EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Philosophy
General Linguistics

Semantics & Pragmatics SoSe 2020

Lecture 4: Formal Semantics I (Propositional Logic)

30/04/2020, Christian Bentz

Overview

Section 1: Introduction

"[...] 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**.

BEGRIFFSSCHRIFT,

EINE DER ARITHMETISCHEN NACHGEBILDETE

FORMELSPRACHE

DES REINEN DENKENS.

VON

Dr. GOTTLOB FREGE,

PRIVATDOCENTEN DER MATHEMATIK AN DER UNIVERSITÄT JENA.

HALLE a/S.

VERLAG VON LOUIS NEBERT.

1879.

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.).
- Sytematically 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.

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*.

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Faculty of Philosophy
General Linguistics

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).

(2) *Mary snores.*

(3) *King Henry VIII snores.*

(4) *The unicorn in the garden snores.*

Formal Definition: Extension

Remember from Lecture 1 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.

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

Formal Definition: Extensions

"Let us denote the **extension** of an expression *A* by putting double brackets '$[\![\,]\!]$' 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.

$[\![\text{Paul}]\!]_s$ = Paul McCartney[1]
$[\![\text{the biggest German city}]\!]_s$ = Berlin
$[\![\text{table}]\!]_s = \{\text{table}_1, \text{table}_2, \text{table}_3, \ldots, \text{table}_n\}$[2]
$[\![\text{sleep}]\!]_s = \{\text{sleeper}_1, \text{sleeper}_2, \text{sleeper}_3, \ldots, \text{sleeper}_n\}$
$[\![\text{eat}]\!]_s = \{\langle\text{eater}_1, \text{eaten}_1\rangle, \langle\text{eater}_2, \text{eaten}_2\rangle, \ldots, \langle\text{eater}_n, \text{eaten}_n\rangle\}$

[1] Zimmermann & 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. $[\![\text{p}]\!]_s$.

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

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.

S_1: The African elephant is the biggest land mamal.
$[\![S_1]\!]_s = 1$, with s being 21st century earth.

S_2: The coin flip landed heads up.
$[\![S_2]\!]_s = 1$, with s being a particular coin flip.

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	**Proposition**
S_1: only one flip landed heads up	$[\![S_1]\!] = \{3, 4\}$
S_2: all flips landed heads up	$[\![S_2]\!] = \{1\}$
S_3: flips landed at least once tails up	$[\![S_3]\!] = \{2, 3, 4\}$
etc.	etc.

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.

Propositional Operators

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

Operator	Alternative Symbols	Name	English Translation
$\neg$	$\sim$, !	negation	*not*
$\wedge$	., &	conjunction	*and*
$\vee$	+, $\mid\mid$	disjunction (inclusive *or*)	*or*
XOR	EOR, EXOR, $\oplus$, $\veebar$	exclusive *or*	*either ... or*
$\rightarrow$	$\Rightarrow$, $\supset$	material implication[3]	*if ..., then*
$\leftrightarrow$	$\Leftrightarrow$, $\equiv$	material equivalence[4]	*if, and only if ..., then*

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.

[3]aka *conditional*.
[4]aka *biconditional*.

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.**[5] The left table below gives the notation according to Zimmermann & Sternefeld, the right table according to Kroeger. We will use the latter for simplicity.

$[\![S_1]\!]_s$	$[\![S_2]\!]_s$	$[\![S_1]\!]_s \wedge [\![S_2]\!]_s$
1	1	1
1	0	0
0	1	0
0	0	0

p	q	p∧q
T	T	T
T	F	F
F	T	F
F	F	F

[5] Kroeger (2019), p. 58 writes that *p* and *q* are variables that represent propositions. However, according to the definitions we have given above this is strictly speaking not correct.

