Chapter udf

Syntax and Semantics

This is a very quick summary of definitions only. It should be expanded to provide a gentle intro to proofs by induction on formulas, with lots more examples.

syn.1 Introduction

pl:syn:int:sec Propositional logic deals with formulas that are built from propositional variables using the propositional connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. Intuitively, a propositional variable p stands for a sentence or proposition that is true or false. Whenever the "truth value" of the propositional variable in a formula is determined, so is the truth value of any formulas formed from them using propositional connectives. We say that propositional logic is *truth functional*, because its semantics is given by functions of truth values. In particular, in propositional logic we leave out of consideration any further determination of truth and falsity, e.g., whether something is necessarily true rather than just contingently true, or whether something is known to be true, or whether something is true now rather than was true or will be true. We only consider two truth values true ($\mathbb{T}$) and false ($\mathbb{F}$), and so exclude from discussion the possibility that a statement may be neither true nor false, or only half true. We also concentrate only on connectives where the truth value of a formula built from them is completely determined by the truth values of its parts (and not, say, on its meaning). In particular, whether the truth value of conditionals in English is truth functional in this sense is contentious. The material conditional $\rightarrow$ is; other logics deal with conditionals that are not truth functional.

In order to develop the theory and metatheory of truth-functional propositional logic, we must first define the syntax and semantics of its expressions. We will describe one way of constructing formulas from propositional variables using the connectives. Alternative definitions are possible. Other systems will choose different symbols, will select different sets of connectives as primitive, and will use parentheses differently (or even not at all, as in the case of so-called

Polish notation). What all approaches have in common, though, is that the formation rules define the set of formulas *inductively*. If done properly, every expression can result essentially in only one way according to the formation rules. The inductive definition resulting in expressions that are *uniquely readable* means we can give meanings to these expressions using the same method—inductive definition.

Giving the meaning of expressions is the domain of semantics. The central concept in semantics for propositional logic is that of satisfaction in a valuation. A valuation $\mathfrak{v}$ assigns truth values $\mathbb{T}$, $\mathbb{F}$ to the propositional variables. Any valuation determines a truth value $\overline{\mathfrak{v}}(\varphi)$ for any formula φ. A formula is satisfied in a valuation $\mathfrak{v}$ iff $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$—we write this as $\mathfrak{v} \vDash \varphi$. This relation can also be defined by induction on the structure of φ, using the truth functions for the logical connectives to define, say, satisfaction of $\varphi \wedge \psi$ in terms of satisfaction (or not) of φ and ψ.

On the basis of the satisfaction relation $\mathfrak{v} \vDash \varphi$ for sentences we can then define the basic semantic notions of tautology, entailment, and satisfiability. A formula is a tautology, $\vDash \varphi$, if every valuation satisfies it, i.e., $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$ for any $\mathfrak{v}$. It is entailed by a set of formulas, $\Gamma \vDash \varphi$, if every valuation that satisfies all the formulas in Γ also satisfies φ. And a set of formulas is satisfiable if some valuation satisfies all formulas in it at the same time. Because formulas are inductively defined, and satisfaction is in turn defined by induction on the structure of formulas, we can use induction to prove properties of our semantics and to relate the semantic notions defined.

syn.2 Propositional Formulas

Formulas of propositional logic are built up from *propositional variables*, the propositional constant $\bot$ and the propositional constant $\top$ using *logical connectives*. pl:syn:fml:sec

1. A denumerable set At_0 of propositional variables p_0, p_1, ...
2. The propositional constant for falsity $\bot$.
3. The propositional constant for truth $\top$.
4. The logical connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (conditional), $\leftrightarrow$ (biconditional)
5. Punctuation marks: (,), and the comma.

We denote this language of propositional logic by $\mathcal{L}_0$.

