Remark 4.0.1 Our view of propositional logic is that the meaning or content of a proposition is its truth value. The semantics for propositional logic consists in assigning truth values to propositions. From our examples using truth tables it is clear that the truth value of a proposition is determined once the truth values of the propositional symbols which occur in it have received a truth value. Now we make this precise using structural recursion.

1 Assignments and valuations

Definition 4.1.1 (assignment) A *Boolean assignment* (or *truth assignment*) is a function $\mathcal{A} : PS \to \{\mathbf{T}, \mathbf{F}\}$. That is, $\mathcal{A}$ assigns to each propositional symbol a unique truth value from $\{\mathbf{T}, \mathbf{F}\}$.

We extend an assignment to all propositions faithfully in accordance with the truth tables in **??** by structural recursion.

Theorem 4.1.2 (Boolean valuation) Every Boolean assignment $\mathcal{A} : \mathbf{PS} \to \{\mathbf{T}, \mathbf{F}\}$ uniquely determines a *Boolean valuation* $v_\mathcal{A} : \mathbf{PROP} \to \{\mathbf{T}, \mathbf{F}\}$ which is *faithful* to the meaning of each connective given by the truth tables in Definition 3.1.3. That is, for every proposition α and β and propositional symbol P:

$$
\begin{aligned}
v_\mathcal{A}(P) &= \mathcal{A}(P) \\
v_\mathcal{A}(\top) &= \mathbf{T} \\
v_\mathcal{A}(\bot) &= \mathbf{F} \\
v_\mathcal{A}(\neg\,\alpha) &= \begin{cases} \mathbf{T} & \text{if } v_\mathcal{A}(\alpha) = \mathbf{F}, \\ \mathbf{F} & \text{if } v_\mathcal{A}(\alpha) = \mathbf{T} \end{cases} \\
v_\mathcal{A}(\alpha \wedge \beta) &= \begin{cases} \mathbf{T} & \text{if } v_\mathcal{A}(\alpha) = \mathbf{T} = v_\mathcal{A}(\beta) \\ \mathbf{F} & \text{if at least one of } v_\mathcal{A}(\alpha), v_\mathcal{A}(\beta) = \mathbf{F}; \end{cases} \\
v_\mathcal{A}(\alpha \vee \beta) &= \begin{cases} \mathbf{T} & \text{if at least one of } v_\mathcal{A}(\alpha), v_\mathcal{A}(\beta) = \mathbf{T}, \\ \mathbf{F} & \text{if } v_\mathcal{A}(\alpha) = \mathbf{F} = v_\mathcal{A}(\beta); \end{cases} \\
v_\mathcal{A}(\alpha \to \beta) &= \begin{cases} \mathbf{T} & \text{if at least one of } v_\mathcal{A}(\alpha) = \mathbf{F}, v_\mathcal{A}(\beta) = \mathbf{T}, \\ \mathbf{F} & \text{if } v_\mathcal{A}(\alpha) = \mathbf{T}, v_\mathcal{A}(\beta) = \mathbf{F}; \end{cases} \\
v_\mathcal{A}(\alpha \leftrightarrow \beta) &= \begin{cases} \mathbf{T} & \text{if } v_\mathcal{A}(\alpha) = v_\mathcal{A}(\beta), \\ \mathbf{F} & \text{if } v_\mathcal{A}(\alpha) \neq v_\mathcal{A}(\beta); \end{cases}
\end{aligned}
$$

Proof. For each logical connectives $\diamond$ let $\mathcal{H}_\diamond : BOOL \to BOOL$ be the function whose graph is determined by the truth table:

α	β	$\mathcal{H}_\neg(\alpha)$	$\mathcal{H}_\wedge(\alpha,\beta)$	$\mathcal{H}_\vee(\alpha,\beta)$	$\mathcal{H}_\to(\alpha,\beta)$	$\mathcal{H}_\leftrightarrow(\alpha,\beta)$
T	**T**	**F**	**T**	**T**	**T**	**T**
T	**F**	**F**	**F**	**T**	**F**	**F**
F	**T**	**T**	**F**	**T**	**T**	**F**
F	**F**	**T**	**F**	**F**	**T**	**T**

