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 $\lnot$, $\land$, $\lor$, $\rightarrow$, and $\leftrightarrow$. Intuitively, a propositional variable p stands for a sentence or proposition that is be true or false. Whenever the "truth value" of the propositional variable in a formula are 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 chose different symbols, will select different sets of connectives as primitive, 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 propositonal 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 $\varphi \land \psi$ 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), $\land$ (conjunction), $\lor$ (disjunction), $\rightarrow$ (conditional), $\leftrightarrow$ (biconditional)
5. Punctuation marks: (,), and the comma.

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", $\land$, $\cdot$, 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 formula.
5. If φ and ψ are formulas, then $(\varphi \wedge \psi)$ is a formula.
6. If φ and ψ are formulas, then $(\varphi \vee \psi)$ is a formula.
7. If φ and ψ are formulas, then $(\varphi \rightarrow \psi)$ is a formula.
8. If φ and ψ are formulas, then $(\varphi \leftrightarrow \psi)$ is a formula.
9. If φ is a formula and x is a variable, then $\forall x\, \varphi$ is a formula.
10. If φ is a formula and x is a variable, then $\exists x\, \varphi$ is a formula.
11. Nothing else is a formula.

The definitions of the set of terms and that of formulas are *inductive definitions*. 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.

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., $\varphi \equiv (\psi \vee \chi)$ means: the string of symbols φ is the same string as the one obtained by concatenating an opening parenthesis, the string ψ, the $\vee$ 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 $\vee$, etc.

syn.3 Preliminaries

pl:syn:pre:sec

Theorem syn.3. Principle of induction on formulas*: If some property P holds of 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. *if holds for and* $(\varphi \wedge \psi)$ *whenever it holds for* φ *and* ψ*;*
3. *if holds for and* $(\varphi \vee \psi)$ *whenever it holds for* φ *and* ψ*;*
4. *if holds for and* $(\varphi \rightarrow \psi)$ *whenever it holds for* φ *and* ψ*;*
5. *if holds for and* $(\varphi \leftrightarrow \psi)$ *whenever it holds for* φ *and* ψ*;*

then P *holds of 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 $\mathrm{Frm} \subseteq S$. So $\mathrm{Frm} = S$, and every formula has propery 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* ψ *in* $\mathrm{Frm}(\mathcal{L}_0)$.
5. $(\psi \wedge \chi)$ *for some formulas* ψ *and* χ.
6. $(\psi \vee \chi)$ *for some formulas* ψ *and* χ.
7. $(\psi \rightarrow \chi)$ *for some formulas* ψ *and* χ.
8. $(\psi \leftrightarrow \chi)$ *for some formulas* ψ *and* χ.

Moreover, such 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 $p_1, \ldots, p_n$ by formulas $\psi_1, \ldots, B_n$ is denoted by $\varphi[\psi_1/p_1, \ldots, \psi_n/p_n]$.

Problem syn.3. Give a mathematically rigorous definition of $\varphi[\psi/p]$ by induction.

syn.4 Valuations and Satisfaction

pl:syn:val:sec

Definition syn.8 (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., $\mathfrak{v}\colon \text{At}_0 \to \{\mathbb{T}, \mathbb{F}\}$.

Definition syn.9. Given a valuation v, define the evaluation function $\overline{\mathfrak{v}}(:)\text{Frm}(\mathcal{L}_0) \to \{\mathbb{T}, \mathbb{F}\}$ inductively by:

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

The valuation clauses correspond to the following truth tables: explanation

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

Theorem syn.10 (Local Determination)**.** *Suppose that $\mathfrak{v}_1$ and $\mathfrak{v}_2$ are valuations that agree on the propositional letters occurring in φ, i.e., $\overline{\mathfrak{v}_1}(p_n) = \overline{\mathfrak{v}_2}(p_n)$ whenever p_n occurs in φ. Then they also agree on any φ, i.e., $\overline{\mathfrak{v}_1}(\varphi) = \overline{\mathfrak{v}_2}(\varphi)$.* *pl:syn:val: thm:LocalDetermination*

Proof. By induction on φ. □

Definition syn.11 (Satisfaction)**.** Using the evaluation function, we can define the notion of *satisfaction of a formula φ by a valuation $\mathfrak{v}$*, $\mathfrak{v} \vDash \varphi$, inductively as follows. (We write $\mathfrak{v} \nvDash \varphi$ to mean "not $\mathfrak{v} \vDash \varphi$.") pl:syn:val: defn:satisfaction

1. $\varphi \equiv \bot$: $\mathfrak{v} \nvDash \varphi$.
2. $\varphi \equiv \top$: $\mathfrak{v} \vDash \varphi$.
3. $\varphi \equiv p_i$: $\mathfrak{M} \vDash \varphi$ iff $\overline{\mathfrak{v}}(p_i) = \mathbb{T}$.
4. $\varphi \equiv \neg\psi$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \nvDash \psi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \vDash \psi$ and $\mathfrak{v} \vDash \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{v} \vDash \varphi$ iff $\mathfrak{v} \vDash \varphi$ or $\mathfrak{v} \vDash \psi$ (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 $\varphi \in \Gamma$.

Proposition syn.12. $\mathfrak{v} \vDash \varphi$ *iff* $\overline{\mathfrak{v}}(\varphi) = \mathbb{T}$. *pl:syn:val: prop:sat-value*

Proof. By induction on φ. □

Problem syn.4. Prove Proposition syn.12

syn.5 Semantic Notions

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

Definition syn.13.
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.

pl:syn:sem:prop:semanticalfacts

Proposition syn.14.

1. *φ is a tautology if and only if $\emptyset \vDash \varphi$;*

2. *If $\Gamma \vDash \varphi$ and $\Gamma \vDash \varphi \rightarrow \psi$ then $\Gamma \vDash \psi$;*

3. *If Γ is satisfiable then every finite subset of Γ is also satisfiable;*

pl:syn:sem:def:Monotony

4. *Monotony: if $\Gamma \subseteq \Delta$ and $\Gamma \vDash \varphi$ then also $\Delta \vDash \varphi$;*

pl:syn:sem:def:Cut

5. *Transitivity: if $\Gamma \vDash \varphi$ and $\Delta \cup \{\varphi\} \vDash \psi$ then $\Gamma \cup \Delta \vDash \psi$;*

Proof. Exercise. □

Problem syn.5. Prove Proposition syn.14

pl:syn:sem:prop:entails-unsat

Proposition syn.15. *$\Gamma \vDash \varphi$ if and only if $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable;*

Proof. Exercise. □

Problem syn.6. Prove Proposition syn.15

pl:syn:sem:thm:sem-deduction

Theorem syn.16 (Semantic Deduction Theorem)**.** *$\Gamma \vDash \varphi \rightarrow \psi$ if and only if $\Gamma \cup \{\varphi\} \vDash \psi$.*

Proof. Exercise. □

Problem syn.7. Prove Theorem syn.16

Photo Credits

Bibliography