.

intro You may be familiar with different terminology and symbols than the ones we use above. Logic texts (and teachers) commonly use either $\sim$, $\neg$, and ! for "negation", $\wedge$, $\cdot$, and & for "conjunction". Commonly used symbols for the "conditional" or "implication" are $\rightarrow$, $\Rightarrow$, and $\supset$. Symbols for "biconditional," "bi-implication," or "(material) equivalence" are $\leftrightarrow$, $\Leftrightarrow$, and $\equiv$. The $\bot$ symbol is variously called "falsity," "falsum," "absurdity," or "bottom." The $\top$ symbol is variously called "truth," "verum," or "top."

pl:syn:fml: defn:formulas **Definition syn.1 (Formula).** The set $\mathrm{Frm}(\mathcal{L}_0)$ of *formulas* of propositional logic is defined inductively as follows:

1. $\bot$ is an atomic formula.
2. $\top$ is an atomic formula.
3. Every propositional variable p_i is an atomic formula.
4. If φ is a formula, then $\neg\varphi$ is a formula.
5. If φ and ψ are formulas, then $(\varphi \land \psi)$ is a formula.
6. If φ and ψ are formulas, then $(\varphi \lor \psi)$ is a formula.
7. If φ and ψ are formulas, then $(\varphi \to \psi)$ is a formula.
8. If φ and ψ are formulas, then $(\varphi \leftrightarrow \psi)$ is a formula.
9. Nothing else is a formula.

The definition of formulas is an *inductive definition.* Essentially, we construct the set of formulas in infinitely many stages. In the initial stage, we pronounce all atomic formulas to be formulas; this corresponds to the first few cases of the definition, i.e., the cases for $\top$, $\bot$, p_i. "Atomic formula" thus means any formula of this form. explanation

The other cases of the definition give rules for constructing new formulas out of formulas already constructed. At the second stage, we can use them to construct formulas out of atomic formulas. At the third stage, we construct new formulas from the atomic formulas and those obtained in the second stage, and so on. A formula is anything that is eventually constructed at such a stage, and nothing else.

When writing a formula $(\psi * \chi)$ constructed from ψ, χ using a two-place connective $*$, we will often leave out the outermost pair of parentheses and write simply $\psi * \chi$.

Definition syn.2 (Syntactic identity). The symbol $\equiv$ expresses syntactic identity between strings of symbols, i.e., $\varphi \equiv \psi$ iff φ and ψ are strings of symbols of the same length and which contain the same symbol in each place.

The $\equiv$ symbol may be flanked by strings obtained by concatenation, e.g., $\varphi \equiv (\psi \lor \chi)$ means: the string of symbols φ is the same string as the one obtained by concatenating an opening parenthesis, the string ψ, the $\lor$ symbol, the string χ, and a closing parenthesis, in this order. If this is the case, then we know that the first symbol of φ is an opening parenthesis, φ contains ψ as a substring (starting at the second symbol), that substring is followed by $\lor$, etc.

syn.3 Preliminaries

pl:syn:pre:sec

Theorem syn.3 (*Principle of induction on formulas*). *If some property P holds for all the atomic formulas and is such that* pl:syn:pre:thm:induction

1. *it holds for $\neg\varphi$ whenever it holds for φ;*
2. *it holds for $(\varphi \land \psi)$ whenever it holds for φ and ψ;*
3. *it holds for $(\varphi \lor \psi)$ whenever it holds for φ and ψ;*
4. *it holds for $(\varphi \to \psi)$ whenever it holds for φ and ψ;*
5. *it holds for $(\varphi \leftrightarrow \psi)$ whenever it holds for φ and ψ;*

then P holds for all formulas.

Proof. Let S be the collection of all formulas with property P. Clearly $S \subseteq \mathrm{Frm}(\mathcal{L}_0)$. S satisfies all the conditions of Definition syn.1: it contains all atomic formulas and is closed under the logical operators. $\mathrm{Frm}(\mathcal{L}_0)$ is the smallest such class, so $\mathrm{Frm}(\mathcal{L}_0) \subseteq S$. So $\mathrm{Frm}(\mathcal{L}_0) = S$, and every formula has property P. □

Proposition syn.4. *Any formula in* $\mathrm{Frm}(\mathcal{L}_0)$ *is* balanced, *in that it has as many left parentheses as right ones.* pl:syn:pre:prop:balanced

Problem syn.1. Prove Proposition syn.4

Proposition syn.5. *No proper initial segment of a formula is a formula.* pl:syn:pre:prop:noinit

Problem syn.2. Prove Proposition syn.5

Proposition syn.6 (Unique Readability). *Any formula φ in* $\mathrm{Frm}(\mathcal{L}_0)$ *has exactly one parsing as one of the following*

1. $\bot$.
2. $\top$.
3. p_n *for some* $p_n \in \mathrm{At}_0$.
4. $\neg\psi$ *for some formula* ψ.
5. $(\psi \land \chi)$ *for some formulas* ψ *and* χ.
6. $(\psi \lor \chi)$ *for some formulas* ψ *and* χ.
7. $(\psi \to \chi)$ *for some formulas* ψ *and* χ.
8. $(\psi \leftrightarrow \chi)$ *for some formulas* ψ *and* χ.