It follows by structural recursion that for each assignment $\mathcal{A}$ to the propositional symbols, there is a unique function $v_\mathcal{A}$ satisfying each of the conditions of the theorem. More formally, $v_\mathcal{A}$ is the unique function which satisfies for proposition α and β and propositional symbol P:

$$\begin{aligned}
v_{\mathcal{A}}(P) &= \mathcal{A}(P) \\
v_{\mathcal{A}}(\top) &= \mathbf{T} \\
v_{\mathcal{A}}(\bot) &= \mathbf{F} \\
v_{\mathcal{A}}(\neg\,\alpha) &= \mathcal{H}_{\neg}(v_{\mathcal{A}}(\alpha)) \\
v_{\mathcal{A}}(\alpha \diamond \beta) &= \mathcal{H}_{\diamond}\big(v_{\mathcal{A}}(\alpha), v_{\mathcal{A}}(\beta)\big) \qquad \text{for each } \diamond \in \{\,\wedge\,,\,\vee\,,\,\rightarrow\,,\,\leftrightarrow\,\}.
\end{aligned}$$

□

Remark 4.1.3 It would be most inconvenient to have to specify the truth value of every propositional symbol to compute the truth value of a single proposition. Fortunately, this is not the case. It is sufficient to consider only the support of the proposition. Recall, that the *support* of a proposition α is the set of propositional symbols which occur in α.

Lemma 4.1.4 If u and v are two Boolean valuations which agree on the support of α, then $u(\alpha) = v(\alpha)$.

Proof. The proof is by structural induction on α. If P is a propositional symbol, then $u(P) = v(P)$ by hypothesis. Of course, $u(\top) = v(\top)$ and $u(\bot) = v(\bot)$ by definition.

Suppose $\alpha = \neg\,\beta$. Since β has the same support as α, u and v agree on the support of β, so by the induction hypothesis $u(\beta) = v(\beta)$. Then,

$$u(\alpha) = \mathcal{H}_{\neg}\big(u(\beta)\big) = \mathcal{H}_{\neg}\big(v(\beta)\big) = v(\alpha).$$

Suppose $\alpha = (\beta \diamond \gamma)$, for a binary connective $\diamond \in \{\,\wedge\,,\,\vee\,,\,\rightarrow\,,\,\leftrightarrow\,\}$. Since the propositional symbols occurring in β and in γ also occur in α, u and v agree on the support of β and γ, so by the induction hypothesis, $u(\beta) = v(\beta)$ and $u(\gamma) = v(\gamma)$. Then,

$$u(\alpha) = \mathcal{H}_{\diamond}\big(u(\beta), u(\gamma)\big) = \mathcal{H}_{\diamond}\big(v(\beta), v(\gamma)\big) = v(\alpha).$$

□

The previous lemma justifies our use of truth tables:

Example 4.1.5 To compute the truth value of $((P\ \rightarrow\ Q)\ \leftrightarrow\ ((\neg\,P)\ \vee\ Q))$ requires only specifying the values of P and Q. So, the truth value of this proposition on any Boolean assignment is determined by its truth table:

P	Q	$\neg\,P$	$P\ \rightarrow\ Q$	$(\neg\,P)\ \vee\ Q$	$((P\ \rightarrow\ Q)\ \leftrightarrow\ ((\neg\,P)\ \vee\ Q))$
T	**T**	**F**	**T**	**T**	**T**
T	**F**	**F**	**F**	**F**	**T**
F	**T**	**T**	**T**	**T**	**T**
F	**F**	**T**	**T**	**T**	**T**

So, $(P\ \rightarrow\ Q)$ means the same thing (always has the same truth value as) $(\neg\,P\ \vee\ Q)$.

