntax of
Propositional
Logic

Section 4: The
Semantics of
Propositional
Logic

Section 5:
Semantic Validity

Section 6:
Beyond
Propositional
Logic

Summary

References



Thank You.

Contact:

Faculty of Philosophy

General Linguistics

Dr. Christian Bentz

SFS Wilhelmstraße 19-23, Room 1.15

chris@christianbentz.de

Office hours:

During term: Wednesdays 10-11am

Out of term: arrange via e-mail