Negation

"When we have said that p and ¬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	¬p
T	F
F	T

(5) S_1: *Peter is your child.*
$p \equiv [\![S_1]\!]_s \in \{T, F\}$
$\neg p \equiv \neg[\![S_1]\!]_s \in \{T, F\}$
Example: if the situation *s* is such that Peter is *not* the child of the person referred to as *you*, then $p \equiv [\![S_1]\!]_s$ = F, and $\neg p \equiv \neg[\![S_1]\!]_s$ = T, otherwise the other way around.

Conjunction

"In the same way, the operator ∧ 'and' can be defined by the truth table [below]. This table says that p∧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∧q
T	T	T
T	F	F
F	T	F
F	F	F

(6) S_1: *Peter is your child.*
$p \equiv [\![S_1]\!]_s \in \{T, F\}$

(7) S_2: *The moon is blue.*
$p \equiv [\![S_2]\!]_s \in \{T, F\}$

$p \wedge q \equiv [\![S_1]\!]_s \wedge [\![S_2]\!]_s \in \{T, F\}$

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 $p \wedge q \equiv [\![S_1]\!]_s \wedge [\![S_2]\!]_s = F$.

Disjunction (Inclusive *or*)

"The operator ∨ 'or' is defined by the truth table [below]. This table says that p∨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∨q
T	T	T
T	F	T
F	T	T
F	F	F

(8) S_1: *Peter is your child.*
$\mathrm{p} \equiv [\![S_1]\!]_s \in \{T, F\}$

(9) S_2: *The moon is blue.*
$\mathrm{p} \equiv [\![S_2]\!]_s \in \{T, F\}$

$\mathrm{p} \vee \mathrm{q} \equiv ([\![S_1]\!]_s \vee [\![S_2]\!]_s) \in \{T, F\}$

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 $\mathrm{p} \vee \mathrm{q} \equiv [\![S_1]\!]_s \vee [\![S_2]\!]_s = \mathrm{T}$.

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

(10) S_1: *Peter is your child.*
$p \equiv [\![S_1]\!]_s \in \{T, F\}$

(11) S_2: *The moon is blue.*
$p \equiv [\![S_2]\!]_s \in \{T, F\}$

$$p \text{ XOR } q \equiv ([\![S_1]\!]_s \text{ XOR } [\![S_2]\!]_s) \in \{T, F\}$$

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
$p \text{ XOR } q \equiv [\![S_1]\!]_s \text{ XOR } [\![S_2]\!]_s = F$.

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

(12) S_1: *Peter is your child.*
$p \equiv [\![S_1]\!]_s \in \{T, F\}$

(13) S_2: *The moon is blue.*
$p \equiv [\![S_2]\!]_s \in \{T, F\}$

$p \rightarrow q \equiv ([\![S_1]\!]_s \rightarrow [\![S_2]\!]_s) \in \{T, F\}$

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 $p \rightarrow q \equiv [\![S_1]\!]_s \rightarrow [\![S_2]\!]_s$ = F. In all other situations, it is T.

Material Equivalence (Biconditional)

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

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

p	q	p ↔ q
T	T	T
T	F	F
F	T	F
F	F	T

(14) S_1: *Peter is your child.*
$p \equiv [\![S_1]\!]_s \in \{T, F\}$

(15) S_2: *The moon is blue.*
$p \equiv [\![S_2]\!]_s \in \{T, F\}$

$$p \leftrightarrow q \equiv ([\![S_1]\!]_s \leftrightarrow [\![S_2]\!]_s) \in \{T, F\}$$

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 $p \leftrightarrow q \equiv [\![S_1]\!]_s \leftrightarrow [\![S_2]\!]_s$ = T. In all other situations, it is F.

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).

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

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Faculty of Philosophy
General Linguistics

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:

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

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

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

$\phi \equiv \neg(\neg\mathrm{p}_1 \vee \mathrm{q}_5) \rightarrow \mathrm{q}$, etc.

The Vocabulary

We can now define a **language L for propositional logic**. The "vocabulary" A of L consits 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:

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

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.[6]

(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.

[6] We could also add the *exclusive or* here as a connective.