Example 4.1.6 Let $\alpha = (P\ \wedge\ (\neg\,P))$. Since P is the only propositional letter in α, and the truth value of α depends only on the truth assignment to P:

P	$\neg\,P$	$P\ \wedge\ (\neg\,P)$
T	**F**	**F**
F	**T**	**F**

Definition 4.1.7 (satisfaction) If there is some Boolean valuation v which assigns $v(\alpha) = \mathbf{T}$, then v *satisfies* α. If $v(\alpha) = \mathbf{F}$, then v *falsifies* α. A proposition α is *satisfiable* if there is a satisfying Boolean valuation. A proposition α is *unsatisfiable* (or *contradictory* if there is no satisfying valuation.

A set of propositions Γ is *satisfied* by a Boolean valuation v if every proposition Γ is satisfied by v. A set of propositions is *satisfiable* if there is a satisfying Boolean valuation. A set of propositions is *unsatisfiable* if there is no satisfying valuation.

Example 4.1.8 The proposition $((P \rightarrow Q) \leftrightarrow ((\neg P) \vee Q))$ is satisfiable by example 4.1.5 .

The set $\{((P \rightarrow Q) \leftrightarrow ((\neg P) \vee Q))\}$ is satisfiable, as witnessed by the same valuation of example 4.1.5 .

The proposition $(P \wedge (\neg P))$ is unsatisfiable by example 4.1.6.

Remark 4.1.9 If $\Gamma = \emptyset$ then Γ is satisfied by every valuation v, so is satisfiable. The reason is that the only way a valuation could *fail* to satisfy Γ is if it falsified some proposition in Γ. Since Γ has no propositions to be falsified (it contains no propositions at all), Γ must be satisfiable by every valuation.

Definition 4.1.10 (tautology) A proposition α is a *tautology* if every valuation satisfies it. That is, $v(\alpha) = \mathbf{T}$ for all valuations v.

Example 4.1.11 The proposition $P \leftrightarrow \neg\neg P$ is a tautology, as shown by the following truth table.

P	$\neg P$	$\neg\neg P$	$P \leftrightarrow \neg\neg P$
T	**F**	**T**	**T**
F	**T**	**F**	**T**

The following are also tautologies (which you check by a truth table)

$$P \rightarrow P, \qquad (P \rightarrow Q) \leftrightarrow (\neg P \vee Q), \qquad P \vee \neg P$$

The set $\{((P \rightarrow Q) \leftrightarrow ((\neg P) \vee Q)), \neg\neg P\}$ is satisfiable, since the valuation $v(P) = \mathbf{T}$ and $v(Q) = \mathbf{T}$ satisfies both members. The set $\{P, \neg P\}$ is unsatisfiable.

Definition 4.1.12 (tautological implication) Given a set of propositions Γ and proposition α, Γ *tautologically implies* α, denoted by $\Gamma \models \alpha$, if every valuation which satisfies Γ also satisfies α.

Remark 4.1.13 The definition of Γ tautologically implies α is intended to capture the intuitive idea that a set of premisses, Γ, *implies* a conclusion α (and α *follows from* the premisses Γ) if it is impossible for each of the premisses in Γ to be simultaneously true and the conclusion α to be false.

Remark 4.1.14 (empty premisses, unsatisfiable premises) The case where $\Gamma = \emptyset$ is a special case. By remark 4.1.9, Γ is satisfied by every valuation. So, if $\emptyset \models \alpha$, then every valuation must satisfy α and α is a tautology. We will write $\models \alpha$ instead of $\emptyset \models \alpha$.