Moreover, this parsing is unique.

Proof. By induction on φ. For instance, suppose that φ has two distinct readings as $(\psi \to \chi)$ and $(\psi' \to \chi')$. Then ψ and ψ' must be the same (or else one would be a proper initial segment of the other); so if the two readings of φ are distinct it must be because χ and χ' are distinct readings of the same sequence of symbols, which is impossible by the inductive hypothesis. □

Definition syn.7 (Uniform Substitution). If φ and ψ are formulas, and p_i is a propositional variable, then $\varphi[\psi/p_i]$ denotes the result of replacing each occurrence of p_i by an occurrence of ψ in φ; similarly, the simultaneous substitution of $p_1, \dots, p_n$ by formulas $\psi_1, \dots, \psi_n$ is denoted by $\varphi[\psi_1/p_1, \dots, \psi_n/p_n]$.

Problem syn.3. For each of the five formulas below determine whether the formula can be expressed as a substitution $\varphi[\psi/p_i]$ where φ is (i) p_0; (ii) $(\neg p_0 \land p_1)$; and (iii) $((\neg p_0 \to p_1) \land p_2)$. In each case specify the relevant substitution.

1. p_1
2. $(\neg p_0 \land p_0)$
3. $((p_0 \lor p_1) \land p_2)$
4. $\neg((p_0 \to p_1) \land p_2)$
5. $((\neg(p_0 \to p_1) \to (p_0 \lor p_1)) \land \neg(p_0 \land p_1))$

Problem syn.4. Give a mathematically rigorous definition of $\varphi[\psi/p]$ by induction.

syn.4 Formation Sequences

pl:syn:fseq:sec Defining formulas via an inductive definition, and the complementary technique of proving properties of formulas via induction, is an elegant and efficient approach. However, it can also be useful to consider a more bottom-up, step-by-step approach to the construction of formulas, which we do here using the notion of a *formation sequence*.

pl:syn:fseq:defn:fseq-frm **Definition syn.8 (Formation sequences for formulas).** A finite sequence $\langle \varphi_0, \dots, \varphi_n \rangle$ of strings of symbols from the language $\mathcal{L}_0$ is a *formation sequence* for φ if $\varphi \equiv \varphi_n$ and for all $i \leq n$, either φ_i is an atomic formula or there exist $j, k < i$ such that one of the following holds:

1. $\varphi_i \equiv \neg\varphi_j$.
2. $\varphi_i \equiv (\varphi_j \land \varphi_k)$.
3. $\varphi_i \equiv (\varphi_j \lor \varphi_k)$.
4. $\varphi_i \equiv (\varphi_j \to \varphi_k)$.
5. $\varphi_i \equiv (\varphi_j \leftrightarrow \varphi_k)$.

Example syn.9.

$$\langle p_0, p_1, (p_1 \wedge p_0), \neg(p_1 \wedge p_0)\rangle$$

is a formation sequence of $\neg(p_1 \wedge p_0)$, as is

$$\langle p_0, p_1, p_0, (p_1 \wedge p_0), (p_0 \rightarrow p_1), \neg(p_1 \wedge p_0)\rangle.$$

As can be seen from the second example, formation sequences may contain 'junk': formulas which are redundant or do not contribute to the construction.

Proposition syn.10. *Every formula φ in $\mathrm{Frm}(\mathcal{L}_0)$ has a formation sequence.* *pl:syn:fseq: prop:formed*

Proof. Suppose φ is atomic. Then the sequence $\langle\varphi\rangle$ is a formation sequence for φ. Now suppose that ψ and χ have formation sequences $\langle\psi_0, \dots, \psi_n\rangle$ and $\langle\chi_0, \dots, \chi_m\rangle$ respectively.