Examples of Valid and Invalid Formulas

Formula	Rule Applied
p ✓	(i)
¬¬¬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 x	–
$\neg(\neg\neg p)$ x	–
$\wedge p \neg q$ x	–
$\neg((p \wedge q \rightarrow r))$ x	–

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.

Example

Assume we want to check whether the formula[7]

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

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.[8]
- 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.

[7] By convention, we leave away the outermost brackets of such formulas.

[8] Alternatively, 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.

Example

- $(\neg(p \vee q) \rightarrow \neg\neg\neg q) \leftrightarrow r$ (iii, $\leftrightarrow$)
  - $(\neg(p \vee q) \rightarrow \neg\neg\neg q)$ (iii, $\rightarrow$)
    - $\neg(p \vee q)$ (ii)
      - $p \vee q$ (iii, $\vee$)
        - p (i)
        - q (i)
    - $\neg\neg\neg q$ (ii)
      - $\neg\neg q$ (ii)
        - $\neg q$ (ii)
          - q (i)
  - r (i)

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 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**.

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:

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

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

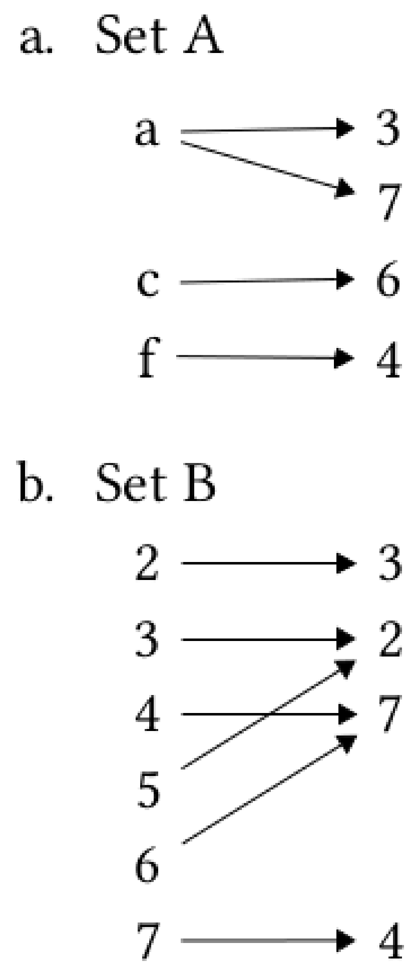
Both sets A and B are relations.

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	**Valid**
A(a) = 3	B(2) = 3
A(a) = 7	B(3) = 2
A(c) = 6	B(4) = 7
A(f) = 4	B(5) = 2
	B(6) = 7
	B(7) = 4



Examples of Functions

Notation [9]	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.

[9] The letters are arbitrarly 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.

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)[10] **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.

[10] An *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.[11]

ϕ	$\neg\phi$
1	0
0	1

For every valuation V and for all formulas ϕ:

(i) $V(\neg\phi) = 1$ iff $V(\phi) = 0$,

which is equivalent to

(i') $V(\neg\phi) = 0$ iff $V(\phi) = 1$.

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

[11] Not 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) $V(\phi \wedge \psi) = 1$ iff $V(\phi) = 1$ and $V(\psi) = 1$.

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

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

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

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) $V(\phi \rightarrow \psi) = 0$ iff $V(\phi) = 1$ and $V(\psi) = 0$.

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

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) $V(\phi \leftrightarrow \psi) = 1$ iff $V(\phi) = V(\psi)$.

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

Valuation Exercise

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

$$\phi \equiv (\neg(\mathrm{p} \vee \mathrm{q}) \rightarrow \neg\neg\neg\mathrm{q}) \leftrightarrow \mathrm{r}. \tag{5}$$

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

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:

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

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.

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Faculty of Philosophy
General Linguistics

Section 5: 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.

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.

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

Conclusion: *Therefore, Socrates is mortal*.

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

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

Summary

- In the **formal definition** of a propositional logic language L we have a "**syntax**" and a "**semantics**" part.
- The syntax consits 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.

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Philosophy
General Linguistics

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