Another special case of $\Gamma \models \alpha$ is when Γ is unsatisfiable. The only way for $\Gamma \models \alpha$ to *fail* to be true is if some valuation v satisfies Γ and falsifies α; but no valuation satisfies Γ. So, $\Gamma \models \alpha$ must be true when Γ is unsatisfiable. For example,

$$\{P, \neg P\} \models Q.$$

Example 4.1.15 Show that

$$\{P \rightarrow Q, \neg Q\} \models \neg P$$

by the truth table method.

P	Q	$P \to Q$	$\neg Q$	$\neg P$
T	**T**	**T**	**F**	**F**
T	**F**	**F**	**T**	**F**
F	**T**	**T**	**F**	**T**
F	**F**	**T**	**T**	**T**

Line 4 is the only line when $P \to Q$ and $\neg Q$ are simultaneously true; but in this case $\neg P$ is true as well. The other three lines have no bearing on whether $\{P \to Q, \neg Q\} \models \neg P$.

You can verify the following are also tautological consequences

$$\begin{aligned} \{P, P \leftrightarrow Q\} &\models Q \\ \{P, Q\} &\models P \wedge Q \\ \{P \to R, Q \to R\} &\models (P \vee Q) \to R. \end{aligned}$$

Convention 4.1.16 If $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ is a finite set, then we will drop the set braces and write $\gamma_1, \ldots, \gamma_n \models \alpha$.

It is also convenient to use the following notation

$$\Gamma, \alpha \quad \text{means } \Gamma \cup \{\alpha\},$$

so that we write $\Gamma, \alpha \models \beta$ instead of $\Gamma \cup \{\alpha\} \models \beta$.

There is an important relationship between satisfiability and tautological implication.

Lemma 4.1.17 Let Γ, α be a set of propositions

$$\Gamma \models \alpha \quad \text{if and only if} \quad \Gamma, \neg \alpha \text{ is unsatisfiable.}$$

It is also convenient to state this in its contrapositive form

$$\Gamma \not\models \alpha \quad \text{if and only if} \quad \Gamma, \neg \alpha \text{ satisfiable.}$$

Proof. Suppose $\Gamma \models \alpha$. Then for every valuation v which satisfies Γ,it is also true that $v(\alpha) = \mathbf{T}$. So, for any such valuation v, $v(\neg \alpha) = \mathbf{F}$. That is, $\Gamma, \neg \alpha$ is unsatisfiable.

Conversely, if $\Gamma, \neg \alpha$ is unsatisfiable, then any valuation v which satisfies Γ must set $v(\neg \alpha) = \mathbf{F}$, that is, $v(\alpha) = \mathbf{T}$. So, any valuation v that satisfies Γ also satisfies α. That is, $\Gamma \models \alpha$. $\square$

Definition 4.1.18 Propositions α and β are *tautologically equivalent*, denoted by $\alpha \simeq \beta$, if both $\alpha \models \beta$ and $\beta \models \alpha$.

Remark 4.1.19 It will be convenient to use the symbol " $\simeq$ " instead of Enderton's use of "$\models\dashv$".

Example 4.1.20 We will use the truth table method to show $A \to B \simeq \neg A \vee B$.

P	Q	$P \to Q$	$\neg P$	$(\neg P) \vee Q$
T	**T**	**T**	**F**	**T**
T	**F**	**F**	**F**	**F**
F	**T**	**T**	**T**	**T**
F	**F**	**T**	**T**	**T**

$P \to Q \models (\neg P) \vee Q$: lines 1,3,4 are valuations which satisfy $(P \to Q)$ and $((\neg P) \vee Q)$ is also satisfied.

$(\neg P) \vee Q \models P \to Q$: again lines 1,3,4 are valuations which satisfy $((\neg P) \vee Q)$ and $(P \to Q)$ is also satisfied.

It is worth noting that, in fact, $(P \to Q)$ and $((\neg P) \vee Q)$ always have the same truth value. You can verify the following tautological equivalences also hold.

$$P \simeq \neg\neg P, \qquad P \vee Q \simeq \neg((\neg P) \wedge (\neg Q)).$$