1. If $\varphi \equiv \neg\psi$, then $\langle\psi_0, \dots, \psi_n, \neg\psi_n\rangle$ is a formation sequence for φ.
2. If $\varphi \equiv (\psi \wedge \chi)$, then $\langle\psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \wedge \chi_m)\rangle$ is a formation sequence for φ.
3. If $\varphi \equiv (\psi \vee \chi)$, then $\langle\psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \vee \chi_m)\rangle$ is a formation sequence for φ.
4. If $\varphi \equiv (\psi \rightarrow \chi)$, then $\langle\psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \rightarrow \chi_m)\rangle$ is a formation sequence for φ.
5. If $\varphi \equiv (\psi \leftrightarrow \chi)$, then $\langle\psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \leftrightarrow \chi_m)\rangle$ is a formation sequence for φ.

By the principle of induction on formulas, every formula has a formation sequence. □

We can also prove the converse. This is important because it shows that our two ways of defining formulas are equivalent: they give the same results. It also means that we can prove theorems about formulas by using ordinary induction on the length of formation sequences.

Lemma syn.11. *Suppose that $\langle\varphi_0, \dots, \varphi_n\rangle$ is a formation sequence for φ_n, and that $k \leq n$. Then $\langle\varphi_0, \dots, \varphi_k\rangle$ is a formation sequence for φ_k.* *pl:syn:fseq: lem:fseq-init*

Proof. Exercise. □

Theorem syn.12. $\mathrm{Frm}(\mathcal{L}_0)$ *is the set of all expressions (strings of symbols) in the language $\mathcal{L}_0$ with a formation sequence.* *pl:syn:fseq: thm:fseq-frm-equiv*

Proof. Let F be the set of all strings of symbols in the language $\mathcal{L}_0$ that have a formation sequence. We have seen in Proposition syn.10 that $\mathrm{Frm}(\mathcal{L}_0) \subseteq F$, so now we prove the converse.

Suppose φ has a formation sequence $\langle \varphi_0, \dots, \varphi_n \rangle$. We prove that $\varphi \in \mathrm{Frm}(\mathcal{L}_0)$ by strong induction on n. Our induction hypothesis is that every string of symbols with a formation sequence of length $m < n$ is in $\mathrm{Frm}(\mathcal{L}_0)$. By the definition of a formation sequence, either φ_n is atomic or there must exist $j, k < n$ such that one of the following is the case:

1. $\varphi_n \equiv \neg\varphi_j$.
2. $\varphi_n \equiv (\varphi_j \land \varphi_k)$.
3. $\varphi_n \equiv (\varphi_j \lor \varphi_k)$.
4. $\varphi_n \equiv (\varphi_j \rightarrow \varphi_k)$.
5. $\varphi_n \equiv (\varphi_j \leftrightarrow \varphi_k)$.

Now we reason by cases. If φ_n is atomic then $\varphi_n \in \mathrm{Frm}(\mathcal{L}_0)$. Suppose instead that $\varphi \equiv (\varphi_j \land \varphi_k)$. By Lemma syn.11, $\langle \varphi_0, \dots, \varphi_j \rangle$ and $\langle \varphi_0, \dots, \varphi_k \rangle$ are formation sequences for φ_j and φ_k respectively. Since these are proper initial subsequences of the formation sequence for φ, they both have length less than n. Therefore by the induction hypothesis, φ_j and φ_k are in $\mathrm{Frm}(\mathcal{L}_0)$, and so by the definition of a formula, so is $(\varphi_j \land \varphi_k)$. The other cases follow by parallel reasoning. □

syn.5 Valuations and Satisfaction

pl:syn:val:sec

Definition syn.13 (Valuations). Let $\{\mathbb{T}, \mathbb{F}\}$ be the set of the two truth values, "true" and "false." A *valuation* for $\mathcal{L}_0$ is a function $\mathfrak{v}$ assigning either $\mathbb{T}$ or $\mathbb{F}$ to the propositional variables of the language, i.e., $\mathfrak{v} \colon \mathrm{At}_0 \to \{\mathbb{T}, \mathbb{F}\}$.

Definition syn.14. Given a valuation $\mathfrak{v}$, define the evaluation function $\overline{\mathfrak{v}} \colon \mathrm{Frm}(\mathcal{L}_0) \to$

$\{\mathbb{T}, \mathbb{F}\}$ inductively by:

$$\begin{aligned}
\overline{\mathfrak{v}}(\bot) &= \mathbb{F};\\
\overline{\mathfrak{v}}(\top) &= \mathbb{T};\\
\overline{\mathfrak{v}}(p_n) &= \mathfrak{v}(p_n);\\
\overline{\mathfrak{v}}(\lnot\varphi) &= \begin{cases} \mathbb{T} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{F};\\ \mathbb{F} & \text{otherwise.}\end{cases}\\
\overline{\mathfrak{v}}(\varphi \land \psi) &= \begin{cases} \mathbb{T} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{T} \text{ and } \overline{\mathfrak{v}}(\psi) = \mathbb{T};\\ \mathbb{F} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{F} \text{ or } \overline{\mathfrak{v}}(\psi) = \mathbb{F}.\end{cases}\\
\overline{\mathfrak{v}}(\varphi \lor \psi) &= \begin{cases} \mathbb{T} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{T} \text{ or } \overline{\mathfrak{v}}(\psi) = \mathbb{T};\\ \mathbb{F} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{F} \text{ and } \overline{\mathfrak{v}}(\psi) = \mathbb{F}.\end{cases}\\
\overline{\mathfrak{v}}(\varphi \to \psi) &= \begin{cases} \mathbb{T} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{F} \text{ or } \overline{\mathfrak{v}}(\psi) = \mathbb{T};\\ \mathbb{F} & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{T} \text{ and } \overline{\mathfrak{v}}(\psi) = \mathbb{F}.\end{cases}\\
\overline{\mathfrak{v}}(\varphi \leftrightarrow \psi) &= \begin{cases} \mathbb{T} & \text{if } \overline{\mathfrak{v}}(\varphi) = \overline{\mathfrak{v}}(\psi);\\ \mathbb{F} & \text{if } \overline{\mathfrak{v}}(\varphi) \neq \overline{\mathfrak{v}}(\psi).\end{cases}
\end{aligned}$$

explanation

The clauses correspond to the following truth tables:

φ	$\lnot\varphi$
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$

φ	ψ	$\varphi \land \psi$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

φ	ψ	$\varphi \lor \psi$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

φ	ψ	$\varphi \to \psi$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$

φ	ψ	$\varphi \leftrightarrow \psi$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$

Problem syn.5. Consider adding to $\mathcal{L}_0$ a ternary connective $\Diamond$ with evaluation given by

$$\overline{\mathfrak{v}}(\Diamond(\varphi, \psi, \chi)) = \begin{cases} \overline{\mathfrak{v}}(\psi) & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{T};\\ \overline{\mathfrak{v}}(\chi) & \text{if } \overline{\mathfrak{v}}(\varphi) = \mathbb{F}.\end{cases}$$

Write down the truth table for this connective.

Theorem syn.15 (Local Determination). *Suppose that $\mathfrak{v}_1$ and $\mathfrak{v}_2$ are valuations that agree on the propositional letters occurring in φ, i.e., $\mathfrak{v}_1(p_n) = \mathfrak{v}_2(p_n)$ whenever p_n occurs in some formula φ. Then $\overline{\mathfrak{v}_1}$ and $\overline{\mathfrak{v}_2}$ also agree on φ, i.e., $\overline{\mathfrak{v}_1}(\varphi) = \overline{\mathfrak{v}_2}(\varphi)$.* pl:syn:val: thm:LocalDetermination

Proof. By induction on φ. □

pl:syn:val: defn:satisfaction **Definition syn.16 (Satisfaction).** We can inductively define the notion of *satisfaction of a formula φ by a valuation $\mathfrak{v}$*, $\mathfrak{v} \vDash \varphi$, as follows. (We write $\mathfrak{v} \nvDash \varphi$ to mean "not $\mathfrak{v} \vDash \varphi$.")

1. $\varphi \equiv \bot$: $\mathfrak{v} \nvDash \varphi$.
2. $\varphi \equiv \top$: $\mathfrak{v} \vDash \varphi$.
3. $\varphi \equiv p_i$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v}(p_i) = \mathbb{T}$.
4. $\varphi \equiv \neg\psi$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \nvDash \psi$.
5. $\varphi \equiv (\psi \land \chi)$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \vDash \psi$ and $\mathfrak{v} \vDash \chi$.
6. $\varphi \equiv (\psi \lor \chi)$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \vDash \psi$ or $\mathfrak{v} \vDash \chi$ (or both).
7. $\varphi \equiv (\psi \rightarrow \chi)$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \nvDash \psi$ or $\mathfrak{v} \vDash \chi$ (or both).
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{v} \vDash \varphi$ iff either both $\mathfrak{v} \vDash \psi$ and $\mathfrak{v} \vDash \chi$, or neither $\mathfrak{v} \vDash \psi$ nor $\mathfrak{v} \vDash \chi$.

If Γ is a set of formulas, $\mathfrak{v} \vDash \Gamma$ iff $\mathfrak{v} \vDash \varphi$ for every $\varphi \in \Gamma$.

pl:syn:val: prop:sat-value **Proposition syn.17.** $\mathfrak{v} \vDash \varphi$ *iff* $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$.

Proof. By induction on φ. □

Problem syn.6. Prove Proposition syn.17

syn.6 Semantic Notions

pl:syn:sem: sec We define the following semantic notions:

Definition syn.18.

1. A formula φ is *satisfiable* if for some $\mathfrak{v}$, $\mathfrak{v} \vDash \varphi$; it is *unsatisfiable* if for no $\mathfrak{v}$, $\mathfrak{v} \vDash \varphi$;
2. A formula φ is a *tautology* if $\mathfrak{v} \vDash \varphi$ for all valuations v;
3. A formula φ is *contingent* if it is satisfiable but not a tautology;
4. If Γ is a set of formulas, $\Gamma \vDash \varphi$ ("Γ entails φ") if and only if $\mathfrak{v} \vDash \varphi$ for every valuation $\mathfrak{v}$ for which $\mathfrak{v} \vDash \Gamma$.
5. If Γ is a set of formulas, Γ is *satisfiable* if there is a valuation $\mathfrak{v}$ for which $\mathfrak{v} \vDash \Gamma$, and Γ is *unsatisfiable* otherwise.

Problem syn.7. For each of the following four formulas determine whether it is (a) satisfiable, (b) tautology, and (c) contingent.

1. $(p_0 \rightarrow (\neg p_1 \rightarrow \neg p_0))$.

2. $((p_0 \land \lnot p_1) \to (\lnot p_0 \land p_2)) \leftrightarrow ((p_2 \to p_0) \to (p_0 \to p_1))$.
3. $(p_0 \leftrightarrow p_1) \to (p_2 \leftrightarrow \lnot p_1)$.
4. $((p_0 \leftrightarrow (\lnot p_1 \land p_2)) \lor (p_2 \to (p_0 \leftrightarrow p_1)))$.

Proposition syn.19. *pl:syn:sem: prop:semanticalfacts*

1. *φ is a tautology if and only if $\emptyset \vDash \varphi$;*
2. *If $\Gamma \vDash \varphi$ and $\Gamma \vDash \varphi \to \psi$ then $\Gamma \vDash \psi$;*
3. *If Γ is satisfiable then every finite subset of Γ is also satisfiable;*
4. *Monotonicity: if $\Gamma \subseteq \Delta$ and $\Gamma \vDash \varphi$ then also $\Delta \vDash \varphi$;* *pl:syn:sem: def:monotonicity*
5. *Transitivity: if $\Gamma \vDash \varphi$ and $\Delta \cup \{\varphi\} \vDash \psi$ then $\Gamma \cup \Delta \vDash \psi$.* *pl:syn:sem: def:Cut*

Proof. Exercise. □

Problem syn.8. Prove Proposition syn.19

Proposition syn.20. *$\Gamma \vDash \varphi$ if and only if $\Gamma \cup \{\lnot\varphi\}$ is unsatisfiable.* *pl:syn:sem: prop:entails-unsat*

Proof. Exercise. □

Problem syn.9. Prove Proposition syn.20

Theorem syn.21 (Semantic Deduction Theorem). *$\Gamma \vDash \varphi \to \psi$ if and only if $\Gamma \cup \{\varphi\} \vDash \psi$.* *pl:syn:sem: thm:sem-deduction*

Proof. Exercise. □

Problem syn.10. Prove Theorem syn.21

Photo Credits

Bibliography